SAPIENZA
UNIVERSITÀ DI ROMA

Dottorato di Ricerca in Statistica Metodologica

Tesi di Dottorato XXIV Ciclo – 2009/2011

Dipartimento di Scienze Statistiche

Non Linear Multivariate Time Series: Contribution to Outlier Detection and Threshold AutoRegressive Models

Domenico Cucina

Abstract

Due to the deficiency of linear models in capturing some commonly observed features of time series data, many non-linear time series models have been proposed in the literature. Two models that have gained much attention are the so-called self-exciting threshold autoregressive (SETAR) model and the outlier model. Setar model has been found very effective for modeling and forecasting non linear time series in a wide range of application fields. Furthermore, SETAR model is able to capture nonlinear characteristics as limit cycles, jump resonance, and time irreversibility. Outlier models are important in time series analysis because they can be improve model identification, parameter estimation and forecasting.

Techniques for vector nonlinear time series modeling have only recently begun to be investigated but multivariate nonlinearity analysis requires more research. In this thesis we dealt with outliers and threshold models in a multivariate framework. In particular the attention is focused on a multivariate SETAR (MSETAR) model where each linear regime follows a vector autoregressive (VAR) process and the thresholds are multivariate and the detection of multiple outliers, especially those occurring close in time.

In chapter 2, we propose a methodology based on genetic algorithms (GAs) for building MSETAR models. The GA is designed to estimate the structural parameters, that is to determine the appropriate number of regimes and find multivariate thresholds parameters. The proposed methodology is tested by means of simulated and real time series.

In chapter 3, a class of meta-heuristic methods to detect multiple additive outliers in multivariate time series is proposed. This class includes: simulated annealing, threshold accepting and genetic algorithms. In contrast with many of the existing methods, they do not require to specify a vector ARMA model for the data and detect any number of potential outliers simultaneously reducing possible masking and swamping effects. A generalised AIC-like criterion is used as an objective function where the penalty constant is suggested by both a simulation study and a theoretical approximation. The comparison and the performance of the proposed methods are illustrated by simulation studies and real data analysis. Simulation results show that the proposed approaches perform well also for detecting patches of additive outliers.

Contents

Chapter 1

Introduction

1.1 Linearity and non linearity

From the pioneering work of Yule (1927) on AR modelling of the sunspot numbers to the work of Box & Jenkins (1970) that marked the maturity of ARMA modeling in terms of theory and methodology, *linear Gaussian time series models* flourished and dominated both theoretical explorations and pratical applications (Fan & Yao 2003). The popularity of these models is certainly due to their relatively simple mathematical tractability and also to the existence of computer software incorporating the Box-Jenkins methodology. The basis for such modelling approaches was the Wold representation theorem: any stationary process $\{X_t\}$ with a purely continuous spectrum and (non-normalized) spectral density function $h(\omega)$ can be represented as a linear combination of the term of an uncorrelated process ϵ_t (Priestley 1981):

$$X_t = \sum_{u=-\infty}^{+\infty} \gamma_u \epsilon_{t-u} \quad \sum_{u=-\infty}^{+\infty} \gamma_u^2 < \infty \tag{1.1}$$

Moreover, if the spectral density function $h(\omega)$ satisfies the Paley-Wiener condition:

$$\int_{-\pi}^{\pi} log\{h(\omega)\} d\omega \ > -\infty, \tag{1.2}$$

then the process $X(t)$ assume the one-sided form:

$$X_t = \sum_{u=0}^{+\infty} \gamma_u \epsilon_{t-u}. \tag{1.3}$$

The condition 1.2 plays a fundamental role in real prediction theory. It's a fairly weak condition, and we may expect it to hold in the vast majority of cases (certainly, in any situations of pratical interest). Wold's theorem shows that any stationary process may be approximated by linear models. This makes us understand the enormous importance of linearity in the study of time series. The statement, however, shows some limitations of such models: the variables are uncorrelated and not independent and the representation may require a potentially infinite number of coefficients. Some considerations are needed to clarify the importance of the dichotomy uncorrelation-independence. The aim of each model is to produce independent residuals (and possibly Gaussian) order to extract all the information in the data. Uncorrelated residuals do not ensure that the structure of the data has been captured by the model. For example, consider the problem of predicting the future value of the process, given observations up to time t. In the case of the strictly independent process, e_t, the past contains no information on the future, and hence the best predictor of a future value of e_t is simply its (unconditional) mean. For the uncorrelated process, ϵ_t, it is still true that if we restricted attention to linear predictors then, in this sense, the past contains no information on the future. However, the past may well contain useful information on the future values if we allow predictors which are non-linear functions of the observations. The following example illustrates this point. Let the process η_t be defined by (Priestley 1981):

$$\eta_t = e_t + \beta e_{t-1} e_{t-2} \tag{1.4}$$

where e_t is an independent process with zero mean and constant variance. It is a clear that η_t also has zero mean and constant variance, and its autocovariance functions assume value zero for all lag $s \neq 0$. Then, η_t is an uncorrelated process, and, as far as its second order properties are concerned, it behaves just like an independent process. However, given observations up to time t one can clearly construct a non-trivial predictor of η_{t+1}. Specifically, if we adopt the mean square error criterion, the optimal predictor of η_{t+h} is its conditional expectation, i.e.:

$$\hat{\eta}_{t+h} = E\left[\eta_{t+h} | \eta_t, \eta_{t-1}, \ldots\right], \tag{1.5}$$

and for $h = 1$ we find from (1.5):

$$\hat{\eta}_{t+1} = \beta e_t e_{t-1} \tag{1.6}$$

As noted by Granger & Andersen (1978), if a process η_t of the above form was obtained as the residual from a more general model, all the conventional test for white noise based on the behaviour of the autocovariance or autocorrelation function would confirm that the residuals were, white noise, and hence there was no further model structure left to fit. However, as we have seen, one could certainly exploit the non linear structure of the η_t process in order to improve the predictors of the original series.

Then, linear models for stationary series may not be adequate even though they produce uncorrelated residuals. In fact, uncorrelated residuals may be very far from the independence. In summary, a model can be said satisfactory when extracting all information from the data, that is, when the residuals of the model are independent.

This means that the covariance matrix is not sufficient to fully characterize a process. But it's well known that, under hypothesis of normality and in this case only, uncorrelation is equivalent to the independence and the covariance matrix completely characterizes the process. In conclusion, if the process is Gaussian, then the Wold representation is an appropriate model.

Wold's theorem provides one of several possible representations, and therefore does not exclude that the nature of relationships between the variables of the process is nonlinear, or that there is a representation of X_t through the use of nonlinear functions, which is simpler and less expensive in terms of parameters of (1.3) that involves an infinite number of parameters h_u (Battaglia 2007).

Some nonstandard features, which we refer to as nonlinear features from now on, have been well-observed in many real time series data:

In the early 1950s, the Australian statistician, Pat Moran, spent many of his working hours at the library of the Department of Zoology, Oxford, which became his office. As a result, he became interested in ecology and met the Oxford ecologist, Charles Elton. In particular, he was interested in the famous 10-year lynx cycle, which was and still is of immense interest to the ecologists. In Moran (1953*a*), among the many available annual records of lynx trappings, he chose the longest one, namely the 1821-1934 record of the MacKenzie River district in Canada. He remarked on the asymmetry of the lynx cycle and that lynx dynamics *would have to be represented by nonlinear equations* (Moran (1953*b*), p.292).

Whittle (1954) analyzed the seiche time series of 660 observations at 15 second intervals of the water level in a rock channel at Island Bay on the Wellington coast in his native country, New Zealand. Whittle noted a significant arithmetical relationship among the periods of the prominent peaks of the spectral density function estimate on time series. Such a relationship is beyond the scope of linear models.

Tong et al. (1985) studied the Jokulsa river system, consisting of three time series in 1972: river-flow, precipitation and temperature. The nonlinearity is a result of the phase change from ice to water. The inadequacy of linear models is self-evident in this case.

Modeling these nonstandard features or other nonstandard as *nonnormality*, *asymmetric cycles*, *bimodality*, *non linear relationship between lagged variables*, *time irreversibity*, *structural breaks* or *outliers* is beyond the scope of Gaussian time series models.

Due to the deficiency of linear models in capturing some commonly observed features of time series data, many non-linear time series models have been proposed in the literature. The first systematic study of non-linear models is due to Wiener in 1958, which considered an extension of the Volterra model of the following form (this representation exists under general conditions):

$$\begin{aligned} X_t &= \sum_{u=0}^{\infty}\gamma_u\epsilon_{t-u} + \sum_{u=0}^{\infty}\sum_{i=0}^{\infty}\gamma_{ui}\epsilon_{t-u}\epsilon_{t-i} + \sum_{u=0}^{\infty}\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\gamma_{uij}\epsilon_{t-u}\epsilon_{t-i}\epsilon_{t-j} + \\ &\quad + \sum_{u=0}^{\infty}\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\sum_{l=0}^{\infty}\gamma_{uijl}\epsilon_{t-u}\epsilon_{t-i}\epsilon_{t-j}\epsilon_{t-l} + \dots \end{aligned} \tag{1.7}$$

The Volterra expansion provides a general representation of a nonlinear time series. If we stop the Volterra series expansion of the first term, we obtain the linear model that represents the purely random component of the Wold decomposition if ϵ_t is a weakly stationary white noise and if the condition (1.2) is satisfied. The general relationship between a linear time series and a nonlinear time series is easy to see: the nonlinear equation has a lot of cross-product terms.

The class of non-linear models is much larger than that of linear models. Once we decide to estimate a nonlinear model, we have the task of deciding which of an arbitrary large number of functions to estimate. The nonlinear models have evolved to represent different possible non-linearity features. The contributions in the literature can be divided roughly into two categories: nonlinearity in conditional mean and nonlinearity in conditional variance (conditional heteroscedasticity).

The first category includes, for example, the non-linear autoregressive models, (NLAR, Jones (1978)), the threshold models (SETAR, Tong & Lim (1980)), the exponential autoregressive models (EXPAR, Ozaki (1982)), outlier models (Fox (1972); Tsay (1988)) and changes in level (Tsay (1986); Tsay (1986); Bai & Perron (2003)). The second category includes, for example, the conditional variance models ARCH (Engle (1982)) and GARCH (Bollerslev (1986)). Other models are not easily classified in this scheme: bilinear models (BL, Subba (1981)) generate sudden explosions in the values of a series. These explosions can also be interpreted as changes in variance and this accounts for the relationship between BL and ARCH.

Priestley (1988) presented a general model (SDM = State Dependent Models), which includes as special cases ARMA, SETAR, BL and EXPAR models. This formulation is perhaps little known for computational difficulties encountered in practical application of the SDM.

This brief overview is not the end of the recent history of non-linearity. Around the same time when non-linear statistical models were developed, another line of investigation on the non-linearity was just beginning, the study of complex nonlinear dynamics or chaos. It is usually believed that Poincaré is the first one who studied chaos. Then Lorenz (1963) revealed the butterfly effect in studying the weather prediction and is thus recognized as the father of chaos. But the formal use of chaos is from the works of May (1976) and Li & Yorke (1975). After that, chaos have been widely studied and a lot of important concepts has been introduced, such as the dimensions, Lyapunov exponents, Fourier transform and Hilbert transform, and attractor reconstruction. Certain deterministic non-linear system may show chaotic behaviour. Time series derived from such system seem stochastic when analyzed with linear techniques. However, uncovering the deterministic structure is important because it allows for construction of more realistic and better models and thus improved predictive capabilities. Chaotic behaviour in deterministic dynamical system is an intrinsicly non-linear phenomenon. A characteristic feature of chaotic system is an extreme sensitivity to changes in initial conditions.

It can easily happen that the different forms of nonlinearity can be confusing. Also it can be difficult to distinguish between nonstationarity and nonlinearity. An example in this sense is the following: if the Fisher equation for the United States is estimated, a change in the model in the late 1970s and early 1980 is expected due to the oil price shocks and subsequent Federal Reserve policy. Traditional unit root tests, such as the augmented Dickey-Fuller (Dickey & W.A. (1979);Dickey & W.A. (1981)), the Phillips & Perron (1988), and the (Kwiatkowski et al. (1992)), interpret this change in the model parameters as non-stationarity. Nevertheless, the model has undergone a shift in the parameters before and after the event (oil price shocks) and could very well be stationary if we run the tests in the pre and post event data separately (Ghos & Dutt (2008)).

The choice of a model for a time series is driven by many considerations, often depending on the purpose of research. In most cases, this choice is fundamentally subjective and based on a priori knowledge or expectations of the researcher.

Techniques for vector nonlinear time series modeling have only recently begun to be investigated. Harvill & Ray (1999) provide a general test of nonlinearity in a vector time series. Granger & Teräsvirta (1993) mention multivariate extensions of nonlinear autoregressive (NLAR), nonlinear moving average (NLMA), and bilinear models in passing, but concentrate on statistical inference for univariate nonlinear models. More recent work by Tsay (1998) discusses testing and modeling multivariate threshold autoregressive models.

The multivariate nonlinearity analysis requires more research. In this thesis we develop techniques for analyzing some forms of multivariate nonlinearity in conditional mean. In particular, we dealt with outliers and threshold models in a multivariate framework.

Several papers that generalize the univariate threshold principle to a multivariate framework have appeared in the literature during the past years. Tiao and Tsay (1994) proposed a univariate SETAR model for the United States gross national product (GNP) series where the thresholds are controlled by two lagged values of the transformed GNP series reflecting the situation of the economy. Tsay (1998) developed a strategy for testing and estimating multivariate threshold models where the threshold variable was controlled by known linear combination of individual variables. Arnold and Gunther (2011) proposed a definition of MSETAR models where each linear regime follows a VAR process and the threshold variable is multivariate. Furthermore, they developed an estimation procedure of the corresponding autoregressive (AR) coefficient matrices. However, the authors suppose that the structural parameters of the model (delay, threshold variable, number and position of thresholds, model order) have to be known a priori.

In the present thesis, we adopt a less restrictive formulation, assuming that the structural parameters are unknown and are jointly estimated with the other parameters of the model.We formulate the task of finding the threshold variable and the other structural parameters as a combinatorial optimization problem. We suggested a genetic algorithm-based procedure for identifying and estimating an MSETAR model with univariate or bivariate threshold variable. The procedure uses a special binary encoding composed of several fragments each of which represents an integer parameter of the MSETAR model.

A simulation experiment demonstrated the validity of the genetic algorithms for implementing the identification and estimation procedure for building a nonlinear model in a multivariate setting.

In this context the most important contribution lies in the choice and estimation of structural parameters of the MSETAR model. The choice of these structural parameters is very difficult since it is not possible to make use of the instruments generally used for the choice of the structural parameters of the SETAR models. A wrong choice of structural parameters also affects the overall performance of the model in explaining the dynamics of the multivariate time series and on the forecasting ability of the model. We realized also a GUI program for estimating a MSETAR model. With the program is also possible to estimate SETAR models which are considered as a particular case of a model MSETAR.

Regarding the problem of outlier detection, in the thesis we have been concerned on detecting multiple outliers, especially those occurring close in time, often have severe masking effect (one outlier masks a second outlier) and smearing effect (misspecification of correct data as outliers) that can easily render the iterative outlier detection methods inefficient. A special case of multiple outliers is a patch of additive outliers. For univariate time series this problem has been addressed firstly by Bruce & Martin (1989) and after by Justel et al. (2001). For multivariate time series, only three procedures have been proposed but none of they deal specifically with the problem of consecutive outliers. Tsay et al. (2000) proposed a sequential detection procedure, which we will call the TPP method, based on individual and joint likelihood ratio statistics; this method requires an initial specification of a vector ARMA model. Galeano et al. (2006), Baragona & Battaglia (2007) proposed a method based on univariate outlier detection applied to some useful linear combinations of the vector time series. The optimal combinations are found by projection pursuit in the first paper and independent component analysis (ICA) in the second one.

We propose a class of meta-heuristic algorithms to overcome the difficulties of iterative procedures in detecting multiple additive outliers in multivariate time series. Our procedures are less vulnerable to the masking and smearing effects because they evaluate several outlier pattern where all observations that are possibly outlying ones are simultaneously considered. In this way, meta-heuristic methods deal efficiently the detection of patch of additive outliers. Each outlier configuration is evaluated by a generalised AIC-criterion where the penalty constant is suggested by both a simulation study and a theoretical approximation. The meta-heuristic algorithms used a approximation of multiple linear interpolator given in Rozanov (1957). More precisely, we use an unbiased estimator of the anomalies for any outlier configuration.

The main contribution of this thesis for the problem of outlier detection in multivariate time series is to reduce the limitations of the iterative procedures in the search of consecutive outliers. Moreover, we attempt to provide an approximation of the penalty term of AIC general criterion which is of a paramount importance in the identification of outliers.

The comparison and the performance of the proposed methods are illustrated by simulation studies and real data analysis. Simulation results show that the proposed approaches perform well for detecting consecutive (patches) additive outliers, while TPP method, used as a comparison, show evident limitations in the case of consecutive outliers. These bad results of the TPP method are also justified analytically.

1.2 Multivariate Time Series

A $s-$dimensional vector time series or multivariate time series arise when several related time series, $x_1(t), x_2(t), \ldots, x_s(t)$, are observed simultaneously over time, instead of observing just a single time series as is the case in univariate time series analysis (Reinsel 1993).

Multivariate time series are considerable in a variety of fields such as engineering, physical sciences, particularly earth sciences (e.g., meteorology and geophysics), economics and business (Reinsel 1993). For example, in an engineering context one may be interested in the study of the simultaneous behaviour over time of current and voltage, or of pressure, temperature, and volume, whereas in economics, we may be interested in the variations of interest rates, money supply, unemployment, and so on, or in sales volume, price, and advertising expenditures for a particular commodity in a business context (Reinsel 1993).

Two of the reasons for analyzing and modeling such multiple time series jointly are:

1. To understand the dynamic relationships among them. They may be contemporaneously related, one series may lead the others or there may be feedback relationships.

2. To improve accuracy of forecasts. When there is information on one series contained in the historical data of another, better forecasts can result when the series are modeled jointly.

Models that are of possible use in representing such multiple time series, considerations of their properties, and methods for relating them to actual data have been extensively discussed in the literature. Quenouille (1957), Whittle (1963), Hannan (1970), Brillinger (1975), Lütkepohl (1993), Hamilton (1994), Reinsel (1993) are just some of the many that have studied and made contribution to the fields of multivariate time series analysis.

1.3 Some Basics

1.3.1 Random Variable

Univariate Real Random Variable. Let $(\Omega, \mathcal{A}, P)$ be a probability space, where Ω is the set of elementary events (sample space), $\mathcal{A}$ is a sigma-algebra of events or subsets of Ω and P is a probability measure defined on $\mathcal{A}$. A random variable X is a mapping from the sample space Ω onto the real line $\mathbb{R}$ such that to each element $\omega \in \Omega$ there corresponds a unique real number, $X(\omega)$. We denote the mean of X with $\mu_X = \mathbb{E}(X)$, the variance of X with $Var(X) = \mathbb{E}[(X - \mu_X)^2]$, and the covariance between X and Y with $cov(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$.

Univariate Complex Random Variables. A complex random variable X is defined as a random variable of the form $X = X_R + iX_I$, where the real and imaginary parts, X_R, and X_I, are real random variables and $i = \sqrt{-1}$. The expectation of real random variable is naturally generalized to the complex case as $\mu_X = \mathbb{E}(X) = \mathbb{E}(X_R) + i\mathbb{E}(X_I) = \mu_{X_R} + \mu_{X_I}$. The variance of X is equal to $Var(X) = \mathbb{E}[|(X - \mu_X)|^2]$ while the covariance between X and Y is defined as $cov(X, Y) = \mathbb{E}[(X - \mu_X)\overline{(Y - \mu_Y)}]$.

Vector of Real Random Variable. A $s-$dimensional random vector variable $\mathbf{X} = [X_1, X_2, \ldots, X_s]'$ is a function from Ω into the $s-$dimensional Euclidean space $\mathbb{R}^s$ such that to each element $\omega \in \Omega$ there corresponds a unique vector, $\mathbf{X}(\omega)$. Mean vector of $\mathbf{X}$ is the column vector of the means of each component $\mu = \mathbb{E}(\mathbf{X}) = [\mathbb{E}(X_1), \mathbb{E}(X_2), \ldots, \mathbb{E}(X_s)]'$. The covariance matrix is defined as $\mathbf{\Sigma} = \mathbb{E}[(\mathbf{X} - \mu)(\mathbf{X} - \mu)']$.

Vector of Complex Random Variable. A $s-$dimensional complex random vector variable $\mathbf{X} = [X_1, X_2, \ldots, X_s]'$ is defined as a vector random variable of the form $\mathbf{X} = \mathbf{X_R} + i\mathbf{X_I}$, where the real and imaginary parts, X_R, and X_I, are $s-$dimensional real random vector variable. Mean vector of $\mathbf{X}$ is defined by $\mu = \mathbb{E}(\mathbf{X_R}) + \mathbb{E}(\mathbf{X_I})$. The covariance matrix is defined as $\mathbf{\Sigma} = \mathbb{E}[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^*]$.

1.3.2 Multivariate Stochastic Process

A $s-$dimensional vector stochastic process or multivariate stochastic process $\mathbf{X}(t) = [X_1(t), X_2(t), \ldots, X_s(t)]'$, is a family of random variables indexed by the symbol t, where t belongs to some given index set, T. If t takes a continuous range of real values (finite or infinite), so that $\mathbf{X}(t)$ is said to be a continuous parameter process. If t takes a discrete set of values, typically, $t = 0, \pm 1, \pm 2, \ldots$, then $\mathbf{X}(t)$ is said to be a discrete parameter process. Alternatively, and in an equivalent way, an $s-$dimensional vector stochastic process may be thought as a function $\mathbf{X}(t, \omega) : T \times \Omega \to \mathbb{R}^s$, where for each fixed $t \in T$, $\mathbf{X}(t, \omega)$ is a $s-$dimensional random vector variable.

A realization of a vector stochastic process is a sequence of vectors $\mathbf{X}(t, \omega)$, $t \in T$, for a fixed ω. In other word a realization of a stochastic process is a function $\mathbf{X}(t, \bullet) : T \to \mathbb{R}^s$. A multiple time series is regarded as such a finite part of a realization, that is, it consist, for example, of values vectors $x_1(\omega), x_2(\omega), \ldots, x_N(\omega)$. The underlying stochastic process is said to have generated the multiple time series or it is called the generating or generation process of time series. A multiple time series $x_1(\omega), x_2(\omega), \ldots, x_N(\omega)$ will be denoted by $x_1, x_2, \ldots, x_N$. The number of observation N is called the sample size or time series length.

Stationary Multivariate Processes

An important concept in the representation of models and analysis of time series, which enables useful modeling results to be obtained from a finite sample realization of the time series, is that of stationarity.

An s vector-valued process $\mathbf{X}(t)$ is *strongly stationary* if the probability distributions of the random vectors $[X(t_1), X(t_2), \ldots, X(t_n)]$ and $[X(t_1 + l), X(t_2 + l), \ldots, X(t_n + l)]$ are the same for arbitrary times $t_1, t_2, \ldots, t_n$, all n and all lags or leads $l = \pm 1, \pm 2, \ldots$. Thus, the probability distribution of observations from stationary vector process is invariant with respect to shift in time. An example of strictly stationary process is a process of independent identically distributed s vector-valued variates with mean vector 0 and covariance matrix equal to I_s. This process is called *strong sense white noise* and is denoted by $\mathbf{e}(t)$.

An s vector-valued process $\mathbf{X}(t)$ is *weakly* or second order stationary if the process possesses finite first and second moments and which satisfies the condition that mean does not depend on t and covariance depends only on lag u:

1. $E[\mathbf{X}(t)] = \mu = (\mu_1, \mu_2, \ldots, \mu_s)', \quad \forall t$

2. $E\{[\mathbf{X}(t) - \mu][\mathbf{X}(t+u) - \mu]'\} = \Gamma(u), \quad \forall t$

Covariance Matrices for a Stationary Vector Process

If we have an s vector-valued process $\mathbf{X}(t)$ with $\mu = 0$, we define the covariance matrix at lag u by:

$$\mathbf{\Gamma}(u) = E\{[\mathbf{X}(t+u)][\mathbf{X}(t)]'\} = \begin{bmatrix} \gamma_{11}(u) & \gamma_{12}(u) & \ldots & \gamma_{1s}(u) \\ \gamma_{21}(u) & \gamma_{22}(u) & \ldots & \gamma_{2s}(u) \\ \ldots & \ldots & \ldots & \ldots \\ \gamma_{s1}(u) & \gamma_{s2}(u) & \ldots & \gamma_{ss}(u) \end{bmatrix} \tag{1.8}$$

For $i \neq j$, $\gamma_{ij}(u) = \mathbb{E}[X_j(t+u)X_i(t)]$ denotes the cross-covariance function between $X_i(t)$ and $X_j(t+u)$, while for $i = j$, $\gamma_{ii}(u)$ denotes the autocovariance function of $X_i(t)$ that depend only on lag u, not on time t, for $i, j = 1, \ldots, s$, $u = 0, \pm1, \pm2, \ldots$.

In this thesis, the term stationary will generally be used in sense of weak stationarity. For a stationarity vector process, the cross-covariance matrix structure provides a useful summary of information on aspects of dynamic interrelations among the components of the process. However, because of higher dimensionality of the vector process, the cross-covariance matrices can generally take on complex structures and may be much more difficult to interpret as a whole as compared with the univariate time series case.

Complex valued multivariate process

So far we have discussed only real valued processes, i.e. processes which at each time point, assume real values. Although, of course, processes which arise in practice are all real valued it is nevertheless convenient sometimes regard them as complex valued, just as in eletrical circuit theory it is sometimes convenient to regard a voltage as a complex variable.

A complex valued process may be defined as a sequence of complex random variable indexed by the symbol t, where $t \in T$: $\mathbf{X}(t) = \mathbf{U}(t) + i\mathbf{V}(t)$ where $\mathbf{U}(t), \mathbf{V}(t)$ are both real valued process. If we suppose that $\mathbf{X}(t)$ is stationary up to order 2, then the mean of $X(t)$ is defined by:

$$\mathbb{E}[\mathbf{X}(t)] = \mathbb{E}[\mathbf{U}(t)] + i\mathbb{E}[\mathbf{V}(t)] = \mu \text{ a constant vector independent of t} \tag{1.9}$$

The covariance matrix $\mathbf{X}(t)$ is defined by (if we suppose that $\mu = 0$):

$$\Gamma(u) = \mathbb{E}\{[\mathbf{X}(t+u)][\mathbf{X}(t)]^*\} \tag{1.10}$$

where $\gamma_{ij}(u) = \mathbb{E}[X_i(t+u)\overline{X_j(t)}]$

Spectral property for a Stationary Vector Process

Spectral Density Matrix. Similar to the univariate case we define the spectral density matrix of the stationary vector process $\mathbf{X}(t)$ as:

$$\mathbf{f}(\lambda) = (2\pi)^{-1} \sum_{u=-\infty}^{\infty} \mathbf{\Gamma}(u)\text{exp}(-i\lambda u), \qquad -\pi < \lambda < \pi \tag{1.11}$$

Then $\mathbf{f}(\lambda)$ is the Fourier transform of the covariance matrix function. The (i,j)th element of the matrix $\mathbf{f}(\lambda)$ denoted as $f_{ij}(\lambda)$ is:

$$f_{ij}(\lambda) = (2\pi)^{-1} \sum_{u=-\infty}^{\infty} \gamma_{ij}(u)\text{exp}(-i\lambda u)$$

For $i = j$, $f_{ii}(\lambda)$ is the spectral density function of the process $X_i(t)$ and is the Fourier transform of the auto-covariance function $\gamma_{ii(u)}$, while for $i \neq j$, $f_{ij}(\lambda)$ is the cross-spectral density function between the process $X_i(t)$ and $X_j(t)$, that is, the Fourier transform of the cross-covariance function $\gamma_{ij}(u)$.

Notice that $f_{ii}(\lambda)$ is real-valued and non-negative, but since $\gamma_{ij}(u) \neq \gamma_{ij}(-u)$ for $i \neq j$, the cross-spectral density function $f_{ij}(\lambda)$ is in general complex-valued with $f_{ij}(\lambda)$ begin equal to $\overline{f_{ji}}(\lambda) = f_{ji}(-\lambda)$, the complex conjugate of $f_{ij}(\lambda)$. Therefore, the spectral density matrix $\mathbf{f}(\lambda)$ is Hermitian, that is, $\mathbf{f}^*(\lambda) = \mathbf{f}(\lambda)$. Moreover, $\mathbf{f}(\lambda)$ is a non-negative definite matrix in the sense that $b'\mathbf{f}(\lambda)b \geq 0$ for any $s-$dimensional vector b, since $b'\mathbf{f}(\lambda)b$ is the spectral density function of a linear combination $b'X(t)$ and hence must be non-negative.

Spectral Representations Let $\mathbf{X}(t)$ be a zero mean $s-$dimensional stationary vector process. Then exists a $s-$dimensional complex-valued continuous-parameter process, $\mathbf{Z}(\lambda) = [Z_1(\lambda), Z_2(\lambda), \ldots, Z_s(\lambda)]$, defined on the interval $[-\pi, \pi]$ such that for all integer t (Rozanov (1957); pag 18):

$$X(t) = \int_{-\pi}^{\pi} e^{i\lambda t} dZ(\lambda) \quad \text{or} \quad X_i(t) = \int_{-\pi}^{\pi} e^{i\lambda t} dZ_i(\lambda) \tag{1.12}$$

where the column vector $d\mathbf{Z}(\lambda)$ has elements $dZ_1(\lambda), dZ_2(\lambda), \ldots, dZ_s(\lambda)$. The representation (1.12) is called *spectral representation* of the multivariate stationary process $\mathbf{X}(t)$.

The $s-$dimensional random process, $\mathbf{Z}(\lambda)$, also called *random spectral measure* of $s-$dimensional process $\mathbf{X}(t)$, has the following properties:

1. $\mathbb{E}[d\mathbf{Z}(\lambda_1)d\mathbf{Z}^*(\lambda_2)] = 0$ if $\lambda_1 \neq \lambda_2$,
2. $\mathbb{E}[dZ_i(\lambda_1)\overline{dZ_j(\lambda_2)}] = 0 \; \forall i, j = 1, 2, \ldots, s$ if $\lambda_1 \neq \lambda_2$,
3. $\mathbb{E}[d\mathbf{Z}(\lambda)d\mathbf{Z}^*(\lambda)] = \mathbf{f}(\lambda)d\lambda$

Hence, properties (1) and (2) show that $dZ_1(\lambda), dZ_2(\lambda), \ldots, dZ_s(\lambda)$ are not only orthogonal but also *cross-orthogonal*. From property (3) we have:

$$\begin{aligned} f_{ii}(\lambda)d\lambda &= \mathbb{E}[dZ_i(\lambda)\overline{dZ_i(\lambda)}] = \mathbb{E}[|dZ_i(\lambda)|^2], \\ f_{ij}(\lambda)d\lambda &= \mathbb{E}[dZ_i(\lambda)\overline{dZ_j(\lambda)}] \; i \neq j \end{aligned} \tag{1.13}$$

Hence, $\mathbf{f}(\lambda)d\lambda$ represents the covariance matrix of $d\mathbf{Z}(\lambda)$,the random vector at frequency λ in the spectral representation of the vector process $\mathbf{X}(t)$. That is, $f_{ii}(\lambda)d\lambda$ represent the variance of $dZ_i(\lambda)$ and $f_{ij}(\lambda)d\lambda$ represent the covariance between $dZ_i(\lambda)$ and $dZ_j(\lambda)$. Alternatively, we may say that, whereas $f_{ii}(\lambda)d\lambda$ represents the average value of the square of the coefficient of $e^{i\lambda t}$, $f_{ij}(\lambda)d\lambda$ represents the average value of the product of the coefficients of $e^{i\lambda t}$ in $X_i(t)$ and $X_j(t)$.

We can note also that substituting (1.12) in (1.8) the spectral representation of the covariance matrix function is:

$$\begin{aligned} \mathbf{\Gamma}(u) &= \int_{-\pi}^{\pi} e^{-i\lambda t} e^{-i\lambda^{'}(t+u)} \mathbb{E}[d\mathbf{Z}(\lambda)d\mathbf{Z}^*(\lambda^{'})] \\ &= \int_{-\pi}^{\pi} e^{-i\lambda u} \mathbb{E}[d\mathbf{Z}(\lambda)d\mathbf{Z}^*(\lambda)] \\ &= \int_{-\pi}^{\pi} e^{-i\lambda u} d\mathbf{H}(\lambda) \end{aligned} \tag{1.14}$$

that is:

$$\begin{aligned}\gamma_{ij}(u) &= \int_{-\pi}^{\pi} e^{-i\lambda t}e^{-i\lambda'(t+u)}\mathbb{E}[dZ_i(\lambda)\overline{dZ_j(\lambda')}] \\ &= \int_{-\pi}^{\pi} e^{-i\lambda u}\mathbb{E}[dZ_i(\lambda)\overline{dZ_j(\lambda)}] \\ &= \int_{-\pi}^{\pi} e^{-i\lambda u}dH_{ij}(\lambda)\end{aligned} \tag{1.15}$$

where:

$$\begin{aligned}dH_{ij}(\lambda) &= \mathbb{E}[dZ_i(\lambda)\overline{dZ_j(\lambda)}] = f_{ij}(\lambda)d\lambda,\ i \neq j, \\ dH_{ii}(\lambda) &= \mathbb{E}[|dZ_i(\lambda)|^2] = f_{ii}(\lambda)d\lambda,\end{aligned} \tag{1.16}$$

The matrix $\mathbf{H}(\lambda)$ is called *spectral distribution matrix.* The diagonal elements $H_{ii}(\lambda)$ are the integrated spectra of the process $X_i(t)$, while $H_{ij}(\lambda)$ is the integrated cross-spectrum between $X_i(t)$ and $X_j(t)$.

Substituting equations(1.16) in (1.15) obtained,

$$\gamma_{ij}(u) = \int_{-\pi}^{\pi} f_{ij}(\lambda)e^{-i\lambda u}d\lambda \quad u = \pm 1, \pm 2, \ldots \tag{1.17}$$

that may be written more concisely in the form:

$$\mathbf{\Gamma}(u) = \int_{-\pi}^{\pi} \mathbf{f}(\lambda)e^{-i\lambda u}d\lambda \quad u = \pm 1, \pm 2, \ldots \tag{1.18}$$

In some texts the spectrum is defined using the covariance matrix generating function, which is a power series with complex terms. The covariance matrix generating function $F(z)$ (where z is a complex number) is defined by:

$$\mathbf{F}(z) = \sum_{u=-\infty}^{\infty} \mathbf{\Gamma}(u)z^u \tag{1.19}$$

The covariance matrix generating function coincides with the spectral density matrix $\mathbf{f}(\lambda)$ if $z = e^{i\lambda}$: $\mathbf{F}(z) = \mathbf{f}(\lambda)$.

1.3.3 Linear Filtering of a Stationary Vector Process

Fundamental to the study of multivariate linear system of stochastic process is the representation of dynamic linear relationship through the formulation of linear filters. A multivariate linear (time-invariant) filter relating an $r-$dimensional input stochastic process $X(t)$ to a $s-$dimensional output stochastic process $Y(t)$ is given by the form:

$$\mathbf{Y}(t) = \sum_{u=-\infty}^{\infty} \mathbf{\Psi}(u)\mathbf{X}(t-u) \tag{1.20}$$

where $\mathbf{Y}(t)$ and $\mathbf{X}(t)$ are column vectors, the $\mathbf{\Psi}(u)$ are $s \times s$ matrices, and $\{\mathbf{\Psi}(u)\}, u = 0, \pm 1, \pm 2, \ldots,$ are called the *impulse response matrices.* From (1.20) we may write the ith output as:

$$Y_i(t) = \sum_{u=-\infty}^{\infty} \Psi_{i1}(u)X_1(t-u) + \ldots + \sum_{u=-\infty}^{\infty} \Psi_{ir}(u)X_r(t-u),\ i = 1, \ldots, s \tag{1.21}$$

The filter is physically realizable or causal when the $\mathbf{\Psi}(u) = 0$ for $u < 0$, so that $\sum_{u=0}^{\infty} \mathbf{\Psi}(u)\mathbf{X}(t-u)$ is expressible in terms of only present and past values of the input process $\mathbf{X}(t)$. The filter is said to be *stable* if $\sum_{u=-\infty}^{\infty} \|\mathbf{\Psi}(u)\| < \infty$, where $\|A\|$ denotes a norm for the matrix A such as $\|A\|^2 = tr\{A'A\}$.

When the filter is stable and the input process $\mathbf{X}(t)$ is stationary with covariance matrices $\mathbf{\Gamma}_x(u)$, the output process $\mathbf{Y}(t) = \sum_{u=-\infty}^{\infty} \mathbf{\Psi}(u)\mathbf{X}(t-u)$ is a stationary process.

Introducing the spectral representation:

$$X_i(t) = \int_{-\pi}^{\pi} e^{i\lambda t} dZ_i^{(x)}(\lambda),\ i = 1, \ldots, r \tag{1.22}$$

$$Y_j(t) = \int_{-\pi}^{\pi} e^{j\lambda t} dZ_j^{(y)}(\lambda),\ j = 1, \ldots, s \tag{1.23}$$

the jth terms of (1.21) can be written as:

$$\int_{-\pi}^{\pi} e^{i\lambda t} G_{ij}(\lambda) dZ_i^{(x)}(\lambda), \tag{1.24}$$

where $G_{ij}(\lambda) = \sum_u \psi_{ij}(u)e^{-i\lambda u}$ represents the *transfer function* between the ith input and the jth output.

Equation (1.21) now gives, for each λ,:

$$dZ_j^{(y)}(\lambda) = G_{j1}(\lambda)dZ_1^{(x)}(\lambda) + \ldots + G_{jr}(\lambda)dZ_r^{(x)}(\lambda),\ j = 1, \ldots, s \tag{1.25}$$

This equation is of considerable importance. In the *time domain* description (1.21), the relationship between the jth output at time t involves weighted linear combination of past, present and future values of all the input processes. However, the *frequency domain* form (1.25) has a much simpler structure. In fact (1.25) is simply the classical multiple linear regression model, and, as in the single input/single output case, has the feature that the spectral proprieties of the output at frequency λ depend only on the spectral properties of the input at the same frequency λ. Writing (1.25) in matrix form we have:

$$d\mathbf{Z}^{(y)}(\lambda) = \mathbf{G}(\lambda) d\mathbf{Z}^{(x)}(\lambda) \tag{1.26}$$

where the $(s \times s)$ square matrix $\mathbf{G}(\lambda) = \sum_u \mathbf{\Psi}(u)e^{-i\lambda u}$ is called the *transfer function matrix.* The system is thus described completely by the transfer function matrix $\mathbf{G}(\lambda)$ which, when written out in full, takes the form:

$$\mathbf{G}(\lambda) = \begin{bmatrix} G_{11}(\lambda) & G_{12}(\lambda) & ... & G_{1s}(\lambda) \\ G_{21}(\lambda) & G_{22}(\lambda) & ... & G_{2s}(\lambda) \\ ... & ... & ... & ... \\ G_{r1}(\lambda) & G_{s2}(\lambda) & ... & G_{rs}(\lambda) \end{bmatrix}$$

where the entry in the ith row and jth column being the transfer function relating the ith input to the jth output. Equation (1.26) gives us immediately the relationship between the spectral matrices of the input and output. For we have:

$$\mathbb{E}[d\mathbf{Z}^{(y)}(\lambda) d\mathbf{Z}^{(y)*}(\lambda)] = \mathbf{G}(\lambda)\mathbb{E}[d\mathbf{Z}^{(x)}(\lambda) d\mathbf{Z}^{(x)*}(\lambda)]\mathbf{G}^*(\lambda) \tag{1.27}$$

which, on using property (3) of random spectral measure, the spectral density matrix of output process $Y(t)$, $\mathbf{f}_y(\lambda)$, is:

$$\mathbf{f}_y(\lambda) = \mathbf{G}(\lambda)\mathbf{f}_x(\lambda)\mathbf{G}^*(\lambda) \tag{1.28}$$

where $\mathbf{f}_x(\lambda)$ is the spectral density matrix of input process $\mathbf{X}(t)$.

Noting that the variance of $Y_j(t)$ is given by integrating the jth diagonal element of $\mathbf{f}_x(\lambda)$, the condition for each output to have finite variance is:

$$tr\{\int_{-\pi}^{\pi} \mathbf{G}(\lambda)\mathbf{f}_x(\lambda)\mathbf{G}^*(\lambda) d\lambda\} < \infty \tag{1.29}$$

where, for any square matrix $\mathbf{A}$, $tr(\mathbf{A})$ denotes the *trace* of $\mathbf{A}$, namely, the sum of the diagonal elements of $\mathbf{A}$.

The covariance matrices of the stationary process $Y(t)$ are given by:

$$\mathbf{\Gamma}_y(u) = \mathbb{E}[\mathbf{Y}(t), \mathbf{Y}^*(t+u)] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \mathbf{\Psi}(i)\mathbf{\Gamma}_x(u+i-j)\mathbf{\Psi}^*(j). \tag{1.30}$$

In Reinsel (1993)) the spectral density matrix of the output $Y(t)$ has the representation:

$$\mathbf{f}_y(\lambda) = \mathbf{G}(e^{i\lambda})\mathbf{f}_x(\lambda)\mathbf{G}^*(e^{-i\lambda})$$

,

where the transfer function (matrix) of the linear filter is defined as $G(z) = \sum_{j=-\infty}^{\infty} \Psi(j)z^j$.

Inverse covariance matrix and inverse process

Inverse covariances and inverse process of a stationary multivariate stochastic process have been defined independently and contemporaneously by Battaglia (1984) and Vitale (1984), one moving from frequency domain and one from time domain. The two definitions coincide. The inverse covariance can also play a role in the analysis of relationships between the components of a multivariate series.

Let $\mathbf{X}(t) = [X_1(t), X_2(t), \ldots, X_s(t)]'$ be a discrete-parameter s-variate second-order stationary process with mean zero for each component and covariance matrix $\mathbf{\Gamma}(h)$ defined in (1.8). We suppose that $\mathbf{X}(t)$ has absolutely continuous spectrum and for each λ, the inverse of spectral density matrix $\mathbf{f}(\lambda)$ defined in (1.11)(Battaglia (1984), pag 118) exists and is integrable. Then we define the matrices of inverse covariance $\mathbf{\Gamma}_\mathbf{i}(h)(h = 0, \pm 1, \pm 2, \ldots)$ by:

$$\mathbf{\Gamma}_\mathbf{i}(u) = \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi} \mathbf{f}^{-1}(\lambda)e^{i\lambda u}d\lambda = \begin{bmatrix} \gamma i_{11}(u) & \gamma i_{12}(u) & ... & \gamma i_{1s}(u) \\ \gamma i_{21}(u) & \gamma i_{22}(u) & ... & \gamma i_{2s}(u) \\ ... & ... & ... & ... \\ \gamma i_{s1}(u) & \gamma i_{s2}(u) & ... & \gamma i_{ss}(u) \end{bmatrix} \tag{1.31}$$

so that:

$$\mathbf{f}^{-1}(\lambda) = 2\pi \sum_{u=-\infty}^{\infty} \mathbf{\Gamma}_\mathbf{i}(u)e^{-i\lambda u} = \begin{bmatrix} p_{11}(u) & p_{12}(u) & ... & p_{1s}(u) \\ p_{21}(u) & p_{22}(u) & ... & p_{2s}(u) \\ ... & ... & ... & ... \\ p_{s1}(u) & p_{s2}(u) & ... & p_{ss}(u) \end{bmatrix} \tag{1.32}$$

As $\mathbf{f}^{-1}(\lambda)$ is also Hermitian for each λ, we have $\mathbf{\Gamma_i}(h) = \mathbf{\Gamma_i}(-h)'$.

For inverse covariance matrices an orthogonality relation may be derived in the same way as in the univariate case (Battaglia 1983). In fact, using (1.31) and the analogous spectral representation (1.18), it is easily seen that:

$$\sum_{u=-\infty}^{\infty} \mathbf{\Gamma_i}(u)\mathbf{\Gamma}'(u+k) = \delta_k I_s \tag{1.33}$$

where δ_k denotes Kronecker's delta.

Further we define the *inverse process* of $\mathbf{X}(t)$ as a linear filter with weights equal to the inverse covariances:

$$\mathbf{Z}(t) = 2\pi \sum_{u=-\infty}^{\infty} \mathbf{\Gamma_i}(u)\mathbf{X}(t-u). \tag{1.34}$$

Using (1.33) it may be verified that $\mathbf{Z}(t)$ is a second-order stationary process with mean zero and covariance matrix equal to the inverse covariance matrix of $\mathbf{X}(t)$:

$$\mathbb{E}[\mathbf{Z}(t)\mathbf{Z}^*(t+u)] = \mathbf{\Gamma_i}(u). \tag{1.35}$$

In addition, the covariances between the components of the process and the component of its inverse process is provided by:

$$\mathbb{E}[\mathbf{X}(t)\mathbf{Z}^*(t+u)] = \sum_u \mathbf{\Gamma}(u)\mathbf{\Gamma_i}(u+h) = \delta_k I_s. \tag{1.36}$$

Thus, the components of $\mathbf{X}(t)$ are uncorrelated with the non-homologous components of $\mathbf{Z}(t)$ for each lag, while the homologous components of the two processes are contemporaneously correlated, but uncorrelated when lagged.

We may use two different ways to estimate the inverse covariance matrix. A first approach is based on the estimation of the spectral density matrix and the Fourier transform of its inverse (Battaglia (1984)). The second one fits a high-order vector autoregressive model to the data and derives estimates of the inverse covariance matrix from the estimated parameters of the model (Battaglia (1984)). Bhansali (1980) has shown that under reasonable regularity conditions both methods give consistent and asymptotically Gaussian estimates. We reported here the second approach where the estimates of the inverse covariance are obtained as follows:

$$\hat{\mathbf{\Gamma}}\mathbf{i}_u = \begin{cases} \sum_{j=u}^{m} \hat{\mathbf{\Phi}}'_{j-u}\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{\Phi}}_j & 0 \leq u \leq m \\ \mathbf{0} & u > m \\ \hat{\mathbf{\Gamma}}\mathbf{i}'_{-u} & u < 0 \end{cases} \tag{1.37}$$

where $\hat{\mathbf{\Phi}}_1, \hat{\mathbf{\Phi}}_2, \ldots, \hat{\mathbf{\Phi}}_m$ are the least squares estimates of the parameter matrices of the VAR(m) model, $\hat{\mathbf{\Sigma}}$ is the estimated variance matrix of the noise and where we set $\hat{\mathbf{\Phi}}_0 = -\mathbf{I}$.

Space of values of a stationary vector process

Let $X(t) = [X_1(t), X_2(t), \ldots, X_s(t)]'$ be an $s-$dimensional stationary process, and H_x be the linear manifold spanned by variables $X_k(t)$,$k = 1, \ldots, s$, $-\infty < t < \infty$, closed with respect to convergence in mean square. This space with scalar product (Rozanov (1957), pag 3):

$$(X_i(t), X_j(t)) = \mathbb{E}[X_i(t)\overline{X_j(t)}] \ \forall i, j = 1, \ldots, s, \ t \in \mathbb{Z} \tag{1.38}$$

is a Hilbert space; we will call it the *space of values* of the process $X(t)$.

We can demonstrate that for any element $h \in H_x$ there exist a vector function $\varphi(\lambda) = [\varphi_1(\lambda), \ldots, \varphi_s(\lambda)]'$ belonging to $L^2(F)$ such that h is representing in the form of integral with respect to the random spectral measure $Z(\lambda)$:

$$h = \int \varphi(\lambda) dZ(\lambda) = \int \sum_{k=1}^{s} \varphi_k(\lambda) dZ_k(\lambda) \tag{1.39}$$

We will call the vector function $\varphi(\lambda)$ the *spectral characteristic* of the random variable h.

We will say that $\varphi(\lambda)$ belongs to the space $L^2(F)$, if the function:

$$\varphi(\lambda)\mathbf{f}(\lambda)\varphi^*(\lambda) = \sum_{k,l=1}^{s} \varphi_k(\lambda)\overline{\varphi_k(\lambda)} f_{kl}(\lambda) \tag{1.40}$$

is integrable.

Minimal Process

Theorem 1. . *In order that an n-dimensional stationary process $X(t)$ with spectral density f be minimal, it is necessary and sufficient that:*

$$\int_{-\pi}^{\pi} tr(f^{-1}(\lambda))d\lambda < \infty.$$

where *tr* denotes the *trace* of a matrix.

1.4 Linear Interpolation of Stationary Vector Process

An important problem in the theory of s-variate ($s \geq 1$) weakly stationary stochastic process $\mathbf{X}(t)$ is to obtain formulas for linear interpolator and interpolation error matrix. This problem seem to have potential application to many different areas of physical, natural and social sciences, that is in the cases where the values of a stochastic process that represent a particular phenomena either are missing at some points or it is not possible to obtain direct measurement at these points. This problem has generated a rather extensive literature beginning with Kolmogorov's fundamental article (Kolmogorov 1941).

Masani (1960) considered a full-rank minimal $s - variate$ process (the missing value is at one point) over $\mathbb{Z}$ and obtained an explicit expression, for the interpolation error matrix in terms of spectral density of the process, thereby extending the $s = 1$ result due to Kolmogorov (1941).

There are a number of different proof of linear interpolation of a stationary vector process, some of which revealed interesting relationship between the spectral theory of stationary vector process and other branches of pure mathematics. Explicit expressions for linear interpolator and interpolation error matrix were obtained by (Rozanov (1957); pag 100-101) using elaborated Fourier and Harmonic analysis techniques. Rozanov's procedure considerate also the case of partially missing observations of the process $\mathbf{X}(t)$. Exact formulas are also given in Battaglia (1984) and Hannan (1970). All formulas suppose that the complete past and the complete future of the stationary process $\mathbf{X}(t)$ are known. We now give a brief sketch of these alternative proofs.

1.4.1 Geometrical Approach to Interpolation

Let $\mathbf{X}(t) = [X_1(t), X_2(t), \ldots, X_s(t)]'$ be a discrete-parameter s-variate second-order stationary process with mean zero for each component, $t \in \mathbb{Z} = [0, \pm 1, \ldots]$. We suppose that $\mathbf{X}(t)$ has absolutely continuous spectrum and for each λ, the inverse of spectral density matrix $\mathbf{f}(\lambda)$ exists and is integrable.

Let $T_k, k = 1, \ldots, s$, be finite subsets of the set of all integers $\mathbb{Z}$. We suppose that all the values $X_k(t)$ of the $s-$dimensional stationary process $\mathbf{X}(t)$ are known, except for the values $X_k(t)$, $t \in T_k, k = 1, \ldots, s$, and it is required to *interpolate* the unknown values $X_k(t)$.

If we measure the error in terms of mean square deviation, the best linear method of interpolation consists in finding the projections of the $X_k(t)$, $t \in T_k$, on closed linear manifold generated by the known variables $X_k(t)$, $t \notin T_k$, $k = 1, \ldots, s$, which we denote by $\bar{H}(T)$.

Let A be $s-$dimensional vector space, and B_λ the subspace of A consisting of all vectors $\mathbf{b} = \{b_k(\lambda)\}$ of the form:

$$\mathbf{b} = \mathbf{a}\mathbf{f}(\lambda) \quad \mathbf{a} \in A \tag{1.41}$$

By the expression $\mathbf{b}\mathbf{f}^{-1}(\lambda)$ for $\mathbf{b} \in B_\lambda$, we will understand any of the vectors $a \in A$ satisfying 1.41.

Obviously, if two vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ lead to the same element $\mathbf{b}$ in (1.41), then:

$$\mathbf{a}_1(\mathbf{b}')^* = \mathbf{a}_2(\mathbf{b}')^* \tag{1.42}$$

for any $\mathbf{b}^{'} = \mathbf{a}^{'}\mathbf{f}(\lambda) \in B_\lambda$, since, by virtue of self-adjointness of the matrix $\mathbf{f}(\lambda)$,

$$(\mathbf{a}_1 - \mathbf{a}_2)(\mathbf{b}')^* = (\mathbf{a}_1 - \mathbf{a}_2)[\mathbf{a}^{'}\mathbf{f}(\lambda)]^* = [(\mathbf{a}_1 - \mathbf{a}_2)\mathbf{f}(\lambda)](\mathbf{a}^{'})^* = (\mathbf{b} - \mathbf{b})(\mathbf{a}^{'})^* \tag{1.43}$$

We define $B(T)$ as the space of vector functions $\mathbf{b}(\lambda) = \{b_k(\lambda)\}$ whose components $b_k(\lambda)$ are trigonometric polynomials of the form:

$$b_k(\lambda) = \sum_{t \in T_k} a_k(t) e^{i\lambda t} \tag{1.44}$$

such that $b(\lambda) \in B_\lambda$ for almost all λ, and such that $\|\mathbf{b}\| = (\mathbf{b}, \mathbf{b})^{1/2} < \infty$, where $(\mathbf{b}, \mathbf{b}^{'})$ is a scalar product in $B(T)$ defined by:

$$(\mathbf{b},\mathbf{b}^{'}) = \int_{-\pi}^{\pi} [\mathbf{b}(\lambda)\mathbf{f}^{-1}(\lambda)(\mathbf{b}')^{*}]d\lambda \tag{1.45}$$

We denote by $\Delta(T)$ the subspace in H_x spanned by the difference :

$$X_k(t) - \hat{X}_k(t), \quad t \in T_k \quad k = 1, 2, \ldots, s \tag{1.46}$$

where $\hat{X}_k(t)$ is the projection of $X_k(t)$ on $\bar{H}(T)$.

Lemma 1. . *The subspace $\Delta(T)$ is isometrically isomorphic to the space $B(T)$ of vector functions.*

Proof. Let $\mathbf{Z}(\lambda) = [Z_1(\lambda), Z_2(\lambda), \ldots, Z_s(\lambda)]^{'}$ be random spectral measure of $\mathbf{X}(t)$. The elements h of the subspace $\Delta(T)$ can be represented in the form:

$$h = \int_{-\pi}^{\pi} \varphi(\lambda) d\mathbf{Z}(\lambda), \tag{1.47}$$

where the vectors function $\varphi = \{\varphi_k\}$ belongs to the space $L^2(F)$, i.e.,

$$\int_{-\pi}^{\pi} \varphi \mathbf{f} \varphi^{*} d\lambda < \infty. \tag{1.48}$$

The orthogonality of h to the subspace $\bar{H}(T)$ means that:

$$\mathbb{E}[h\bar{X}_l(t)] = \int_{-\pi}^{\pi} e^{-i\lambda t} \sum_{k=1}^{s} [\varphi_k(\lambda) f_{kl}(\lambda)] d\lambda = 0, \tag{1.49}$$

for all l and t ($l = 1, \ldots, s, -\infty < t < \infty$) except for $t \in T_l$. If we put $\mathbf{b}(\lambda) = \varphi(\lambda)\mathbf{f}(\lambda)$ the (1.49) shows that the vector function $\mathbf{b}(\lambda) = \{b_k(\lambda)\}$ belongs to the space $B(T)$:

$$b_k(\lambda) = \sum_{l=1}^{s} \varphi_l(\lambda) f_{lk}(\lambda) = \sum_{t \in T_k} a_k(t) e^{-i\lambda t}, \quad k = 1, \ldots, s \tag{1.50}$$

$$\|\mathbf{b}\|^2 = \int_{-\pi}^{\pi} \mathbf{b}(\lambda)\mathbf{f}^{-1}(\lambda)\mathbf{b}^{*}(\lambda) d\lambda = \int_{-\pi}^{\pi} \varphi(\lambda)\mathbf{f}(\lambda)\varphi^{*}(\lambda) d\lambda = \mathbb{E}\,|h|^2 \tag{1.51}$$

On the other hand, if one takes an arbitrary vector function $\mathbf{b}(\lambda)$ from $B(T)$ and sets $\varphi(\lambda) = \mathbf{b}(\lambda)\mathbf{f}^{-1}(\lambda)$ then

$$\int_{-\pi}^{\pi} \varphi(\lambda)\mathbf{f}(\lambda)\varphi^*(\lambda)d\lambda = \int_{-\pi}^{\pi} \mathbf{b}(\lambda)\mathbf{f}^{-1}(\lambda)\varphi^*(\lambda)d\lambda < \infty \tag{1.52}$$

and the random variable of the form $\int_{-\pi}^{\pi} \varphi(\lambda)d\mathbf{Z}(\lambda)$ is orthogonal to $\bar{H}(T)$:

$$\mathbb{E}[h\bar{X}_l(t)] = \int_{-\pi}^{\pi} e^{-i\lambda t}\sum_{k=1}^{s}[\varphi_k(\lambda)f_{kl}(\lambda)]d\lambda = \int_{-\pi}^{\pi} e^{-i\lambda t}b_l(\lambda)d\lambda = 0, \tag{1.53}$$

for all l and t, except for $t \in T_l$. But this means that h belongs to the subspace $\Delta(T)$, and, moreover, by virtue of (1.53),

$$\mathbb{E}\,|h|^2 = \|\mathbf{b}\|^2$$

We proceed now to a direct determination of the quantities $\hat{X}_k(t), t \in T_k$ which gives the best forecast by linear interpolation. Let $T_k = t_0$, $T_l = 0$ for $l \neq k$. As we already know, $\hat{X}_k(t_0)$ can be represented in the following form:

$$\hat{X}_k(t_0) = \int_{-\pi}^{\pi} \hat{\varphi}_k(\lambda)d\mathbf{Z}(\lambda) \tag{1.54}$$

The problem of linear interpolation consist, essentially, of determining the vector functions $\varphi_k(\lambda) = [\varphi_{k1}(\lambda), \varphi_{k2}(\lambda), \ldots, \varphi_{ks}(\lambda)]$.

Since the difference $X_k(t_0)$ - $\hat{X}_k(t_0)$ belongs to the space $\Delta(t)$, we obtained, from Lemma 1, that the vector function:

$$\mathbf{b}_k(\lambda) = [e^{i\lambda t_0}\delta_k - \hat{\varphi}_k(\lambda)]f(\lambda) = [b_{k1}(\lambda), b_{k2}(\lambda), \ldots, b_{ks}(\lambda)]$$

belongs to the space B(T), and, in particular, that:

$$b_{kj} = \sum_{t \in T_k} a_{kj}(t)e^{i\lambda t}, j = 1, 2, \ldots, s.$$

Thus, the vector function (row vector) $\hat{\varphi}_k(\lambda)$ has the form

$$\hat{\varphi}_k(\lambda) = e^{i\lambda t_0}\delta_k - \mathbf{b}_k(\lambda)\mathbf{f}^{-1}(\lambda), \tag{1.55}$$

where δ_k is a s-dimensional vector which has a 1 in the k-th position and zero in the other positions and the problem of linear interpolation reduces to finding the coefficients a_{kj} of the trigonometric polynomials $b_{kj}(\lambda)$. These coefficients can easily be found from a linear system of equations, expressing the fact that $\hat{X}_k(t_0)$ is orthogonal to $\Delta(T)$.

If the process $\mathbf{X}(t)$ is minimal then the vector functions of the form $e^{i\lambda t \delta_l}$, $t \in T_l$, $l = 1, \ldots, s$, form a basis in the space $B(T)$, and if one denotes by $h^{l,t}$ the corresponding variables in the space $\Delta(T)$, then the orthogonality of $\hat{X}_k(t_0)$ to $\Delta(T)$ is equivalent to the following:

$$\mathbb{E}[\hat{X}_k(t_0) h^{l,t}] = \int_{-\pi}^{\pi} e^{-i\lambda t} [\hat{\varphi}_k(\lambda) \mathbf{f}(\lambda) \bar{\mathbf{p}}_l(\lambda)] d\lambda = 0 \tag{1.56}$$

where $p_l(\lambda) = [p_{l1}(\lambda), p_{l2}(\lambda), \ldots, p_{ls}(\lambda)]$ is the lth row of the inverse $\mathbf{f}^{-1}(\lambda)$ of $\mathbf{f}(\lambda)$. Taking into consideration the form (1.55) of the vector function $\hat{\varphi}_k(\lambda)$, system (1.56) can be rewritten in the form:

$$\sum_{j=1}^{n} \sum_{s \in T_j} \gamma i_{jl}(s - t) a_{kj}(s) \;=\; 0 \qquad \text{for } t \in T_l \quad t \neq t_0, \;\; l \neq k \tag{1.57}$$

$$\sum_{j=1}^{n} \sum_{s \in T_j} \gamma i_{jk}(s - t_0) a_{kj}(s) \;=\; 1 \qquad \text{for } t \in T_k \quad t = t_0, \;\; l = k \tag{1.58}$$

Here the $\gamma i_{jl}(s)$ are Fourier coefficients of the elements $p_{jl}(\lambda)$ of the matrix $\mathbf{f}^{-1}$, that is, inverse covariance:

$$\gamma i_{jl}(s) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i\lambda s} p_{jl}(\lambda) d\lambda.$$

Theorem 2. . *Suppose that the spectral density f of the s-dimensional process $X(t)$ satisfies theorem 1. Then the random variables $\hat{X}_k(t_0)$, giving the best linear interpolation, can be found from formula (1.54), in which the vector functions $\hat{\varphi}_k(\lambda)$ are determined from the system of equations (1.57).*

Case 1: partial missing value for one component series

We suppose that $T_1 = \{t_0\}$ and $T_2 = \ldots = T_n = \{\varnothing\}$. In this case we have to determine $\hat{X}_1(t_0)$ and then only the vector function $\hat{\varphi}_1(\lambda)$. The vector function $b_1(\lambda)$ assume following form:

$$\mathbf{b}_1(\lambda) = [a_{11}(t_0)e^{i\lambda t_0}, 0, \ldots, 0]$$

.

The system (1.57) is defined only for $t \in T_1$:

$$\gamma i_{11}(0)a_{11}(t_0) = 1 \ \Rightarrow a_{11}(t_0) = [\gamma i_{11}(0)]^{-1}, \tag{1.59}$$

Substituting the value of $a_{11}(t_0)$ found by (1.59) in equation (1.55):

$$\hat{\varphi}_1(\lambda) = e^{i\lambda t_0}\delta_1 - \mathbf{b}_1(\lambda)\mathbf{f}^{-1} = e^{i\lambda t_0}\delta_1 - e^{i\lambda t_0}[\frac{1}{\gamma i_{11}(0)}f_{11}(\lambda), \ldots, \frac{1}{\gamma i_{11}(0)}f_{n1}(\lambda)] \tag{1.60}$$

Then:

$$\hat{X}_1(t_0) = \int_{-\pi}^{\pi} e^{i\lambda t_0}\delta_1 d\mathbf{Z}(\lambda) - \int_{-\pi}^{\pi} e^{i\lambda t_0}[\frac{1}{\gamma i_{11}(0)}f_{11}(\lambda), \ldots, \frac{1}{\gamma i_{11}(0)}f_{n1}(\lambda)]d\mathbf{Z}(\lambda) \tag{1.61}$$

whence:

$$\begin{aligned}\hat{X}_1(t_0) \quad = \quad & \int_{-\pi}^{\pi} e^{i\lambda t_0} dZ_1(\lambda) - \int_{-\pi}^{\pi} e^{i\lambda t_0}\frac{1}{\gamma i_{11}(0)}f_{11}(\lambda)dZ_1(\lambda) - \ldots - \\ & - \int_{-\pi}^{\pi} e^{i\lambda t_0}\frac{1}{\gamma i_{11}(0)}f_{n1}(\lambda)dZ_n(\lambda) \end{aligned} \tag{1.62}$$

Hence, writing:

$$f_{ij}^{-1}(\lambda) = \frac{1}{2\pi}\sum_{u=-\infty}^{\infty} e^{-i\lambda u}\gamma i_{ij}(u) \tag{1.63}$$

we have:

$$\int_{-\pi}^{\pi} e^{i\lambda t_0}\frac{1}{\gamma i_{11}(0)}f_{ij}(\lambda)dZ_i(\lambda) = \int_{-\pi}^{\pi} e^{i\lambda t_0}\frac{1}{\gamma i_{11}(0)}\sum_{u=-\infty}^{\infty} e^{-i\lambda u}\gamma i_{ij}(u)dZ_i \tag{1.64}$$

If we used equation of spectral representation, the ith integral of (1.62) becomes:

$$\sum_{u=-\infty}^{\infty}\int_{-\pi}^{\pi} e^{i\lambda(t_0-u)}\frac{1}{\gamma i_{11}(0)}dZ_i = \frac{\gamma i_{ij}(u)}{\gamma i_{11}(0)}\sum_{u=-\infty}^{\infty}\gamma i_{ij}(u)X(t_0-u) \tag{1.65}$$

Using this last relation we can write (1.62) as:

$$\begin{aligned}\hat{X}_1(t_0) &= X_1(t_0) - \frac{1}{\gamma i_{11}(0)} \sum_{u=-\infty}^{\infty} \gamma i_{11}(t) X_1(t_0-u) - \frac{1}{\gamma i_{11}(0)} \sum_{u=-\infty}^{\infty} \gamma i_{21}(u) X_2(t_0-u) - \ldots - \\ & - \frac{1}{\gamma i_{11}(0)} \sum_{u=-\infty}^{\infty} \gamma i_{s1}(t) X_n(t_0-u) \qquad (1.66)\end{aligned}$$

that can be written as:

$$\hat{X}_1(t_0) = X_1(t_0) - \frac{1}{\gamma i_{11}(0)} \sum_{u=-\infty}^{\infty} \sum_{j=1}^{s} \gamma i_{j1}(u) X_j(t_0-u)$$

Obviously the summation on the right of equation(1.66) is not defined for missing data $X_1(t_0)$. This quantity appears when $u = 0$ and in this case we have: $X_1(t_0) - a_{11}(t_0)\gamma i_{11}(0)X_1(t_0) = 0$ according to the system (1.59).

Case 2: partial missing value for two component series

We suppose that $T_1 = \{t_1\}$, $T_2 = \{t_2\}$ and $T_3 = \ldots = T_n = \{\varnothing\}$ and have to interpolate $X_1(t_1)$, that is, determine $\hat{X}_1(t_1)$.

When we have to interpolate the missing data of component k-th of stochastic process it is only necessary to determine the vector function $b_k(\lambda)$. In our case, as $k = 1$ we have to determine the function $b_1(\lambda)$. If there is only one missing data in a single component then this function has only one nonzero element at k-th column ($b_1(\lambda)$ is a row vector). If there are two missing data in the k-th component then the function $b_k(\lambda)$ has always only one element different from zero in correspondence of the k-th column but this element is the sum of two exponentials with coefficients different from zero. If instead there are two components that each have one missing then the function $b_k(\lambda)$ has two elements different from zero. In our case the function has two components different from zero in column 1 and 2. In fact we have:

$$b_{11}(\lambda) = a_{11}(t_1)e^{i\lambda t_1}, b_{12}(\lambda) = a_{12}(t_2)e^{i\lambda t_2}, b_{13} = b_{14} = \ldots = 0$$

and then:

$$\mathbf{b}_1(\lambda) = [a_{11}(t_1)e^{i\lambda t_1}, a_{12}(t_2)e^{i\lambda t_2}, 0, \ldots, 0]$$

We have to determine through the system (1.57) the two coefficients $a_{11}(t_1)$ and $a_{12}(t_2)$. The equations are the conditions that arise from the following reasoning: if we interpolate one missing data but there are two missing data we can not use the two data. So when we make the linear combination of the available data we have to ensure that these data do not appear. In this case the system (1.57) becomes:

$$\begin{cases} \gamma i_{11}(0)a_{11}(t_1)+\gamma i_{21}(t_2-t_1)a_{12}(t_2) &= 1 \\ \\ \gamma i_{12}(t_1-t_2)a_{11}(t_1)+\gamma i_{22}(0)a_{12}(t_2) &= 0 \end{cases}$$

then:

$$\begin{cases} a_{11}(t_1) &= [\gamma i_{21}(t_2-t_1)-\frac{\gamma i_{11}(0)\gamma i_{22}(0)}{\gamma i_{12}(t_2-t_1)}]^{-1} \\ \\ a_{12}(t_2) &= -\frac{\gamma i_{22}(0)}{\gamma i_{21}(t_2-t_1)\gamma i_{12}(t_1-t_2)-\gamma i_{11}(0)\gamma i_{22}(0)} \end{cases}$$

Substituting the values of the coefficients $a_{11}(t_1)$ e $a_{12}(t_2)$ in (1.55) we have:

$$\begin{aligned} \hat{\varphi}_1(\lambda) &= e^{i\lambda t_1}\delta_1 - b_1(\lambda)\mathbf{f}^{-1} \\ &= e^{i\lambda t_1}\delta_1 - [a_{11}(t_1)e^{i\lambda t_1}f_{11}(\lambda)+a_{12}(t_2)e^{i\lambda t_2}f_{21}(\lambda),\ldots,a_{11}(t_1)e^{i\lambda t_1}f_{1s}(\lambda)+ \\ &\quad + a_{12}(t_2)e^{i\lambda t_2}f_{2s}(\lambda)] \end{aligned} \tag{1.67}$$

$$\hat{\varphi}_1(\lambda) = \begin{bmatrix} e^{i\lambda t_1} \\ 0 \\ \cdots \\ 0 \end{bmatrix}'_{1\times s} - \begin{bmatrix} a_{11}(t_1)e^{i\lambda t_1}f_{11}(\lambda)+a_{12}(t_2)e^{i\lambda t_2}f_{21}(\lambda) \\ a_{11}(t_1)e^{i\lambda t_1}f_{12}(\lambda)+a_{12}(t_2)e^{i\lambda t_2}f_{22}(\lambda) \\ \\ \cdots \\ a_{11}(t_1)e^{i\lambda t_1}f_{1s}(\lambda)+a_{12}(t_2)e^{i\lambda t_2}f_{2s}(\lambda) \end{bmatrix}'_{1\times s}$$

$$\begin{aligned} \hat{X}_1(t_1) &= \int_{-\pi}^{\pi} e^{i\lambda t_0}\delta_1 dZ(\lambda) - \int_{-\pi}^{\pi} a_{11}(t_1)e^{i\lambda t_1}f_{11}(\lambda)dZ_1(\lambda) - \\ &\quad - \int_{-\pi}^{\pi} a_{12}(t_2)e^{i\lambda t_2}f_{21}(\lambda)dZ_1(\lambda) - \ldots - \int_{-\pi}^{\pi} a_{11}(t_1)e^{i\lambda t_1}f_{1s}(\lambda)dZ_s(\lambda) - \\ &\quad - \int_{-\pi}^{\pi} a_{12}(t_2)e^{i\lambda t_2}f_{2s}(\lambda)dZ_s(\lambda) \end{aligned} \tag{1.68}$$

$$\begin{aligned} \hat{X}_1(t_1) &= X_1(t_1) - a_{11}(t_1)\sum_u \gamma i_{11}(u)X_1(t_1-u) - a_{12}(t_2)\sum_t \gamma i_{21}(u)X_1(t_2-u) - \ldots - \\ &\quad - a_{11}(t_1)\sum_t \gamma i_{1s}(u)X_s(t_1-u) - a_{12}(t_2)\sum_t \gamma i_{2s}(t)X_s(t_2-u) \end{aligned} \tag{1.69}$$

Case 3: one missing value for all component series

In this case we suppose that $T_1 = T_2 = \ldots = T_n = T = \{t_0\}$. Then, the values $\hat{X}_k(t_0)$ of the process $X(t)$ are all unknown for the same time t_0. Let $\hat{\mathbf{X}}(t_0) = [\hat{X}_1(t_0), \ldots, \hat{X}_s(t_0)]$. We have:

$$\hat{\mathbf{X}}(t_0) = \int_{-\pi}^{\pi} \hat{\varphi}(\lambda) dZ(\lambda) \tag{1.70}$$

where, by virtue of (1.55), the matrix function $(s \times s)$ $\hat{\varphi}(\lambda)$ has the form:

$$\hat{\varphi}(\lambda) = e^{i\lambda t_0}\mathbf{I}_s - \sum_{s \in T} e^{i\lambda s}\mathbf{a}(s)\mathbf{f}^{-1}(\lambda) \tag{1.71}$$

For the matrix coefficients $(s \times s)$ $a(s)$ we obtained from (1.57) the following system of equations:

$$\begin{aligned} \sum_{s \in T} \mathbf{\Gamma i}(s - t_0)\mathbf{a}(s) &= \mathbf{I}_s, \\ \sum_{s \in T} \mathbf{\Gamma i}(s - t_0)\mathbf{a}(s) &= \mathbf{0}_s \text{ for } t \neq t_0 \end{aligned} \tag{1.72}$$

where:

$$\mathbf{\Gamma i}(s) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i\lambda s}\mathbf{f}^{-1} d\lambda. \tag{1.73}$$

The system of equations (1.72) will then appear as:

$$\mathbf{\Gamma i}(0)\mathbf{a}(t_0) = \mathbf{I}_s \tag{1.74}$$

We find that:

$$\mathbf{a}(t_0) = [\mathbf{\Gamma i}(0)]^{-1} \tag{1.75}$$

The expression of interpolator is then:

$$\begin{aligned} \hat{\mathbf{X}}(t_0) &= \int_{-\pi}^{\pi} e^{i\lambda t}\mathbf{I}_s dZ(\lambda) - \int_{-\pi}^{\pi} [\mathbf{\Gamma i}(0)]^{-1}\mathbf{f}^{-1} dZ(\lambda) \\ &= \mathbf{X}(t_0) - [\mathbf{\Gamma i}(0)]^{-1} \sum_u \mathbf{\Gamma i}(u)\mathbf{X}(t_0 - u) \end{aligned} \tag{1.76}$$

If we denote with $\epsilon_k = X_k(t_0) - \hat{X_k(t_0)}$ the errors interpolation and with $\sigma_{kj} = \mathbb{E}[\epsilon_k \bar{\epsilon_j}]$, then the matrix of error $\sigma^2 = \{\sigma_{kj}\}$ of linear interpolation is easily found from the representation:

$$\Sigma = 2\pi[\Gamma_i(0)]^{-1} \tag{1.77}$$

Case 4: Two missing values for all component series

We suppose that $T_1 = T_2 = \ldots = T_n = T = \{t_0, t_1\}$. We have:

$$\hat{\mathbf{X}}(t_0) = \int_{-\pi}^{\pi} \hat{\varphi}(\lambda) d\mathbf{Z}(\lambda) \tag{1.78}$$

where, by virtue of (1.55), the matrix function $(s \times s)$ $\hat{\varphi}(\lambda)$ has the form:

$$\hat{\varphi}(\lambda) = e^{i\lambda t_0}\mathbf{I}_s - e^{i\lambda t_0}\mathbf{a}(t_0)\mathbf{f}^{-1}(\lambda) - e^{i\lambda t_1}\mathbf{a}(t_1)\mathbf{f}^{-1}(\lambda) \tag{1.79}$$

For the matrix coefficients $(s \times s)$ $a(s)$ we obtained from (1.57) the following system of equations:

$$\begin{aligned} \mathbf{\Gamma i}(0)\mathbf{a}(t_0) + \mathbf{\Gamma i}(t_1 - t_0)\mathbf{a}(t_1) &= \mathbf{I}_s, & (1.80) \\ \mathbf{\Gamma i}(t_0 - t_1)\mathbf{a}(t_0) + \mathbf{\Gamma i}(0)\mathbf{a}(t_1) &= \mathbf{0}_s \ \text{ for } t \neq t_0 & (1.81) \end{aligned}$$

$$\begin{bmatrix} \mathbf{\Gamma i}(0) & \mathbf{\Gamma i}(t_1 - t_0) \\ \mathbf{\Gamma i}(t_0 - t_1) & \mathbf{\Gamma i}(0) \end{bmatrix} \begin{bmatrix} \mathbf{a}(t_0) \\ \mathbf{a}(t_1) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_s \\ \mathbf{0}_s \end{bmatrix}$$

$$\mathbf{a}(t_0) = [\mathbf{\Gamma i}(0) - \mathbf{\Gamma i}(t_1 - t_0)\mathbf{\Gamma i}^{-1}(0)\mathbf{\Gamma i}(t_0 - t_1)]^{-1} \tag{1.82}$$

$$\mathbf{a}(t_1) = -[\mathbf{\Gamma i}(0) - \mathbf{\Gamma i}(t_1 - t_0)\mathbf{\Gamma i}^{-1}(0)\mathbf{\Gamma i}(t_0 - t_1)]^{-1}\mathbf{\Gamma i}(t_1 - t_0)\mathbf{\Gamma i}^{-1}(0) \tag{1.83}$$

The expression of interpolator is:

$$\begin{aligned} \hat{\mathbf{X}}(t_0) &= \int_{-\pi}^{\pi} e^{i\lambda t_0} I_s - \int_{-\pi}^{\pi} \mathbf{a}(t_0)\mathbf{f}^{-1} dZ(\lambda) - \int_{-\pi}^{\pi} \mathbf{a}(t_1)\mathbf{f}^{-1} dZ(\lambda) & (1.84) \\ &= X(t_0) - \mathbf{a}(t_0)\sum_u \mathbf{\Gamma i}(u) X(t_0 - u) - \mathbf{a}(t_1)\sum_u \mathbf{\Gamma i}(u) X(t_1 - u) \end{aligned}$$

The two summations are undefined for $\mathbf{X}(t_0)$ and $\mathbf{X}(t_1)$. In the first sum, $\mathbf{X}(t_0)$ appears when $u = 0$ while in the second sum when $u = (t_1 - t_0)$:

$$\mathbf{X}(t_0) - \mathbf{a}(t_0)\mathbf{\Gamma i}(0)\mathbf{X}(t_0) - \mathbf{a}(t_1)\mathbf{\Gamma i}(t_1 - t_0)\mathbf{X}(t_0) = \mathbf{I}_s$$

because $\mathbf{a}(t_0)\mathbf{\Gamma i}(0) - \mathbf{a}(t_1)\mathbf{\Gamma i}(t_1 - t_0) = \mathbf{I}_s$. $\mathbf{X}(t_1)$ appears when $u = t_0 - t_1$ in the first sum and when $u = 0$ in the second sum:

$$\mathbf{a}(t_0)\mathbf{\Gamma i}(t_0 - t_1)\mathbf{X}(t_1) - \mathbf{a}(t_1)\mathbf{\Gamma i}(0)\mathbf{X}(t_1) = \mathbf{0}_s$$

because $\mathbf{a}(t_0)\mathbf{\Gamma i}(t_0 - t_1) - \mathbf{a}(t_1)\mathbf{\Gamma i}(0) = \mathbf{0}_s$

1.4.2 Frequency domain approach to interpolation

Hannan (1970) deals with the linear interpolator problem considering the case where $T_1 = T_2 = \ldots = T_s = T = \{t_0\}$. The author determines the optimal linear interpolator trying the linear combination $\hat{\mathbf{X}}(t_0)$ of $\mathbf{X}(t_0 - j), j \neq t_0$, which minimizes the error of interpolation $\left\|\mathbf{X}(t_0) - \hat{\mathbf{X}}(t_0)\right\|^2$. The demonstration that leads to the optimal linear interpolator is reported below.

We introduce the response functions:

$$h_N(e^{i\lambda}) = \sum_{j=-N}^{N} A_N(j)e^{ij\lambda}, \tag{1.85}$$

where the term for $j = t_0$ is omitted. Now we seek for a response function h such that:

$$\lim_{N\to\infty}[\int_{-\pi}^{\pi}(h - h_N)d\mathbf{H}(\lambda)(h - h_N)^*] = 0 \tag{1.86}$$

and

$$[\int_{-\pi}^{\pi}(I_s - h)d\mathbf{H}(\lambda)(I_s - h)^*]$$

is minimized. If we determined the transfer function h, the optimal interpolator results to be given by:

$$\hat{\mathbf{X}}(t) = \int_{-\pi}^{\pi} e^{-it\lambda}h(e^{-i\lambda})d\mathbf{Z}(\lambda)$$

while the covariance matrix of interpolation errors is given by:

$$\mathbf{\Sigma} = \mathbb{E}\{[\mathbf{X}(t) - \hat{\mathbf{X}}(t)][\mathbf{X}(t) - \hat{\mathbf{X}}(t)]'\}.$$

Evidently may take $t_0 = 0$ without any loss of generality. If $\mathbf{H}(\lambda)$ is not a.c., we know that the singular part of $\mathbf{F}(\lambda)$ corresponds to a perfectly predictable process and thus one which may be perfectly interpolated. This leads us to treat the a.c. case. We assume that there is no non-null vector α such that $\alpha'\mathbf{X}(t) \equiv 0$, almost surely.

Theorem 3. . *Let $\mathbf{X}(t)$ satisfy the above assumption and have a.c. spectrum and let $\mathbf{f}^{-1}(\lambda)$ be the inverse of $\mathbf{f}(\lambda)$. The necessary and sufficient condition that $\mathbf{\Sigma}$ be nonsingular is the condition that $\mathbf{f}^{-1}(\lambda)$ be integrable. Then the response function of the optimal interpolating filter is:*

$$h = I_s - \{\frac{1}{2\pi}\int_{-\pi}^{\pi} \mathbf{f}^{-1}(\lambda)d\lambda\}\mathbf{f}^{-1}(\lambda) \tag{1.87}$$

and covariance matrix of interpolation errors is:

$$\mathbf{\Sigma} = \{\frac{1}{2\pi}\int_{-\pi}^{\pi} [2\pi\mathbf{f}(\lambda)]^{-1}\}^{-1} \tag{1.88}$$

Proof. Evidently, since $[\mathbf{X}(0) - \hat{\mathbf{X}}(0)]$ is orthogonal to $\mathbf{X}(t)$ $\forall t \neq 0$, for each pair of vectors α, β of complex numbers we must have:

$$\mathbb{E}\{\alpha^*[\mathbf{X}(0) - \hat{\mathbf{X}}(0)]\mathbf{X}(t)'\beta\} = 0,\ t \neq 0 \tag{1.89}$$

and using the definition of scalar product we have:

$$\alpha^* \int (I_s - h)\mathbf{f}(\lambda)e^{it\lambda}d\lambda\beta = 0,\ t \neq 0$$

Since the Fourier coefficients are zero in the case of a constant function, this implies that:

$$(I_s - h)\mathbf{f} = \mathbf{C}$$

where C is a constant matrix. Thus

$$(I_s - h) = \mathbf{C}\mathbf{f}^{-1}$$

This solution is not unique, but any solution differs from it by a matrix which, when multiplied on the right by f, is annihilated and thus leads to the same $\mathbf{\Sigma}$. Moreover, (1.89) also shows that:

$$\int_{-\pi}^{\pi}(I_s - h)fh^* d\lambda = 0,$$

since h is a limit in mean square of expression of the form of (1.85). Thus:

$$\mathbf{\Sigma} = \int (I_s - h)\mathbf{f} d\lambda = 2\pi \mathbf{C},$$

which shows that $\mathbf{C} = \mathbf{C}^* = \bar{\mathbf{C}}$. Now, assuming the integrability of $\mathbf{f}^{-1}$ for the first time, we have

$$\mathbf{\Sigma} = \mathbf{C}\int_{-\pi}^{\pi}\mathbf{f}^{-1}\mathbf{f}\mathbf{f}^{-1}d\lambda = \mathbf{C}\int_{-\pi}^{\pi}\mathbf{f}^{-1}d\lambda\mathbf{C},$$

and

$$\mathbf{C} = \mathbf{C}\{\frac{1}{2\pi}\int_{-\pi}^{\pi}\mathbf{f}^{-1}(\lambda)d\lambda\}\mathbf{C}$$

of which a solution is

$$\mathbf{C} = \{\frac{1}{2\pi}\int_{-\pi}^{\pi}\mathbf{f}^{-1}(\lambda)d\lambda\}^{-1}. \tag{1.90}$$

From last equation we have:

$$(I_s - h) = \{\frac{1}{2\pi}\int_{-\pi}^{\pi}\mathbf{f}^{-1}(\lambda)d\lambda\}^{-1}\mathbf{f}^{-1} = \mathbf{\Gamma_i}(0)\mathbf{f}^{-1} \tag{1.91}$$

and

$$\mathbf{\Sigma} = 2\pi[\mathbf{\Gamma_i}(0)]^{-1}, \tag{1.92}$$

that coincides with the equation(1.77).

Thus we can take C given by (1.90) and the theorem results, save for the assertion concerning the nonsingularity of $\mathbf{\Sigma}$. If $\mathbf{f}^{-1}(\lambda)$ is integrable then certainly $\mathbf{\Sigma}$ is nonsingular for otherwise there must be a vector α, so that

$$\int \alpha' \mathbf{f}^{-1}(\lambda)\alpha d\lambda = 0, \quad \alpha'\alpha = 1 \tag{1.93}$$

Taking α as first row of an orthogonal matrix $\mathbf{P}$ this implies that $\mathbf{Pf}(\lambda)\mathbf{P}'$ must have null elements, for all λ, in the first row and column, which implies that $\alpha'\mathbf{X}(t) \equiv 0$, almost surely. On the other hand, if $\mathbf{\Sigma}$ is nonsingular then since $\int(I_s - h)\mathbf{f}(I_s -$

$h)'d\lambda = \int (4\pi)^{-2}\mathbf{\Sigma f}^{-1}\mathbf{\Sigma} d\lambda$ we see that $\mathbf{\Sigma f}^{-1}\Sigma$ is integrable and thus so must $\mathbf{f}^{-1}$ be integrable. This completes the proof.

Substituting the equation (1.92) in equation (1.87) we obtained the formula (1.76) of optimal interpolator found by Rozanov (1957).

1.4.3 Time domain approach to interpolation

Battaglia (1984) consider the linear interpolation problem for a multivariate stationary process $\mathbf{X}(t)$ and suppose that $T_1 = T_2 = \ldots = T_s = T = \{t = 0\}$. The problem is to determine a linear transformation of $\{\ldots, \mathbf{X}(t-2), \mathbf{X}(t-1), \mathbf{X}(t+1), \mathbf{X}(t+2), \ldots\}$:

$$\sum_{u \neq 0} \mathbf{a}(u)\mathbf{X}(t-u)$$

with $\{a(u)\}$ real matrices $s \times s$, such that the linear combination $\sum_{u\neq 0} \mathbf{a}(u)\mathbf{X}(t-u)$ is as close possible to $\mathbf{X}(t)$. To this aim, Battaglia (1984) consider the variance matrix:

$$\mathbb{E}\{[\mathbf{X}(t) - \sum_{u\neq 0} \mathbf{a}(u)\mathbf{X}(t-u)][\mathbf{X}(t) - \sum_{u\neq 0} \mathbf{a}(u)\mathbf{X}(t-u)]'\}. \tag{1.94}$$

and minimize it according to the positive-definiteness ordering. This ordering is defined for Hermitian matrices by $\mathbf{A} \geq \mathbf{B}$ if $\mathbf{A} - \mathbf{B} \geq 0$ where $\mathbf{M} \geq 0$ means that the matrix $\mathbf{M}$ is positive semidefinite, and matches with the orderings induced by the values of determinants and traces. To find the matrix that minimize the mean square error, the author has expressed it as a sum of a matrix independent of the c_u and a positive semidefinite matrix. The demonstration was done in the frequency domain, replacing the variance-covariance matrix with the integral of its spectral density.

Let $\mathbf{I}(t) = \sum_{u\neq 0} \mathbf{a}(u)\mathbf{X}(t-u)$, and denote by $\mathbf{A}(\lambda) = I_s - \sum_{u\neq 0} \mathbf{a}(u)e^{-i\lambda u}$ the transfer function of $\mathbf{I}(t)$. The residual (or interpolation error) $\mathbf{X}(t) - \mathbf{I}(t)$ has variance-covariance matrix given by:

$$\mathbb{E}\{[\mathbf{X}(t) - \mathbf{I}(t)][\mathbf{X}(t) - \mathbf{I}(t)]'\} = \int_{-\pi}^{\pi} \mathbf{A}(\lambda)\mathbf{f}(\lambda)\mathbf{A}(\lambda)^* d\lambda, \tag{1.95}$$

Now,

$$\begin{aligned}\mathbf{A}(\lambda)\mathbf{f}(\lambda)\mathbf{A}(\lambda)^* \;&=\; \boldsymbol{\Gamma}_{\mathbf{i}}{}^{-1}(0)\mathbf{A}(\lambda)^* + \mathbf{A}(\lambda)^*\boldsymbol{\Gamma}_{\mathbf{i}}{}^{-1}(0) - \boldsymbol{\Gamma}_{\mathbf{i}}{}^{-1}(0)\mathbf{f}^{-1}(\lambda)\boldsymbol{\Gamma}_{\mathbf{i}}{}^{-}1(0) + \\ &\quad + \{\mathbf{A}(\lambda) - \boldsymbol{\Gamma}_{\mathbf{i}}{}^{-1}(0)\mathbf{f}^{-1}\}\mathbf{f}(\lambda)\{\mathbf{A}(\lambda) - \boldsymbol{\Gamma}_{\mathbf{i}}{}^{-1}(0)\mathbf{f}^{-1}\}^* \qquad (1.96)\end{aligned}$$

Since the last matrix in the second line of (1.96) is positive semidefinite ($\mathbf{f}(\lambda)$ is positive semidefinite), it follows that:

$$\mathbf{A}(\lambda)\mathbf{f}(\lambda)\mathbf{A}(\lambda)^* \geq \boldsymbol{\Gamma}_{\mathbf{i}}{}^{-1}(0)\mathbf{A}(\lambda)^* + \mathbf{A}(\lambda)^*\boldsymbol{\Gamma}_{\mathbf{i}}{}^{-1}(0) - \boldsymbol{\Gamma}_{\mathbf{i}}{}^{-1}(0)\mathbf{f}^{-1}(\lambda)\boldsymbol{\Gamma}_{\mathbf{i}}{}^{-1}(0) \qquad (1.97)$$

integrating, and considering that:

$$\int_{-\pi}^{\pi} \mathbf{f}^{-1}(\lambda)d\lambda = \boldsymbol{\Gamma}_{\mathbf{i}}(0); \quad \int_{-\pi}^{\pi} \mathbf{A}(\lambda)d\lambda = \int_{-\pi}^{\pi} \mathbf{A}(\lambda)^*d\lambda = \mathbf{I}_s \qquad (1.98)$$

he obtained:

$$\mathbb{E}\{[\mathbf{X}(t) - \mathbf{I}(t)][\mathbf{X}(t) - \mathbf{I}(t)]'\} = \boldsymbol{\Gamma}_{\mathbf{i}}(0) \qquad (1.99)$$

The minimum is attained when:

$$\{\mathbf{A}(\lambda) - \boldsymbol{\Gamma}_{\mathbf{i}}{}^{-}1(0)\mathbf{f}^{-1}\}\mathbf{f}(\lambda)\{\mathbf{A}(\lambda) - \boldsymbol{\Gamma}_{\mathbf{i}}{}^{-1}(0)\mathbf{f}^{-1}\}^* \qquad (1.100)$$

equals to zero matrix for each λ, i.e. when:

$$\mathbf{A}(\lambda) = \boldsymbol{\Gamma}_{\mathbf{i}}{}^{-1}(0)\mathbf{f}^{-1}(\lambda), \qquad (1.101)$$

so that $\mathbf{a}(u) = -\boldsymbol{\Gamma}_{\mathbf{i}}{}^{-1}(0)\boldsymbol{\Gamma}_{\mathbf{i}}(u)$. We can see that the equation (1.101) coincides with equation (1.91) found by Hannan (1970).

Chapter 2

Multivariate Self-Exciting Threshold Autoregressive Modeling by Genetic Algorithms

2.1 Introduction

Several papers that generalize the univariate threshold principle to a multivariate framework have appeared in the literature during the past years. Tiao & Tsay (1994) proposed a univariate SETAR model for the United States gross national product (GNP) series where the thresholds are controlled by two lagged values of the transformed GNP series reflecting the situation of the economy. Tsay (1998) developed a strategy for testing and estimating multivariate threshold models where the threshold variable was controlled by known linear combination of individual variables. Arnold & Gunther (2001) proposed a definition of MSETAR models where each linear regime follows a VAR process and the threshold variable is multivariate. Furthermore, they developed a estimation procedure of the corresponding autoregressive (AR) coefficient matrices. However, the authors suppose that the model structural parameters (delay, threshold variable, number and position of thresholds, model order) have to be known a priori. In the present framework, we adopt a less restrictive formulation, assuming that the structural parameters are unknown and are jointly estimated with the other parameters of the model. We formulated the task of finding the threshold variable and the others structural parameters as a combinatorial optimization problem (Medeiros et al. 2002).

Combinatorial optimization is a field of applied mathematics that treats a special type of mathematical optimization problem where the set of feasible solutions is finite. The gradient based methods cannot be used in such a space as the search space is discrete and derivatives and usual notions of continuity and convexity do not apply. If the size of the problem is small often exhaustive enumeration of all potential solutions is feasible and it is the best way to obtain an exact solution. However, often such method is unfeasible because in combinatorial problems the solution space grows very large as a function of the problem size. For moderate size dynamic programming offers several algorithms that can provide good solutions or even exact solutions. Nonetheless, more complex problems may be tackled only with the use of heuristic methods. Moreover, as the computing time needed to get a solution becomes usually exponentially large even heuristics may be unfit for optimization and we have to resort to meta heuristic algorithms that may provide in polynomial time a good sub-optimal solution or even the exact solution in some special cases. These problems are included in the class of the NP-complete combinatorial optimization problems as no polynomial time algorithm is known that may produce the optimum solution.

A widespread class of meta heuristics that have been found effective in statistical application involving NP-complete optimization task are the GAs. GAs have been employed to solve optimization problems that arise in the design of many complex systems, e.g. communication systems, networks, operations research, medicine and biochemistry. Formulation of basic principles is due to Holland (1975) while introduction and discussion of detailed theory and applications of GAs as optimization algorithms may be found in many textbooks. See, e.g., Goldberg (1989) and Mitchell (1996), two nice introductory books, Back et al. (1997), where related fields too such as evolution strategies and genetic programming are illustrated, Gen & Cheng (1997) and Haupt & Haupt (2004), who cope with applications and present examples from several different fields. In the present framework we have to deal with a very large space of potential optimal solutions as threshold variable (components and delay), the thresholds and the AR orders have to be found that optimize some suitable objective function. Applications of GAs to threshold modeling in the univariate case have been suggested by Wu & Chang (2002) and Baragona et al. (2004), and extensions have been studied to non stationary case by Battaglia & Protopapas (2011, 2012), to double threshold generalized autoregressive conditional heteroscedastic (GARCH) models by Baragona & Cucina (2008).

The rest of the paper is organized as follows. Section 2.2 gives a general description of MSETAR model. Section 2.3 presents the GAs methodology used for identification and estimation of MSETAR models. Section 2.4 presents some numerical examples illustrating the performance of the proposed procedure for MSETAR model building. Several models are considered and results from a Monte Carlo experiment are displayed and commented. Section 2.5 shows an application concerned with a real data set.

2.2 The MSETAR model formulation

Consider a K-dimensional time series $Y_t = (y_{1t}, y_{2t}, ..., y_{Kt})'$. Let $l_1, l_2, \ldots, l_K$ be positive integers and for each $1 \leq i \leq K$ $(R^i_{j_i})_{j_i=1,2,\ldots,l_i}$ a disjunctive decomposition of the real axis:

$$\mathbb{R} = \bigcup_{j_i=1}^{l_i} R^i_{j_i} \qquad i = 1, 2, \ldots K$$

$$R^i_{j_i} = (r^{(i)}_{j_{i-1}}, r^{(i)}_{j_i}] \qquad -\infty = r^{(i)}_0 < r^{(i)}_1 < \ldots < r^{(i)}_{l_i} = \infty$$

Let $J = (j_1, j_2, ..., j_K)$. A K-dimensional MSETAR model is defined as

$$Y_t = \sum_J \left[\Phi^{(J)}_0 + \sum_{i=1}^{p_J} \Phi^{(J)}_i Y_{t-i} + U^{(J)}_t \right] I^{(J)}(Y_{t-d}) \tag{2.1}$$

where d is the delay parameter and the indicator function $I^{(J)} : Y_{t-d} \to \{0, 1\}$ which determines the current regime is defined by the relation

$$I^{(J)}(Y_{t-d}) = 1 \Leftrightarrow y_{i(t-d)} \in R^i_{j_i} \qquad i = 1, 2, \ldots, K.$$

A drawback with Model (2.1) may occur when the value of l_i is greater than 2 or the number of components K is greater than 2, because the number of regimes increases quickly. Indeed a model with a large number of regimes is difficult to interpret. For this reason we consider only MSETAR with bivariate threshold variable $Y_{t-d} = (y_{i_1,t-d_1}, y_{i_2,t-d_2})'$, $\quad i_1, i_2 = 1, 2, \ldots, K$, $l_1 = l_2 = 2$, and d_1, d_2 are assumed to vary in the set of the integers $\{1, \ldots, d_{\max}\}$. The integer $d_{\max}$ is chosen as a convenient upper bound for the allowed lags. A bivariate SETAR model may be written

$$Y_t = \sum_{j_1=1}^{2} \sum_{j_2=1}^{2} I^{(j_1,j_2)}(Y_{t-d}) \left[\Phi^{(j_1,j_2)}_0 + \sum_{i=1}^{p_{j_1,j_2}} \Phi^{(j_1,j_2)}_i Y_{t-i} + U^{(j_1,j_2)}_t \right], \tag{2.2}$$

where the threshold variable is a bivariate vector where the entries are two lagged series chosen among the components of the multivariate time series $(y_{1,t-d_1}, y_{2,t-d_2}, \ldots, y_{K,t-d_K})'$. Now let us consider these partitions of the real line

$$\mathbb{R} = \bigcup_{j=1}^{l_1} R_j^1 = \bigcup_{j=1}^{2} R_j^1 = R_1^1 \cup R_2^1 = (-\infty, r_1^{(1)}] \cup (r_1^{(1)}, \infty)$$

$$\mathbb{R} = \bigcup_{j=1}^{l_2} R_j^2 = \bigcup_{j=1}^{2} R_j^2 = R_1^2 \cup R_2^2 = (-\infty, r_1^{(2)}] \cup (r_1^{(2)}, \infty),$$

then the indicator functions of Model (2.2) assume the form

$$I^{(1,1)}(Y_{t-d}) = \begin{cases} 1 \;\Leftrightarrow \begin{cases} y_{i_1,t-d_1} \in R_1^1 \\ y_{i_2,t-d_2} \in R_1^2 \end{cases} \Leftrightarrow \begin{cases} y_{i_1,t-d_1} \leq r_1^{(1)} \\ y_{i_2,t-d_2} \leq r_1^{(2)} \end{cases} \\ 0 \;\; \text{otherwise} \end{cases}$$

$$I^{(1,2)}(Y_{t-d}) = \begin{cases} 1 \;\Leftrightarrow \begin{cases} y_{i_1,t-d_1} \in R_1^1 \\ y_{i_2,t-d_2} \in R_2^2 \end{cases} \Leftrightarrow \begin{cases} y_{i_1,t-d_1} \leq r_1^{(1)} \\ y_{i_2,t-d_2} > r_1^{(2)} \end{cases} \\ 0 \;\; \text{otherwise} \end{cases}$$

$$I^{(2,1)}(Y_{t-d}) = \begin{cases} 1 \;\Leftrightarrow \begin{cases} y_{i_1,t-d_1} \in R_2^1 \\ y_{i_2,t-d_2} \in R_1^2 \end{cases} \Leftrightarrow \begin{cases} y_{i_1,t-d_1} > r_1^{(1)} \\ y_{i_2,t-d_2} \leq r_1^{(2)} \end{cases} \\ 0 \;\; \text{otherwise} \end{cases}$$

$$I^{(2,2)}(Y_{t-d}) = \begin{cases} 1 \;\Leftrightarrow \begin{cases} y_{i_1,t-d_1} \in R_2^1 \\ y_{i_2,t-d_2} \in R_2^2 \end{cases} \Leftrightarrow \begin{cases} y_{i_1,t-d_1} > r_1^{(1)} \\ y_{i_2,t-d_2} > r_1^{(2)} \end{cases} \\ 0 \;\; \text{otherwise.} \end{cases}$$

These functions determine the current regime that is defined by a sub-region of the real plane $\mathbb{R} \times \mathbb{R}$ with x-axis equal to $y_{i_1,t-d_1}$ and y-axis equal to $y_{i_2,t-d_2}$. In Fig. 2.2 an example is given where the threshold components are $y_{1,t-d}$ and $y_{2,t-d}$,

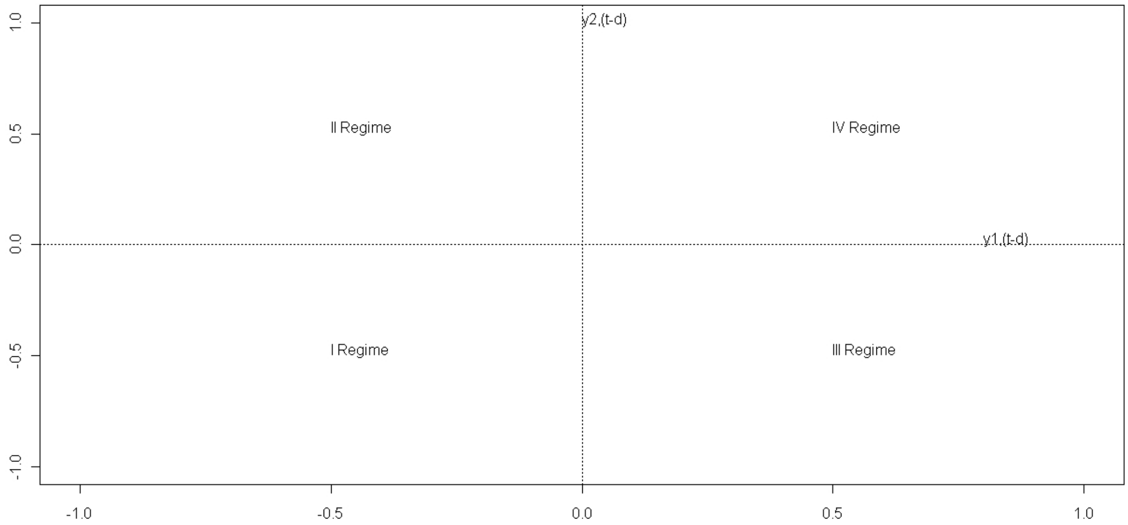

Figure 2.1: Threshold variables space for bivariate MSETAR model

$y_{it} \in (-1, 1)$ and the thresholds $r_1^{(1)}$ and $r_1^{(2)}$ are assumed to be zero and divide $(-1, 1) \times (-1, 1) \subset \mathbb{R} \times \mathbb{R}$ into four sub-regions, one for each regime.

The most important step in the identification and estimation of Model (2.2) consists in finding the correct elements of threshold variable Y_{t-d} and the position of thresholds. Once the threshold variables and the corresponding thresholds are specified, the orders P_J are determined with the use of the Akaike (1974) automatic identification criterion (AIC). Though several other such criteria have been suggested and comparisons have been made (see, e.g., Sayyareha et al. 2011) no definite results have been offered whether some may be considered the best one in all circumstances. So we adopt the well known and widely used AIC criterion adjusted to support model order choice, i.e. the minimum AIC estimate (Tong 1990). Given a candidate set of lags, $p_1, ..., p_{\max}$, we have to estimate several linear models and select the order that minimizes the information criteria. Once structural parameters of model (threshold variable, number and position of thresholds, model order) have been determined, the remaining coefficients of the model can be estimated by ordinary least squares.

The structural parameters take discrete values and their combinations amount to a very large number. In this work we formulated the task of finding the elements of threshold variable and the position of thresholds as a combinatorial optimization problem and we develop GAs to solve the problem.

2.3 The genetic algorithm for MSETAR modeling

GAs are simplified schemes of the evolutionary processes that develop in nature and have been used as all purposes optimization tools once the association between adaptation to the environment and objective function, and individual competing for survival and possible alternative solutions has been established. Results from application in several distant fields justified the development of GAs as numerical optimizers with the introduction of problem oriented variants of their basic features.

The general scheme of the GAs optimizers includes an initial population of potential solutions and an iterative loop where the current population is evaluated in terms of the fitness function of its individuals. The three usual genetic operators are selection, crossover and mutation. Though others have been suggested, e.g. inversion and splicing (see Michalewicz 1996) these only operators have been widely used in practical applications and many variants have been suggested to improve their potential in improving the average and the best fitness function and contemporaneously maintaining diversity among individuals. The three operators produce a new generation by choosing the most fit individuals, recombining their genetic material and allowing mutation to occur. This new generation replaces either partially or in full the old population according to some definite rules. The new population may either be constrained to have the same size than the past one or it may even be allowed to increase its size. An important feature in this 'reproduction' process is the 'elitist strategy', i.e. if the best individual found in the past generation is not selected for reproduction, it is included anyway in the new generation provided that no better individual has been produced. This ensures that the best fitness function never decreases through iterations. In addition, if an optimum exists, then the elitist GA converges asymptotically to this optimum (Rudolph 1997, Reeves & Rowe 2003).

Now we may explain the three operators as they have been used in our optimization problem and the encoding that has been adopted. Each solution (the 'individual') is represented as a string of digits (the 'chromosome'). Each digit may be thought of as a 'gene' which may take values ('alleles') in a given set according to its position (the 'locus') and meaning. The definition of the sets of allelic values allows possible constraints to be taken properly into account. Some features have been assumed that have become standards in GAs applications. For instance, the elitist strategy has been applied in such a way the best individual in the past generation that has to be included in the new population replaces the worst individual in the new generation. Finally, no stopping rule has been specified and the algorithm is allowed to run all iterations whose number has been fixed in advance. Indeed the asymptotic convergence results do not give information about the rate of convergence in real world data applications and the suggested number of iterations (e.g. Aytug & Koehler 2000) often results in an impractical large number. So usually the number of iterations is assumed rather large compared to the available computing resources and the requested timeliness of estimation results.

2.3.1 Encoding

The encoding uses a chromosome of length 15 for each individual in the current population. The 'locus' of each gene in the chromosome is important not only because it defines the meaning of the gene but also because only some genes have binary digits as allelic values while most of them have integer numbers as alleles with possibly different minimum and maximum values. Notice that each integer number is represented as a binary string (field) and the genetic operators apply on each field, for instance the crossover operator only operates at the boundaries between the binary fields. The chromosome we adopted in our GA is composed of the following genes:

- (1) A binary digit that acts as a switch, its value is 0 if the threshold variable is univariate, i.e. it refers to a single component series, 1 if the threshold variable is multivariate. The decoding of the rest of the chromosome depends on this first gene.

Genes $2-7$ *alleles under consideration provided that the first gene is* 0.

- (2) This gene encodes which component series has to be assumed as the threshold variable. It may assume the allelic values $1, 2, \ldots, K$.
- (3) Number of regimes (either 2, 3 or 4).
- (4-6) Positions of the thresholds. Assuming $t = 1$ the timing of the first observation, each of such positions is the time t associated to an observation in the chosen sequence (gene 2). So each position may range from 1 to n. How many genes have to be considered depends on the number of regimes as specified by the preceding gene 3.
- (7) Delay d for the scalar threshold variable, $d \in \{1, 2, \ldots, d_{\max}\}$.

This and the subsequent genes are meaningful for the current individual in the population if the first gene allele is equal to 1.

- (8) This gene encodes the index i_1 of the component series which is to be considered as the first element of the vector threshold variable, $i_1 \in \{1, 2, \ldots, K\}$.
- (9) The second element i_2 of the vector threshold variable, $i_2 \in \{1, 2, \ldots, K\}, i_2 \neq i_1$.
- (10) Position of the threshold for the first component series. The encoding follows the same rules as for genes (4-6).
- (11) Position of the threshold for the component series used as a second element in the threshold vector. The same rules as before are used for encoding.
- (12) Delay d_1 for the first element of the vector threshold variable, $d_1 \in \{1, 2, \ldots, d_{\max}\}$.
- (13) Delay d_2 for the second element of the vector threshold variable, $d_2 \in \{1, 2, \ldots, d_{\max}\}$.
- (14) This gene is a binary digit. If it is equal to 1 then two regions in the partition induced by the vector threshold variable in the space of the values of the MSETAR model may merge, and the number of regimes is determined by following gene (15). Otherwise the number of regimes remains 4 as depicted in Fig. 2.2.
- (15) This gene specifies which of the regions merge together. With reference to Fig. 2.2, values are:
    - (1) the regimes I and II merge and the number of regimes is 3,
    - (2) the regimes III and IV merge and the number of regimes is 3,
    - (3) the regimes I and III merge and the number of regimes is 3,
    - (4) the regimes II and IV merge and the number of regimes is 3,
    - (5) the regimes I and IV merge and the number of regimes is 3,
    - (6) the regimes II and III merge and the number of regimes is 3,
    - (7) the regimes I merges with IV and II merges with III and the number of regimes is 2.

The encoding as defined above is rather elaborated and requires a special decoding algorithm. In addition, special algorithms have to be designed for the computation of the fitness function in the selection step, and non standard crossover and mutation operators are needed. However, this does not impacts too much the overall computational burden provided that each one of the decoding steps are carefully programmed.

For example, let us consider the following chromosome, which is intended to encode a K-dimensional MSETAR with $K = 4$ and 2-dimensional threshold variable. For the sake of simplicity the genes whose alleles are integer numbers are written as integers, though their internal representation is a binary string, for instance the integer 3 in the third genes is reserved three bits so that it is actually encoded as 011.

1	1	3	180	100	50	1	**1**	**3**	**40**	**120**	**1**	**1**	**0**	3

The first gene denotes that the threshold variable is bivariate so the decoding continues at locus 8. The components indexed as 1 and 3 are to be assumed as threshold variables (8-9). The thresholds values have to be taken equal to the 40-th observation of the first component and the 120-th observation of the third component, i.e. $r_1^{(1)} = y_{1,40}$ and $r_1^{(2)} = y_{3,120}$. The delay parameters follow equal to 1 for both threshold variable components, which is $Y_{t-d} = (y_{1,t-1}, y_{3,t-1})'$. The allelic value in locus 14 means that we don't allow regions defined by the thresholds to merge, so the number of regimes is equal to 4. The last gene may be neglected.

2.3.2 Fitness function

The fitness function measures the adaptation of the individual to the environment. In the present context the chromosome of each individual encodes a MSETAR model which is to be considered as better as smallest its AIC index. A transform of the AIC may be used to obtain positive fitness function values so that the optimization problem may be put in terms of maximization of the fitness function as it is usual in the GAs. So let

$$\text{Fitness} = \exp\{-\text{AIC}\}, \tag{2.3}$$

where

$$\begin{aligned}
\mathrm{AIC} &= \frac{1}{n}\sum_{j=1}^{\ell} \mathrm{AIC}_j, \\
\mathrm{AIC}_j &= n_j \log\left\{\det(\tilde{\Sigma}_u^{(j)})\right\} + 2m_j K^2, \qquad (2.4) \\
\tilde{\Sigma}_u^{(j)} &= \frac{1}{n_j}\sum_t \hat{u}_t^{(j)}(\hat{u}_t^{(j)})'.
\end{aligned}$$

In Eqn. (2.4) the number of regimes is set equal to ℓ, while the number of observations in the j-th regime is n_j, with $n = \sum_j n_j$ the total number of observations, and $\{\hat{u}_t^{(j)}\}$ are the estimated model residuals in regime j.

2.3.3 Selection

Basically the well known 'roulette wheel rule' is used for selecting from the current population the individuals candidate for inclusion in the next generation. The roulette wheel rule amounts to choose individuals with probability proportional to their respective fitness function value. The widespread usage of this rule explains the reason why in GAs the fitness function is usually constrained to positive values as otherwise such rule would be impractical. Individuals are allowed to be selected more than once and the number of choices is a fraction Gs of the population size s, G being the generational gap. The elitist strategy is adopted as a correction of this rule that ensures asymptotical convergence and constrains the fitness to be a non decreasing function of the iteration number. The elitist strategy may be implemented either directly or indirectly by setting $G < 1$ and choosing deterministically, i.e. the best ones or even the single best one, the $(1 - G)s$ individuals that are selected outside the intervention of the roulette wheel rule mechanism. Normalization of the fitness function may be used for scaling the transform (2.3) in such a way the selection probabilities defined by the roulette wheel rule are close each other. For instance, the 'sigma truncation scaling' consists in applying the normalization transform

$$\mathrm{Fitness}^* = \mathrm{Fitness} - \left(\bar{F} - c\sigma\right),$$

where $\bar{F}$ is the population mean, c is a suitable real positive constant and σ the standard deviation, and in truncating the low fitness individuals.

2.3.4 Crossover and mutation

The general crossover operator generates new individual chromosomes according to the following rules:

- Pairs of individuals randomly chosen mate and produce a pair of offsprings that may share genes of both parents.
- This operator is applied with a fixed probability (usually larger than 0.5 but smaller than one) to each pair.
- Several different types of crossover are common, the simplest is called one point crossover.
    - A same locus in the chromosomes of the two paired individuals is chosen at random: the genes which appear before that locus remain unchanged, while the genes appearing after the crossover point are exchanged.
    - This operation applies to each binary field in the chromosome.

As for mutation, general criteria may be the following:

- Mutation is needed to introduce innovation into the population (since selection and crossover only mix the existing genes)
- It is generally considered a rare event (like it is in nature).
- A small probability p_m is selected, usually less than 0.1, and each gene of each individual chromosome is subject to mutation with probability p_m, independently of all other genes.
- If the gene coding is binary, for instance, a mutation simply changes 0 to 1 or vice versa.

The new generation is created by selecting individuals from both the parent generation and the offspring generation. There are several alternative methods for replacing population individuals with new offsprings, e.g. 'crowding' (de Jong 1975). As a matter of fact there are two objectives that seem most important to define the transition from the past generation to the new one, i.e. to maintain diversity among the individuals and to avoid that the population is biased towards the best individual. The two objectives seem reasonable as we have to avoid simultaneously both premature convergence to some local optimum and poor or limited exploration of the solution space, i.e. the set of all feasible potential solutions. Many different techniques that we may adopt to deal with these problems have been proposed in the literature and allow suitable modifications of the standard rules for choosing the individuals that have to be included in the next current population.

2.3.5 Convergence of genetic algorithms

If GAs are employed as optimization methods we are concerned with the problem of defining in probability terms how close the best solution found in the last iteration is to the actual optimum. Let $x_{\text{best}}^{(g)}$ be the chromosome of the fittest individual found at generation g, then $\{f[x_{\text{best}}^{(g)}], g = 1, 2, \ldots\}$ defines a sequence of random variables. Jennison & Sheehan (1995) provided a revised updated version of the 'schema theorem'. Rudolph (1997) demonstrated theorems concerned with global optimum convergence of GAs in an elitist strategy framework. The Markov chains theory offers some insights into the asymptotic convergence property of GAs, here we only recall a result for chromosomes composed of genes that take binary allelic values. Let each chromosome have M binary genes and let the population be composed by s individuals. The possible populations are $\binom{s+2^M-1}{s}$ (combinations with repetition of the 2^M possible different individuals in sets of cardinality s). Though very large, the number of states of the process is finite, and it may be considered a finite Markov chain. Then suppose that there is only an optimal individual, coded by chromosome y. Let j denote the state corresponding to the population composed of all individuals equal to y: the transition matrix P has a 1 in the diagonal at position j, it is an absorbing state and convergence is certain. Details and a complete discussion may be found e.g. in Rudolph (1997), Reeves & Rowe (2003).

2.4 A simulation experiment

To evaluate the performance of the GA, we simulated three MSETAR models discarding the first 500 observations to avoid any initialization effects. From the first two models we simulated 100 replications each with 150, 400 and 1000 observations. For the last model we simulated 100 replications each with 400, 600 and 1000 observations. The number of observations has been chosen so that enough observations fall in each regime. For the first two models (Eqn.s (2.5) and (2.6)) the regimes are defined by only a single partition of the real axis for the first component of the process, that is the current regime is exclusively determined by the first component. For the third model (Eqn. (2.7)) the regimes are defined by a partition of $\mathbb{R} \times \mathbb{R}$ and both component series provide the bivariate threshold variable. The GA parameters have been chosen 100 the population size, 1000 the number of generations, 0.9 the crossover probability and 0.01 the mutation probability. The maximum VAR order is $p_{\max} = 4$ and the maximum delay is $d_{\max} = 10$.

The evaluation of the procedure performance is concerned with three aspects, i.e. (1) correct selection of threshold variable, (2) correct specification of threshold values and number of regimes, and (3) accuracy of the parameter estimates.

2.4.1 Example 1

In the first simulation experiment we consider time series generated by the MSETAR model (Tsay 1998)

$$Y_t = \begin{cases} \Phi_1^{(1)} Y_{t-1} + U_t^{(1)} & y_{1,t-1} \leq 0 \\ \Phi_1^{(2)} Y_{t-1} + U_t^{(2)} & y_{1,t-1} > 0 \end{cases} \tag{2.5}$$

where

$$\Phi_1^{(1)} = \begin{bmatrix} 0.7 & 0.0 \\ 0.3 & 0.7 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 1.0 & 0.2 \\ 0.2 & 1.0 \end{bmatrix}, \qquad \Phi_1^{(2)} = \begin{bmatrix} -0.7 & 0.0 \\ -0.3 & -0.7 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 1.0 & -0.3 \\ -0.3 & 1.0 \end{bmatrix}.$$

The innovations $U_t^{(1)}$ e $U_t^{(2)}$ are independent multivariate normal with mean 0 and variance Σ_1 and Σ_2 respectively. The threshold variable is considered to be the first entry of the series with delay parameter equals to one. The threshold value is set equal to zero.

In Table 2.1 the percentages of correct identification over 100 replications of the number of regimes and of the threshold variable are shown. The label 'Thr.Var' denotes the correct selection of the component series that is used as threshold variable. 'Delay' label denotes the lag of the threshold variable. The label 'N.Reg.' denotes the number of regimes. The results displayed in Table 2.1 show that detection of the threshold variable and identification of the number of regimes and delay are performed satisfactorily. The percentages are greater than 88%.

In Table 2.2 the average bias and root mean square error (RMSE) of the estimates of coefficients and threshold parameters for Model (2.5) are displayed. Only the estimates from the replications where exact match of structural parameters (variable threshold and number of regimes) occurred are considered. In this case we can see

Table 2.1: Relative frequency of correctly selecting the component series which performs as threshold variable, the delay parameter and the number of regimes for sample sizes 150, 400 and 1000 observations, based on 100 replications

$n = 150$			$n = 400$			$n = 1000$		
Thr.Var	Delay	N.Reg.	Thr.Var	Delay	N.Reg.	Thr.Var	Delay	N.Reg.
100	88	91	96	100	96	100	100	100

Table 2.2: Average bias and RMSE over 100 replications of the estimates of the autoregressive coefficients and threshold parameter based on sample size of 150, 400 and 1000 observations

Coefficient	$n = 150$		$n = 400$		$n = 1000$	
	bias	RMSE	bias	RMSE	bias	RMSE
$\phi_{11}^{(1)}$	0.0270	0.1080	0.0018	0.0107	-0.0115	0.0123
$\phi_{21}^{(1)}$	0.0241	0.1259	-0.0181	0.0460	-0.0099	0.0137
$\phi_{12}^{(1)}$	-0.0703	0.1248	0.0469	0.0541	-0.0038	0.0081
$\phi_{22}^{(1)}$	-0.0360	0.2204	0.0196	0.0887	-0.0069	0.0083
$\phi_{11}^{(2)}$	0.0457	0.1681	-0.0226	0.0422	0.0014	0.0047
$\phi_{21}^{(2)}$	0.0844	0.2198	0.0430	0.0596	0.0282	0.0288
$\phi_{12}^{(2)}$	0.0752	0.1640	0.0540	0.0610	-0.0006	0.0090
$\phi_{22}^{(2)}$	-0.0323	0.1290	-0.0172	0.0498	0.0174	0.0188
r^*	-0.0231	0.2311	-0.0164	0.1404	-0.0065	0.0185

that the estimated coefficients are quite accurate, i.e. they are close on the average to their true values. The accuracy of the estimates improves as the sample size increases. It has to be considered that our GA method does not aim at estimating the exact threshold parameter but at detecting the observation that divides the time series in the two regimes. If we consider the misplaced observations, it results that these are, on the average and for sample size $n = 150$, $n = 400$ and $n = 1000$ respectively, 13%, 8% and 3%. So the assignment of observations to regimes may be considered quite satisfactory and more accurate as larger the sample size, even in the presence of rather large RMSE for $n = 150$ and $n = 400$.

Table 2.3: Relative frequency (based on 100 replications) of selecting correctly the index of the component to be used as threshold variable, the delay parameter and the number of regimes for sample sizes 150, 400 and 1000 observations

$n = 150$			$n = 400$			$n = 1000$		
Thr.Var	Delay	N.Reg.	Thr.Var	Delay	N.Reg.	Thr.Var	Delay	N.Reg.
100	94	93	94	97	89	100	100	100

2.4.2 Example 2

The second simulation experiment is concerned with the MSETAR model (Tsay 1998)

$$Y_t = \begin{cases} \Phi_1^{(1)} Y_{t-1} + U_t^{(1)} & y_{1,t-1} \leq -3.3 \\ \Phi_1^{(2)} Y_{t-1} + U_t^{(2)} & -3.3 < y_{1,t-1} \leq 3.3 \\ \Phi_1^{(3)} Y_{t-1} + U_t^{(3)} & y_{1,t-1} > 3.3 \end{cases} \tag{2.6}$$

where

$$\Phi_1^{(1)} = \begin{bmatrix} -0.7 & 0.0 \\ 0.2 & -0.9 \end{bmatrix} \qquad \Phi_1^{(2)} = \begin{bmatrix} 1.2 & 0.0 \\ 0.0 & 0.6 \end{bmatrix} \qquad \Phi_1^{(3)} = \begin{bmatrix} -0.8 & 0.0 \\ 0.2 & 0.8 \end{bmatrix} \qquad \Sigma_1 = \Sigma_2 = I.$$

The innovations $U_t^{(1)}$ e $U_t^{(2)}$ are independent multivariate normal with mean 0 and variance $\Sigma_j = I, j = 1, 2$ where I denotes the identity matrix. The model has three regimes and the first component of the bivariate series with delay parameter 1 determines the current regime. The threshold values are -3.3 and 3.3.

The percentages of correct identification of the number of regimes and threshold component over 100 replications are summarized in Table 2.3. From Table 2.3 we may observe that our GAs-based procedure determines the correct threshold variable and number of regimes with high percentages which increase as the sample size is larger.

In Table 2.4 the estimates for Model (2.6) are reported. The estimates were considered only for the replications where exact match of structural parameters (excluding thresholds) occurred (about 90%). Bias and RMSEs seem rather small and decrease as the sample size increases, but both bias and RMSE of the estimates of the thresholds $r_1^{(1)}$ and $r_1^{(2)}$. However, if we consider again the number of misplaced

Table 2.4: Average bias and RMSE over 100 replications of the estimates of the autoregressive coefficients and threshold parameters based on sample sizes 150, 400 and 1000 observations

Coefficient	$n = 150$		$n = 400$		$n = 1000$	
	bias	RMSE	bias	RMSE	bias	RMSE
$\phi_{11}^{(1)}$	0.0685	0.1606	0.0313	0.0503	0.0168	0.0170
$\phi_{21}^{(1)}$	-0.0342	0.2200	-0.0224	0.0312	-0.0023	0.0062
$\phi_{12}^{(1)}$	0.0504	0.1914	0.0057	0.0320	0.0009	0.0015
$\phi_{22}^{(1)}$	0.0563	0.1825	0.0063	0.0226	0.0029	0.0119
$\phi_{11}^{(2)}$	-0.0460	0.1556	0.0137	0.0316	0.0050	0.0984
$\phi_{21}^{(2)}$	-0.0333	0.2284	0.0022	0.1021	-0.0002	0.0874
$\phi_{12}^{(2)}$	0.0603	0.1600	0.0086	0.0170	-0.0094	0.0098
$\phi_{22}^{(2)}$	-0.0198	0.1352	0.0085	0.0353	0.0056	0.0077
$\phi_{11}^{(3)}$	-0.0271	0.1136	-0.1103	0.1107	0.0167	0.0168
$\phi_{21}^{(3)}$	-0.0853	0.1366	-0.0330	0.0660	-0.0121	0.0131
$\phi_{12}^{(3)}$	-0.0031	0.1854	-0.0011	0.0220	-0.0004	0.0041
$\phi_{22}^{(3)}$	0.0240	0.2656	-0.0332	0.0351	0.0895	0.0896
$r_1^{(1)}$	-0.3791	0.4329	-0.2060	0.2222	-0.2916	0.2916
$r_1^{(2)}$	0.1668	0.1909	0.3336	0.3350	0.3105	0.3105

Table 2.5: Relative frequency of correctly selecting the threshold variable, delay parameter and number of regimes for sample sizes of 400, 600 and 1000 observations based on 100 replications

$n = 400$			$n = 600$			$n = 1000$		
Thr.Var	Delay	N.Reg.	Thr.Var	Delay	N.Reg.	Thr.Var	Delay	N.Reg.
79	72	79	89	78	84	93	90	91

observations we obtain the percentages $11\%, 10\%, 4\%$ for $n = 150$, $n = 400$ and $n = 1000$ respectively. This circumstance seems to indicate that in this case too the assignment of observations to regimes has been performed rather satisfactorily.

2.4.3 Example 3

In the third simulation experiment we consider time series generated according to the model

$$Y_t = \begin{cases} \Phi_1^{(1)} Y_{t-1} + U_t^{(1)} & y_{1,t-1} > 0 \quad y_{2,t-1} \leq 0 \\ \Phi_1^{(2)} Y_{t-1} + U_t^{(2)} & y_{1,t-1} > 0 \quad y_{2,t-1} > 0 \\ \Phi_1^{(3)} Y_{t-1} + U_t^{(3)} & y_{1,t-1} \leq 0 \quad y_{2,t-1} \leq 0 \\ \Phi_1^{(4)} Y_{t-1} + U_t^{(4)} & y_{1,t-1} \leq 0 \quad y_{2,t-1} > 0 \end{cases} \tag{2.7}$$

where

$$\Phi_1^{(1)} = \begin{bmatrix} 0.7 & -0.2 \\ -0.1 & 0.6 \end{bmatrix} \qquad \Phi_1^{(2)} = \begin{bmatrix} 0.5 & -0.4 \\ 0.1 & 0.3 \end{bmatrix}$$

$$\Phi_1^{(3)} = \begin{bmatrix} -0.5 & 0.2 \\ -0.1 & 0.5 \end{bmatrix} \qquad \Phi_1^{(4)} = \begin{bmatrix} -0.5 & -0.9 \\ 0.8 & -0.1 \end{bmatrix} \qquad \Sigma_j = I, j = 1, \ldots, 4.$$

The $U_t^{(j)}$ are independent bivariate normal random variables with mean 0 and variance $\Sigma_j = I, j = 1, \ldots, 4$ where I denotes the identity matrix. The model has four regimes which depend on the lagged component series with delay equal to 1. The threshold values are equal to 0 for both threshold components.

The percentages of replications for which the correct threshold variable and number of regimes were selected are given in Table 2.5. The results displayed in Table 2.5

Table 2.6: Average bias and RMSE over 100 replications of the estimates of the autoregressive coefficients and threshold parameters based on sample sizes 400, 600 and 1000 observations

Coefficient	$n = 400$		$n = 600$		$n = 1000$	
	bias	RMSE	bias	RMSE	bias	RMSE
$\phi_{11}^{(1)}$	0.0311	0.1909	-0.0218	0.1058	0.0198	0.0225
$\phi_{21}^{(1)}$	0.0112	0.1743	0.0182	0.0334	0.0046	0.0060
$\phi_{12}^{(1)}$	0.0676	0.1504	-0.0263	0.0958	0.0059	0.0064
$\phi_{22}^{(1)}$	0.0157	0.1860	-0.0146	0.0363	-0.0002	0.0085
$\phi_{11}^{(2)}$	-0.0441	0.1828	0.0021	0.0341	-0.0080	0.0101
$\phi_{21}^{(2)}$	-0.0778	0.2012	-0.0348	0.0437	-0.0051	0.0073
$\phi_{12}^{(2)}$	0.0430	0.1922	-0.0046	0.0130	-0.0034	0.0054
$\phi_{22}^{(2)}$	-0.0495	0.2073	-0.0272	0.0809	0.0008	0.0084
$\phi_{11}^{(3)}$	0.0360	0.1690	0.0368	0.1068	-0.0155	0.0188
$\phi_{21}^{(3)}$	0.0193	0.1383	0.0183	0.0283	0.0054	0.0063
$\phi_{12}^{(3)}$	-0.0455	0.2052	-0.0377	0.0398	0.0070	0.0070
$\phi_{22}^{(3)}$	0.0212	0.1851	-0.0089	0.0569	0.0015	0.0064
$\phi_{11}^{(4)}$	0.0360	0.1411	0.0368	0.0564	-0.0155	0.0172
$\phi_{21}^{(4)}$	-0.0208	0.1202	0.0060	0.0376	-0.0027	0.0061
$\phi_{12}^{(4)}$	0.0306	0.1780	-0.0294	0.0887	0.0169	0.0175
$\phi_{22}^{(4)}$	0.0304	0.1908	-0.0177	0.0313	0.0175	0.0179
$r_1^{(1)}$	-0.0097	0.1341	-0.0063	0.0666	-0.0040	0.0041
$r_1^{(2)}$	-0.0022	0.1173	-0.0036	0.0059	-0.0002	0.0037

show that the exact recovery of the threshold variable and number of regimes seems more difficult for models with bivariate threshold variable, and percentages of success greater than 90% are attained only if $n = 1000$ whereas percentages of exact match are below 90% if $n = 400$ and $n = 600$. Detection of structural parameters is performed satisfactorily by the GAs-based procedure if $n = 1000$ while convergence seems slow if only $n = 400$ or $n = 600$ observations are available.

In Table 2.6 the average bias and RMSE of the estimates of coefficients and thresholds for Model (2.7) are displayed. Only the estimates from the replications where exact match of structural parameters (except thresholds) occurred (more than 70%) are considered. In this case, too, the estimated coefficients are quite accurate, i.e. they are close on the average to their true values. Both bias and RMSEs decrease as the sample size increases.

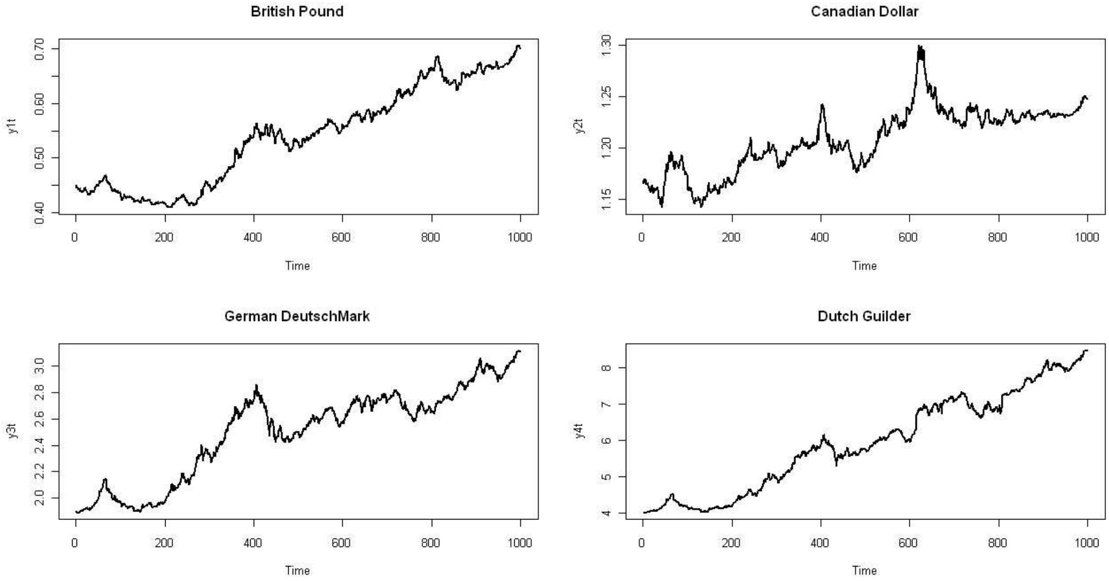

Figure 2.2: Exchange Rate Data

2.5 An application to real world data

As an illustration, we applied the MSETAR model to study an exchange rate data set. Exchange rate data have be found to exhibit a non linear behavior and many non linear models have been suggested which include univariate threshold models (e.g., Chappell et al. 1996), and univariate threshold GARCH models (e.g., Baragona & Cucina 2008). The exchange rates are the British pound, Canadian dollar, German Deutschmark, Dutch guilder, all expressed as number of units of the foreign currency per United States dollar. The time frame of the study is January 1980 to March 1984. Then there are 1000 observations. The data are daily data. The plot of the components time series are displayed in Fig. 2.2.

We run our GAs-based procedure with the same parameters used in the simulation experiment in Section 2.4. The final estimated model is a two-regime MSETAR with the following form:

$$Y_t = \begin{cases} \Phi_1^{(1)} Y_{t-1} + U_t^{(1)} & y_{1,t-1} \leq 0.5770 \\ \Phi_1^{(2)} Y_{t-1} + U_t^{(2)} & y_{1,t-1} \geq 0.5770 \end{cases}$$

where

$$\Phi_1^{(1)} = \begin{bmatrix} 0.9772 & -0.0065 & -0.1173 & -0.2084 \\ 0.0021 & 1.0015 & 0.0270 & 0.0464 \\ 0.0004 & -0.0022 & 0.9868 & -0.0403 \\ 0.0015 & 0.0013 & 0.0110 & 1.0282 \end{bmatrix}$$

$$\Phi_1^{(2)} = \begin{bmatrix} 0.9994 & 0.0117 & -0.0255 & 0.0751 \\ -0.0026 & 0.9949 & 0.0738 & 0.0588 \\ 0.0029 & -0.0018 & 0.8974 & -0.1301 \\ -0.0006 & 0.0005 & 0.0289 & 1.0335 \end{bmatrix}.$$

The number of observations in each regime are 644 and 355. The driving variable is the British pound which determines the regime switch for the exchange rates. The critical exchange rate is the value 0.57 when the British pound approximately doubles the value of the United States dollar. The goodness of fit of the estimated model may be considered satisfactory on the basis of the residual variances that are, on the entire time span, 0.0000119, 0.0000087, 0.0002587, and 0.0018356 for each of the four component series respectively.

Chapter 3

Meta-heuristic Methods for Outliers Detection in Multivariate Time Series

3.1 Introduction

Outliers are commonly defined as observations which appear to be inconsistent with the remainder of the data set, and may be due to occasional unexpected events. The detection of outliers is an important problem in time series analysis because they can have adverse effects on model identification, parameter estimation (see Chang & Tiao (1983)) and forecasting (see Chen & Liu (1993)). The presence of just a few items of anomalous data can lead to model misspecification, biased parameter estimation, and poor forecasts. Therefore, it is essential to identify outliers data, estimate their magnitude and correct the time series, avoiding false identifications (i.e. observations that are identified as outliers while they are not). Several approaches have been proposed in the literature for handling outliers in univariate time series. Among these methods we can distinguish those based on an explicit model (parametric approach) from the ones using non-explicit models (nonparametric approach). For the parametric approach, Fox (1972) developed a likelihood ratio test for detecting outliers in a pure autoregressive model. Chang & Tiao (1983), Chang et al. (1988), Tsay (1986, 1988), Chen & Liu (1993) extended this test to an autoregressive integrated moving-average (ARIMA) model and proposed an iterative procedure for detecting multiple outliers. For the non-parametric approach, Ljung (1989), Ljung (1993), Peña (1990), Gómez et al. (1993), Baragona & Battaglia (1989) and Battaglia & Baragona (1992) proposed specific procedures based on the relationship between additive outliers and linear interpolator, while Baragona et al. (2001) used a genetic algorithm.

For multivariate time series, only three procedures have been proposed. Tsay et al. (2000) proposed a sequential detection procedure, which we will call the TPP method, based on individual and joint likelihood ratio statistics; this method requires an initial specification of a vector ARMA model. Galeano et al. (2006), Baragona & Battaglia (2007) proposed a method based on univariate outlier detection applied to some useful linear combinations of the vector time series. The optimal combinations are found by projection pursuit in the first paper and independent component analysis (ICA) in the second one. Barbieri (1991) used a Bayesian method and finally a graphical method was explored by Khattree & Naik (1987).

Multiple outliers, especially those occurring close in time, often have severe masking effect (one outlier masks a second outlier) and smearing effect (misspecification of correct data as outliers) that can easily render the iterative outlier detection methods inefficient. A special case of multiple outliers is a patch of additive outliers. For univariate time series this problem has been addressed firstly by Bruce & Martin (1989). They define a procedure for detecting outlier patches by detecting blocks of consecutive observations. Other useful references for the patch detection are McCulloch & Tsay (1994), Barnett et al. (1997) and Justel et al. (2001). For multivariate time series, only Baragona & Battaglia (2007) report simulation results for an outlier patch.

Unlike the univariate case where there are specific procedures on the identification of consecutive outliers, in multivariate time series framework, methods for identification of consecutive outliers do not exist.

We propose a class of meta-heuristic algorithms to overcome the difficulties of iterative procedures in detecting multiple additive outliers in multivariate time series. This class includes: simulated annealing (SA)(Kirkpatrick et al. (1983), Rayward-Smith et al. (1996)), threshold accepting (TA) (Winker (2001)) and genetic algorithm (GA) (Holland (1975); Goldberg (1989)). These methods are illustrated in appendix. Our procedures are less vulnerable to the masking and smearing effects because they evaluate several outlier pattern where all observations that are possibly outlying ones are simultaneously considered. In this way, meta-heuristic methods deal efficiently the detection of patch of additive outliers.

Each outlier configuration is evaluated by a generalised AIC-criterion where the penalty constant is suggested by both a simulation study and a theoretical approximation. So, the meta-heuristic algorithms seem able to provide more flexibility and adaptation to the outlier detection problem.

3.2 Algorithm Features

This section further describes the algorithms implementation we used for outlier detection. A successful implementation of meta-heuristic methods is certainly crucial to obtain satisfactory results. Before a meta-heuristic method can be applied to a problem some important decisions have to be made. The three meta-heuristic methods require a suitable encoding for the problem and an appropriate definition of objective function. In addition, the algorithms TA and SA require the structure of the neighborhood while for genetic algorithms, operators of selection, crossover and mutation have to be chosen. The following sections describe the choices made.

3.2.1 Solution Encoding

An appropriate encoding scheme is a key issue for meta-heuristic methods. For all algorithms we use a binary encoding for the solutions of the outliers problem as suggested in Baragona et al. (2001). Any solution ξ^c is a binary string of length N, where N is the number of observations of the time series: $\xi^c = (\xi_1^c, \xi_2^c, \ldots, \xi_N^c)$, where ξ_i^c takes the value 1 if at time i there is an outlier (we assume that all the s components are influenced) and 0 otherwise. Then, ξ^c represent a chromosome of GA and ξ_i^c a gene. Obviously, the number of outliers for a given time series is unknown. We allow for solutions with a maximum number of outliers equal to g. The value of g should be chosen according to the series length and every relevant a priori information on its accuracy and instability. The constant g should be chosen large enough to allow for the detection of any reasonable number of outliers in the series.

Binary encoding implies that the solution space Ω consists of $\sum_{k=0}^{g} \binom{N}{k}$ distinct elements, since the total number of outliers is limited to a constant g.We can see that Ω is really large even when g is considerably lower than the length of the time series. All our algorithms either severely penalise solutions with a maximum number of outliers larger then g , or do not consider such solutions at all. TA and SA algorithms are built so that they do not evaluate solutions with more than g outliers. With regard to the GA, chromosomes not belonging to Ω will be severely penalised subtracting a positive quantity (the penalty factor *pen*) to the fitness (function to be maximised), so that the algorithm tends to avoid these chromosomes. We set the value of *pen* to 1,000.

3.2.2 Neighbourhood search in simulated annealing and threshold accepting

Each solution $\xi^c \in \Omega$ has an associated set of *neighbours*, $N(\xi^c) \subset \Omega$, called the neighbourhood of ξ^c where every $\xi^n \in N(\xi^c)$ may be reached directly from ξ^c by an operation called *move*.Given the current solution ξ^c, its neighborhood is constructed using three different moves: add an outlier; remove an outlier; change the position of an outlier.Since a maximum of g outliers is allowed, moves are applied according to the current solution in the following way: if ξ^c doesn't contain outliers (i.e., it is a string where every bit is 0), algorithms can only introduce an outlier; if ξ^c contains less than g outliers, algorithms can add, remove or change the position of an outlier, with probability 1/3; if ξ^c already contains g outliers, algorithms cannot proceed adding an outlier but can only remove or change the position of one of them, with probability 1/2.

3.2.3 Objective function

Let $y_t = [y_{1,t}, \ldots, y_{s,t}]'$ be a vector time series generated from a Gaussian s-dimensional jointly second order stationary real-valued process Y_t, with mean zero for each component, covariance matrix $\mathbf{\Gamma}_u$ and inverse covariance matrix $\mathbf{\Gamma i}_u$ for integer lag u. When outliers are present, y_t is perturbed and unobservable. We suppose that k perturbations $\omega_t = [\omega_{1,t}, \ldots, \omega_{s,t}]'$ impact the series y_t at time points t_j, $j = 1, \ldots, k$ such that at each t_j they affect all s components. The total number of outlying data is equal to $h = ks$. Denote the observed time series by $z_t = [z_{1,t}, \ldots, z_{s,t}]'$ generated by the observable multivariate stochastic process Z_t. Given a sample of N observations we may write the following model

$$z = y + \mathbf{X}\omega, \tag{3.1}$$

where $z = [z_1', \ldots, z_N']'$ is the vector obtained by stacking the s component observations at each time point, $y = [y_1', \ldots, y_N']'$ is the vector obtained by stacking the s component of the unobservable outlier free time series at each time point, $\omega = [\omega_{t_1}', \ldots, \omega_{t_k}']'$ is the vector obtained by stacking the s components of the k outliers and $\mathbf{X}$ is a $Ns \times h$ pattern design matrix defined as follows.

For each t_j with $j = 1, \ldots, k$, the $[(t_j - 1)s + r, (j - 1)s + r]$-th entry is one for $r = 1, \ldots, s$. All the remaining entries are zero.

Matrix $\mathbf{X}$ contains information about the perturbed time indices of a given outlier pattern. Thus, each feasible solution ξ corresponds to a matrix $\mathbf{X}$.

The natural logarithm of the likelihood for z may be written

$$L_{(z;\mathbf{X},\omega)} = -\frac{Ns}{2}\log(2\pi) - \frac{1}{2}\log(\det\boldsymbol{\Gamma}) - \frac{1}{2}(z-\mathbf{X}\omega)'\boldsymbol{\Gamma}^{-1}(z-\mathbf{X}\omega), \tag{3.2}$$

where $\boldsymbol{\Gamma}$ denotes the $Ns \times Ns$ block Toeplitz matrix with $\boldsymbol{\Gamma}_{i-j}$ as the (i,j)-th block. Assuming both $\boldsymbol{\Gamma}$ and $\mathbf{X}$ known, the maximisation of (3.2) with respect to ω yields:

$$\hat{\omega} = (\mathbf{X}'\boldsymbol{\Gamma}^{-1}\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Gamma}^{-1}z. \tag{3.3}$$

If we approximate $\boldsymbol{\Gamma}^{-1}$ with $\boldsymbol{\Gamma}\mathbf{i}$ (Shaman (1976)), where $\boldsymbol{\Gamma}\mathbf{i}$ denotes the $Ns \times Ns$ block Toeplitz matrix with $\boldsymbol{\Gamma}\mathbf{i}_{i-j}$ as the (i,j)-th block, the maximum likelihood estimate (3.3) of ω takes the form:

$$\hat{\omega} = (\mathbf{X}'\boldsymbol{\Gamma}\mathbf{i}\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Gamma}\mathbf{i}z. \tag{3.4}$$

Since $\boldsymbol{\Gamma}\mathbf{i}$ is unknown, we have to estimate it from the data. We used here the autoregressive approach described in section (1.3.3).

If we look at the expression (1.37) can see that the estimate of the inverse covariance depends on estimates of autoregressive parameters and the estimated variance-covariance matrix $\hat{\boldsymbol{\Sigma}}$ of innovations. In the presence of outliers the residuals of VAR model are contaminated, hence $\hat{\boldsymbol{\Sigma}}$ may be biased. For obtaining a better estimate we use the $\alpha\%$ trimmed method. To compute the $\alpha\%$ trimmed variance-covariance matrix $\hat{\boldsymbol{\Sigma}}$, we first remove the 5% largest values (according to their absolute values) and then compute $\hat{\boldsymbol{\Sigma}}$ based on trimmed sample.

The natural logarithm of the maximised likelihood is obtained by replacing ω by $\hat{\omega}$ and $\boldsymbol{\Gamma}^{-1}$ by $\hat{\boldsymbol{\Gamma}}\mathbf{i}$ in (3.2) :

$$\hat{L}_{(z;\mathbf{X},\omega)} = -\frac{Ns}{2}\log(2\pi) - \frac{1}{2}\log(\det\hat{\boldsymbol{\Gamma}}\mathbf{i}) - \frac{1}{2}z'\hat{\boldsymbol{\Gamma}}\mathbf{i}z - \frac{1}{2}(\mathbf{X}'\hat{\boldsymbol{\Gamma}}\mathbf{i}z)'(\mathbf{X}'\hat{\boldsymbol{\Gamma}}\mathbf{i}\mathbf{X})^{-1}\mathbf{X}'\hat{\boldsymbol{\Gamma}}\mathbf{i}z. \tag{3.5}$$

The matrix $\hat{\boldsymbol{\Gamma}}\mathbf{i}$ is fixed for any outlier pattern $\mathbf{X}$, so that the maximised likelihood in (3.5) depends only on matrix $\mathbf{X}$. Since matrix $\mathbf{X}$ conveys all information about the outlier's location, it seems natural to detect the outlier pattern by determining the matrix $\mathbf{X}$ maximising the quadratic form in (3.4)

$$L_{\mathbf{X}} = \frac{1}{2}(\mathbf{X}'\hat{\boldsymbol{\Gamma}}\mathbf{i}z)'(\mathbf{X}'\hat{\boldsymbol{\Gamma}}\mathbf{i}\mathbf{X})^{-1}\mathbf{X}'\hat{\boldsymbol{\Gamma}}\mathbf{i}z. \tag{3.6}$$

Obviously the likelihood increases when the number of estimated parameters $\hat{\omega}$, i.e. the number of outliers, is increased. Thus, in a similar fashion as identification criteria for model selection (see Bhansali & Downham (1977)), we contrast the likelihood with a linear function of the number of outliers. So, the search of outliers in a multivariate series is equivalent to search the chromosome ξ or the design matrix X that minimizes the following objective function:

$$f(\xi) = -2L_{\mathbf{X}} + ch, \tag{3.7}$$

where c is an arbitrary constant and h is the actual number of outliers. The function $f(\xi)$ depends on both the matrix $\mathbf{X}$ and the penalty constant c. Different values are suggested in literature for the constant c (see Bhansali & Downham (1977)). We propose two alternative approaches for selecting appropriate c values in Section (3.4.1). In a genetic algorithm, the fitness function assigns a positive real number to any possible solution in order to evaluate its plausibility, therefore in the GA we adopt the following non-decreasing transform of (3.7):

$$fitness = exp(-f(\xi)/\beta) \tag{3.8}$$

where β is a parameter of scale. In the following experiments this parameter is set equal to 100.

3.2.4 Cooling schedules

The choice of a schedule is a discussed issue as there was a conflict, since early applications of SA, between theory (logarithmic coolings) and practice (geometric schedules). No universally valid conclusion seems to emerge from the literature. A general advice is however to cool the system slowly enough at stages where the objective function is rapidly improving. An appropriately tuned geometric schedule seems able to satisfy this requirement and yields good results in a reliable manner. Then, in our work the geometric schedule is used :

$$T_t = aT_{t-1}, \tag{3.9}$$

where a is a constant close to 1.

This schedule assumes that the annealing process will continue until the temperature reaches zero. In practise, it is not necessary to let the temperature reach zero because as it approaches zero the chances of accepting a worse move are almost the same as the temperature being equal to zero. Therefore, the stopping criteria can either be a suitably low temperature or when the system is frozen at the current temperature. Some implementations keep the temperature decreasing until some other condition is met. For example, no change in the best state for a certain period of time.That is, a particular phase of the search normally continues at a certain temperature until some sort of equilibrium is reached. This might be a certain number of iterations or it could be until there has been no change in state for a certain number of iterations.

3.2.5 Operators and other implementation issues in the genetic algorithms

We do not use the “standard” randomly generated initial populations (Goldberg (1989)), while in the algorithms used here, the initial populations consist of chromosomes with just one outlier, different from each other (the size of the population is less than the number of observations). At the beginning, all possible single-outlier chromosomes are generated and sorted in terms of fitness value and the initial population consists of the chromosomes having the largest fitness. In this way we evaluate from the beginning the most promising one-outlier patterns (see Baragona et al. (2001)).

The “roulette wheel” rule is used for parent selection. The probability of a chromosome being selected as a parent is proportional to the rank of its fitness. Each selected couple of parents will produce two “children” by methods of crossover and mutation.

The crossover operator used is “uniform crossover” Goldberg (1989). For each gene of the first child, one of the parents is selected at random (with equal probability of selection) and its corresponding gene is inherited at the same position. The other parent is used to determine the second child’s corresponding gene.

Finally, a probability is chosen for randomly changing the value of each gene of the child-chromosome (mutation). In our encoding, where we have only two admissible values for a gene (“0” and “1”) the application of the mutation operator is pretty straightforward.

The entire population of chromosomes is replaced by the offsprings created by the crossover and mutation processes at each generation except for the best chromosome, which survives to the next generation. This *elitist* strategy ensures that the fitness will never decrease through generations (Rudolph (1994)).

3.3 The TPP procedure

Let $y_t = [y_{1,t}, \ldots, y_{s,t}]'$ be a k-dimensional vector time series following the stationary and invertible vector autoregressive moving average (VARMA) model:

$$\Phi(B)y_t = \Theta(B)\epsilon_t, t = 1, \ldots, N, \tag{3.10}$$

where B is the backshift operator such that $By_t = y_{t-1}$, $\Phi(B) = (I - \Phi_1 B - \Phi_2 B^2 - \ldots \Phi_p B^p)$ and $\Theta(B) = (I - \Theta_1 B - \Theta_2 B^2 - \ldots \Theta_p B^p)$ are $k \times k$ matrix polynomials of finite degrees p and q and $\epsilon_t = (\epsilon_{1t}, \ldots, \epsilon_{kt})$ is a sequence of independent and identically distributed (iid) Gaussian random vectors with mean 0 and positive-definite covariance matrix Σ. For the VARMA model in equation (3.10), we have the AR representation $\Pi(B)y_t = \epsilon_t$ where $\Pi(B) = \Theta(B)^{-1}\Phi(B) = I - \sum_{i=1}^{\infty} \Pi_i B^i$.

Given an observed time series $\mathbf{z} = [z_1, \ldots, z_N]$ where $z_t = [z_{1,t}, \ldots, z_{s,t}]'$ Tsay et al. (2000) generalized additive univariate outliers to the vector case in a direct manner using the representation

$$z_t = y_t + \omega I_t^{(h)} \tag{3.11}$$

where $I_t^{(h)}$ is a dummy variable such that $I_h^{(h)} = 1$ and $I_t^{(h)} = 0$ if $t \neq h$, $\omega = (\omega_1, \omega_2, \ldots, \omega_k)'$ is the size of the outlier, and y_t follows a VARMA model.

Tsay et al. (2000) showed that when the model order is known, the estimate of the size of an additive multivariate outlier at time h is given by:

$$\hat{\omega}_{A,h} = -(\sum_{i=0}^{N-h} \hat{\Pi}_i' \Sigma^{-1} \hat{\Pi}_i)^{-1} \sum_{i=0}^{N-h} \hat{\Pi}_i' \Sigma^{-1} \tag{3.12}$$

The covariance matrix of this estimate is $\Sigma_{A,h}^{-1} = (\sum_{i=0}^{N-h} \hat{\Pi}_i' \Sigma^{-1} \hat{\Pi}_i)^{-1}$. Tsay et al. (2000) proposed an iterative procedure similar to that of the univariate case to detect multivariate outliers. Assuming no outlier, the procedure starts building a multivariate ARMA model for the series under study and let $\hat{a}_t$ be the estimated residuals and $\hat{\Pi}_i$ the estimated coefficients of the autoregressive representation. The second step of the procedure requires the calculation of the test statistic:

$$J_{max} = \max_{1 \le t \le N} \{J_t\},$$

where $J_t = \hat{\omega}_{A,t}' \Sigma_{A,h}^{-1} \hat{\omega}_{A,h}$. As in the univariate case, if J_{max} is significant at time index t_0 we identify a additive multivariate outlier at t_0. Once an outlier is identified, its impact on underlying time series is removed, using the model in equation (3.11). The adjusted series is treated as a new time series and the detecting procedure is iterated. The TPP method terminates when no significant outlier is detected. Tsay et al. (2000) used simulation to generate finite sample critical values of statistic J_{max}.

3.4 Performance of meta-heuristic methods

To test the performance of meta-heuristic algorithms for identifying outliers in multivariate time series we applied the proposed methods to simulated time series models of the class VARIMA. We consider eight vector VARMA models, four bivariate ($s = 2$) and four trivariate models ($s = 3$). The sample sizes used are $N = 200$ and $N = 400$. The models considered in this simulation study and reported in Galeano et al. (2006), Lütkepohl (1993), Tsay et al. (2000) are listed below.

Model 1 - VAR(1) bivariate model: $\Phi_1 = \begin{bmatrix} 0.6 & 0.2 \\ 0.2 & 0.4 \end{bmatrix}$.

Model 2 - VAR(1) bivariate model: $\Phi_1 = \begin{bmatrix} 0.2 & 0.3 \\ -0.6 & 1.1 \end{bmatrix}$.

Model 3 - VAR(2) bivariate model: $\Phi_1 = \begin{bmatrix} 0.5 & 0.1 \\ 0.4 & 0.5 \end{bmatrix}$ $\Phi_2 = \begin{bmatrix} 0.0 & 0.0 \\ 0.25 & 0.0 \end{bmatrix}$.

Model 4 - VARMA(1,1) bivariate model: $\Phi_1 = \begin{bmatrix} 0.6 & 0.2 \\ 0.2 & 0.4 \end{bmatrix}$ $\Theta_1 = \begin{bmatrix} -0.7 & 0.2 \\ -0.1 & 0.4 \end{bmatrix}$.

Model 5 - VAR(1) trivariate model: $\Phi_1 = \begin{bmatrix} 0.6 & 0.2 & 0.0 \\ 0.2 & 0.4 & 0.0 \\ 0.6 & 0.2 & 0.5 \end{bmatrix}$.

Model 6 - VAR(1) trivariate model: $\Phi_1 = \begin{bmatrix} 0.2 & 0.3 & 0.0 \\ -0.6 & 1.1 & 0.0 \\ 0.2 & 0.3 & 0.6 \end{bmatrix}$.

Model 7 - VAR(2) trivariate model:

$$\Phi_1 = \begin{bmatrix} -0.3 & 0.15 & 0.95 \\ 0.0 & -0.15 & 0.3 \\ 0.0 & 0.2 & -0.25 \end{bmatrix} \qquad \Phi_2 = \begin{bmatrix} -0.15 & 0.1 & 0.9 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.35 & 0.0 \end{bmatrix}.$$

Model 8 - VARMA(1,1) trivariate model:

$$\Phi_1 = \begin{bmatrix} 0.6 & 0.2 & 0.0 \\ 0.2 & 0.4 & 0.0 \\ 0.6 & 0.2 & 0.5 \end{bmatrix} \qquad \Theta_1 = \begin{bmatrix} -0.7 & 0.0 & 0.0 \\ -0.1 & -0.3 & 0.0 \\ -0.7 & 0.0 & -0.5 \end{bmatrix}.$$

where the covariance matrix of the Gaussian noise is the identity matrix for seven models. For the Model 2, it has diagonal entries equal to 1.0 and all off-diagonal entries equal to -0.2.

We have considered three different outlier configurations. The first two instances have a small contamination: the first configuration has two isolated outliers at time indices $t = 100, 150$, and the second one has a patch of two outliers introduced at time indices $t = 100, 101$. The last one consists in a heavier contamination, that includes two isolated outliers and a patch of three outliers introduced at time indices $t = 40, 100, 101, 102, 150$. For the first two cases the size of each outlier is chosen equal to $\omega = (3.5, 3.5)'$ for bivariate models and is chosen equal to $\omega = (3.5, 3.5, 3.5)'$ for the trivariate models. When the contamination is heavier we set the size of each outlier equal to $\omega = (5.0, 5.0)'$ for bivariate models and we set $\omega = (5.0, 5.0, 5.0)'$ for the trivariate models. For each model, sample size and outliers configuration, we generate a set of 100 time series.

We may consider several criteria for evaluating the performance procedure. Since the proposed procedures are designed to detect the outliers avoiding false identifications, we used as criteria of evaluation the relative frequency of correct outlier detection, defined as a correct identification of outlier pattern. For the case of two outliers $(100, 150$ or $100, 101)$ this means the relative frequency of detecting both outliers and only them, while for the case of five outliers the relative frequency of detecting all five outliers and only them. For each method, we include the relative frequency of partial correct configuration detection (the relative frequency of only one outlier correctly detected or the relative frequency of less than five outliers correctly detected) and the relative frequency of wrong identifications (i.e., solutions where at least one observation identified as outlier in fact is not).

To apply the algorithms we need to determine the values of two types of parameters, one concerning the outlier problem itself and the other one regarding the meta-heuristic algorithms. The parameters of the outlier detection problem are three: the constant c in (3.7), the order of the multivariate autoregressive process m in (1.37) and the maximum number of outliers g.

3.4.1 The problem of parameters tuning

The constant c

In order to obtain the critical values of the test statistics for outlier detection (in univariate and multivariate time series) one can rely on simulation, using a large number of series from different models (Tsay et al. (2000), Galeano et al. (2006)). Programs TRAMO and SCA, for example, have outlier detection routines that use critical values obtained by such a simulation study. In our work we follow the same idea to establish the value of the constant c through a Monte Carlo experiment.

We consider the eight vector VARMA models listed above and sample sizes $N = 200, 400$. For each model and sample size, we generate a set of 500 time series and apply the algorithms to each set, employing different values of c and recording the corresponding values of the type I error α (where α is the frequency of clean observations identified as outliers).

Table 3.1 provides the c values obtained via simulation according to different values of α, models, dimensions and sample sizes. We observed that the three meta-heuristic algorithms lead to similar simulation results, therefore in Table 3.1 we do not consider the effect of these algorithms on the constant c. Table 3.1 suggests the following observations. First, for each α, we see only minor differences in the c values among different models given dimension and sample size. Second, the estimated c values increase with the sample size N and decrease with the dimension s. In general, the sample size and the time series dimension are important factors affecting the behaviour of constant c, while the type of model does not seem to have a significant effect.

Table 3.1: Simulation study: c values corresponding to different type I error α

N	s	Model	α		
			0.10	0.05	0.01
200	2	1	7.17	7.68	9.53
		2	7.33	7.93	9.25
		3	7.29	7.89	9.20
		4	7.18	7.84	9.50
	3	5	5.71	6.13	7.03
		6	5.78	6.30	7.20
		7	5.72	6.20	7.50
		8	5.67	6.17	7.50
400	2	1	8.10	8.83	10.20
		2	8.05	8.59	10.50
		3	7.93	8.55	9.80
		4	7.57	8.19	9.68
	3	5	6.13	6.70	8.13
		6	6.23	6.78	8.13
		7	6.15	6.67	8.00
		8	5.80	6.33	7.80

In real application, it may be necessary to analyze time series with different sample sizes and different number of components. To address this need, we suggest a theoretical approximation to derive the constant c.

Let us consider a test where under the null hypothesis the time series is outlier free and under the alternative hypothesis a single outlier occurs at unknown time t. We may use as statistic test:

$$\Lambda_{max} = \max_{1\leq t\leq N}\{\Lambda_t\},$$

where $\Lambda_t = (\mathbf{X_t'\hat{\Gamma}i}z)'(\mathbf{X_t'\hat{\Gamma}iX_t})^{-1}(\mathbf{X_t'\hat{\Gamma}i}z)$ and $\mathbf{X_t}$ is the pattern design corresponding to just one outlier at time t. The statistic Λ_t is a quadratic form and is distributed approximately as a chi-squared random variable with s degrees of freedom under the null hypothesis of no outliers. The finite sample distribution of Λ_{max} is complicated because of the correlation between the Λ_t. We may obtain the approximate percentiles of Λ_{max} assuming the independence among the Λ_t (though a relatively strong hypothesis)

$$P(\Lambda_{max} < \lambda_\alpha) = [P(\chi^2_s < \lambda_\alpha)]^N = 1-\alpha$$

or

$$P(\chi^2_s < \lambda_\alpha) = (1-\alpha)^{1/N},$$

where λ_α is the $(1-\alpha)$th quantile of the chi-square distribution with s degrees of freedom. We reject the null hypothesis if Λ_{max} is greater than the quantile λ_α at the α significance level.

Now, a problem arises, when the value of N increases the quantity $(1-\alpha)^{1/N} \to 1$ and $\lambda_\alpha \to \infty$. To solve this problem we approximate the distribution of Λ_{max} with the Gumbel distribution:

$$P\left(\frac{\Lambda_{max}-d_N}{c_N} < \nu_\alpha\right) = \exp(-e^{-\nu_\alpha}) = 1-\alpha,$$

where $d_N = 2(\log N + (\frac{s}{2}-1)\log(\log N) - \log\Gamma(\frac{s}{2}))$ and $c_N = 2$, and we obtain the quantiles for Λ_{max} as $\lambda_\alpha = c_n\nu_\alpha + d_N$.

Now we can choose the constant c so that, whenever the null hypothesis of no outlier is accepted, the fitness of the chromosome with no outlier is larger than the one of the best one-outlier chromosome, or similarly $\Lambda_{max} < cs$, therefore put $c = \lambda_\alpha/s$.

In Table 3.2 we observe that the resulting theoretical c values are always slightly larger than the simulated ones, so that by using them the test is more conservative.

The discrepancy between the theoretical and simulated c values may be caused by the dependence among the Λ_t variables.

The c values used in our simulation experiments are the simulated ones values reported in Table 3.2 corresponding to $\alpha = 0.05$

Table 3.2: Simulated and *theoretical* c values corresponding to different type I error α, dimensions s and sample sizes N

N	s	α		
		0.10	0.05	0.01
200	2	7.2	7.9	9.4
		7.5	*8.3*	*9.9*
	3	5.7	6.2	7.3
		5.9	*6.4*	*7.5*
400	2	7.9	8.5	10.0
		8.2	*8.9*	*10.6*
	3	6.0	6.6	8.0
		6.3	*6.7*	*8.0*

The parameters m and g

To determine the value of order m in (1.37) we used the FPE criterion (Lütkepohl (1993)). Alternatively we could use Akaike's Information Criterion which differs from FPE essentially by a term of order $O(N^{-2})$ and thus the two criteria are almost equivalent for large N (Lütkepohl (1993)).

The value of the parameter g should be chosen by taking into account the length of the time series and all other relevant information. The value g affects the choice of the iteration number. If we increase the value for g it seems reasonable to increase also the iteration number of the meta-heuristic algorithms because a larger solution space has to be explored. The selected value for g is 5 for all algorithms.

3.4.2 Meta-heuristic control parameters tuning

A correct choice of the value of the control parameters is important for the performance of the meta-heuristic algorithms. For the genetic algorithms, choices have to be made for the crossover probability (*pcross*), mutation probability (*pmut*), population size (*pop*) and the number of generations or termination criterion (*gen*) (see section A.4 in appendix).

For the simulated annealing algorithm we have to determine the initial temperature (T_0), final temperature (T_f), number of internal loop iterations at any temperature (SA_{iter}), and the constant a in (3.9), characterising the cooling schedule. As reported in section (A.2) in appendix, the number of evaluations of the objective function I_{tot}^{SA} depends on the choice of these parameters. Generally we establish a number of I_{tot}^{SA} and the parameters are chosen in order to meet this constraint (see section A.2 in appendix).

Threshold accepting requires two parameters: the number of thresholds (N_t) and the number of internal loop iterations at any threshold (TA_{iter}). Also in this case, if we set I_{tot}^{TA}, N_t and TA_{iter} must be chosen in such a way that their product is equal to I_{tot}^{TA} (see section A.3 in appendix).

Unfortunately, the correct choice of the suitable parameter values is a difficult task because a wide range of values needs to be considered for each parameter and some parameters may be correlated with each other. Few theoretical guidelines are available while experience with practical applications of meta-heuristic algorithms is offered by a vast literature.

Regarding the TA, two simple procedures that can be used to generate the threshold sequences are reported in section (A.3) of appendix. First, one might use a linear threshold sequence decreasing to zero and, alternatively, one might use a data driven generation of the threshold sequence (see algorithm (3) in the appendix) suggested by Winker & Fang (1997). In our simulation experiments we set the value of M in algorithm (3) to 2,000. There are several examples in literature suggesting that the two procedures are equivalent, while in some applications the method proposed by Winker & Fang (1997) yields better results. As far as the number of thresholds N_t is concerned, Gilli & Winker (2009) suggested the minimum value for N_t around 10. However, when the total number of iterations I_{tot}^{TA} becomes very large, N_t might be increased.

Some guidelines for the choice of GA parameters may be found in de Jong (1975), Schaffer et al. (1989), da Graça Lobo (2000), Eiben et al. (1999), South et al. (1993). de Jong (1975) studies the effects of some control parameters of GA on its performance, concerning the population size, and the crossover and mutation probabilities. Using five different function optimisation scenarios, De Jong systematically varies these parameters, analyses the results and thus establishes guidelines for robust parameter choice. De Jong suggests population size $pop = 50$, probability of crossover $pcross = 0.6$, probability of mutation $pmut = 0.001$ and the adoption of the elitist strategy. However, other empirical studies (Eiben et al. (1999), South et al. (1993), da Graça Lobo (2000), Gao (2003), Grefenstette (1986)) indicate different values for these parameters.

Regarding the SA algorithm, the initial temperature must be set to a high value enough to allow a move to almost any neighbourhood state. However, if the temperature starts at too high a value then the search can move to any neighbour and thus transform the search (at least in the early stages) into a random search. Then, a very high initial temperature may influence the quality of the performance and the length of the computational time. If we know the maximum distance (objective function difference) between one neighbour and another then we can use this information to calculate a starting temperature. Another method, suggested in (Rayward-Smith, 1996), is to start with a very high temperature and cool it rapidly until about 60% of worst solutions are being accepted. This forms the real starting temperature and it can now be cooled more slowly. A similar idea, suggested in (Dowsland, 1995), is to rapidly heat the system until a certain proportion of worse solutions are accepted and then slow cooling can start. This can be seen to be similar to how physical annealing works in that the material is heated until it is liquid and then cooling begins (i.e. once the material is a liquid it is pointless carrying on heating it).

Theoretically, the cooling rate parameter a in (3.9) assumes values between 0 and 1, while Eglese (1990) reports that values used in practice lie between 0.8 and 0.99. Park & Kim (1998) suggest a systematic procedure, based on the simplex method for non linear programming, to determine parameter values.

In conclusion we can say that there is no uniformly best choice of parameters, but specific problems may require different values. Baragona et al. (2011) suggest that a good choice may be obtained by considering a range of possible values for the same problems. In our applications these parameters values are chosen by a tuning experiment. For each algorithm, different combinations of parameters values are tried, keeping the number of the objective function evaluations constant. We select the parameter combination that yields the largest frequency of true outlier pattern detection.

A simulation experiment for tuning parameters

The remaining parameter values are chosen by means of a tuning experiment where a set of 200 time series with $N = 400$ have been generated by Model 2, and outliers at time indices 100 and 150 are analysed. All the algorithms run with a total of 2,000 evaluations of the objective function.

For the SA, the T_f is always kept equal to 0.05. Since T_f has the role of stopping criterion, a value close to zero seems reasonable, thus the probability of accepting a worse solution during the last iterations is very small. The examined values for a are [0.90, 0.94, 0.95, 0.96] and for T_0 are [2, 4, 6, 8, 10]. For each combination, the number of internal loop iterations SA_{iter} is equal to the ratio between the total number of evaluations of the objective function (2000) and the number of different temperatures (the number depending on T_0 and a). Table 3.3 shows the frequencies of correct identifications (based on 200 time series) for each pair of a and T_0. When decreasing the value of a, the best performance is obtained by increasing the value of T_0. The pair $a = 0.95$ and $T_0 = 8$ is used.

Table 3.3: SA tuning experiment: frequencies of correct identifications for different values of T_0 and a.

a	T_0				
	2	4	6	8	10
0.90	0.825	0.845	0.850	0.830	0.870
0.94	0.820	0.850	0.860	0.880	0.880
0.95	0.835	0.880	0.840	0.900	0.855
0.96	0.820	0.835	0.875	0.870	0.845

For the GA algorithms, we compare the frequency of the correct outlier pattern identification for 8 different combinations of population size *pop* and number of generations *gen*, keeping the mutation probability *pmut* and the crossover probability *pcross* constant for all experiments. The values considered for the population size are [10, 20, 30, 40, 50, 70, 100, 200], for the number of generations are [10, 20, 30, 40, 50, 70, 100, 200], while $pcross = 0.001$ and $pmut = 0.6$ (these values were suggested by de Jong (1975)).

Table 3.4 suggests for the parameter *pop* an average value (between 70 and 100). In a second stage, different combinations of *pmut* and *pcross* are considered from $pmut = \{0.1, 0.01, 0.001, 0.0005\}$ and $pcross = \{0.4, 0.6, 0.8, 0.9\}$ whereas the population size and the number of generations are kept constant at 100 and 20, respectively. The results of some combinations of *pmut* and *pcross* are reported in table 3.4. The results indicate that better results are obtained for average values of crossover probability *pcross* and very low values, but not too much, of mutation probability *pmut*. Based on these results, we use as values: $pmut = 0.001$ and $pcross = 0.6$.

For TA algorithm, we compared a linear sequence of thresholds and a sequence generated by the method given in Winker & Fang (1997). The linear sequences were generated considering different initial thresholds and different rates of decrease. The initial thresholds {6, 8, 10, 14} are used while the values {0.90, 0.96} are considered as rates of decrease. For the method proposed by (Winker & Fang (1997)) , we considered 8 combinations of the number of thresholds N_t and number of iterations SA_{iter} choices from N_t ={10, 20, 30, 40 , 50, 70, 100, 200} and SA_{iter} = {10, 20, 30, 40, 50, 70, 100, 200 }. With regard to the linear sequence, the results suggest to use a high threshold and a rate of decrease of the thresholds not very rapid. For the method proposed by (Winker & Fang (1997)) the best result is obtained in correspondence to number of thresholds N_t equal to 100. However, there is not a constant improvement as the number of thresholds is incremented and also the differences are not very marked. Observing the thresholds provided by Winker & Fang (1997) method, we observed that the initial threshold is large enough (slightly more than 14) and the thresholds decrease very slowly. This particular result depends on the type of problem considered. The value of the objective function for the solutions that belong to a neighborhood can be very different because the removal or insertion of a given anomaly can lead to great changes in the value of the AIC. This means that the distribution $F(\Delta)$ (see algorithm (3) in the appendix) does not appear to be symmetrical around zero, but is asymmetric towards higher values. From these results it was decided to use a sequence of thresholds $N_t = 100$ obtained by the method of Winker.

Table 3.4: TA and GA tuning experiment: frequencies of correct identifications for different combinations of parameters.

TA		GA			
(N_t, TA_{iter})	f_{TA}	(pop, gen)	f_{GA}	$(pmut, pcross)$	f_{GA}
(10, 200)	0.860	(10,200)	0.815	(0.01,0.4)	0.850
(20,100)	0.865	(20,100)	0.830	(0.01,0.6)	0.875
(30,70)	0.860	(30,70)	0.850	(0.01,0.8)	0.835
(40,50)	0.880	(40,50)	0.850	(0.01,0.9)	0.825
(50,40)	0.875	(50,40)	0.840	(0.001,0.4)	0.880
(70,30)	0.885	(70,30)	0.885	(0.001,0.8)	0.880
(100,20)	0.885	(100,20)	0.885	(0.001,0.9)	0.850
(200,10)	0.855	(200,10)	0.880	(0.0005,0.6)	0.830

We summarize the parameter values used in the simulations. We imposed that the objective (fitness) function were evaluated not more than 10,000 times: I_{tot}^{TA}= I_{tot}^{SA}=I_{tot}^{GA}=10,000. For the algorithm SA we chose $T_0 = 8.0$, $T_f = 0.05$, $SA_{iter} = 100$, $a = 0.95$. For the algorithm TA, we set $N_t = 100$ and $TA_{iter} = 100$. For the genetic algorithm we selected $pcross = 0.6$, $pmut = 0.001$, $pop = 100$, $gen = 100$. With $g = 5$, the solution space Ω is of order 2×10^9 when the sample size is $N = 200$, and it is of order 8×10^{10} when the sample size is $N = 400$ whereas the meta-heuristic algorithms reach a satisfying convergence to the optimum evaluating the objective function (fitness) no more than $10,000$ times.

3.5 Results

In Tables 3.5, 3.6 and 3.7 we report the results of the three meta-heuristic algorithms and the TPP detection procedure. In Tables 3.5 and 3.6, the rows labelled P_2 summarise the relative frequency of the correct outlier pattern (both outliers detected and only them), the rows labelled P_1 summarise the relative frequency of only one outlier correctly detected and the rows labelled E summarise the relative frequency of the solutions with wrong identifications (i.e., observations that are identified as outliers while they are not). The complement to one of the sum of these three frequencies is the frequency of the no outlier solution. In Table 3.7, the rows labelled P_5 summarise the relative frequency of the correct outlier pattern (all five outliers detected and only them), the rows labelled $P_{<5}$ summarise the relative frequency of less than five outliers correctly detected and the rows labelled E summarise the relative frequency of solutions with wrong identifications (i.e., observations that are identified as outliers while they are not). The complement to one of the sum of these three frequencies is again the frequency of the no outlier solution.

Table 3.5 shows that each of the four algorithms has a high percentage of success when the two outliers are far from each other ($t = 100, 150$). The frequencies of full identifications are nearly equivalent for the four methods. The results are mixed and no method seems uniformly superior to the others. For some models the frequency of correct identification of the TPP method is larger than the corresponding meta-heuristic frequency, while for other models the converse is true.

Table 3.6 reports simulation results concerning the outliers patch detection where outliers are introduced at time indices $t = 100, 101$. We can see from this table that for almost all models the meta-heuristic algorithms detect the outlier patch with frequencies higher than those achieved by the TPP. Only for the model (7) the TPP method provides satisfactory results. Moreover, for almost all the models the TPP's frequency of wrong identification E is considerable larger than the corresponding frequencies achieved by meta-heuristic methods. In comparison to the preceding case (two outliers for each other) here the frequency of the no outlier solution is larger, and the largest for the TPP method. Finally, we can see that the frequencies P_2 for models with 200 observations are less than same models with 400 observations. This may be due to the fact that the solution space is larger and the meta-heuristic methods are were easily trapped in some local optimum.

In Table 3.7 are reported the results for the configuration with 5 outliers where three are consecutive. The configuration is very complex and very difficult to detect if the size of the outlier is not large enough. For this reason outlier sizes are set to 5.0 for the instants $40, 100, 101, 102, 150$. In the table 3.7 we can see that the relative frequencies of correct configuration P_5 obtained through the meta-heuristic methods are very different and depending on the model. For some models the relative frequency of correct outlier detection are very low.

To reduce the lack of convergence, we reported the simulations allowing for a total number of objective function (fitness) evaluations increased to 100,000 (instead of 10,000), both for the most complex configuration $(40, 100, 101, 102, 150)$ and for thesimpler one $(100, 101)$.

Table 3.8-3.9 shows the results obtained for the configurations $100, 101$ and $40, 100, 101, 102, 150$ setting the number of evaluations equal to 100,000. We can see an improvement of the results in both cases but the increase of the frequencies of correct identification is very large for the case of 5 outliers. Now the relative frequencies of correct configuration detection obtained through the meta-heuristic methods are high and much larger than those obtained with the TPP method for seven of the eight models considered. For some models the correct pattern is always found (frequency P_5 assumes the value 1). The meta-heuristic algorithms show a better performance than the TPP also in the third configuration outliers (see Table 3.9).

Tables 3.8 and 3.9 evidently illustrate masking and smearing problems encountered by the TPP procedure when additive outliers exist in a patch. It has been noticed that this problems persist despite the size of outliers whereas the meta-heuristic methods improve their performance when the outliers are inserted with a bigger magnitude. Detecting a set of consecutive outliers seems much more difficult and affected by the underlying models. The good performance of TPP in model 7 depends on the particular parameters of the model generating data. The three algorithms proposed here clearly outperform the TPP method to detect patch of additive outliers.

To understand the poor TPP's results, let us to consider the situation in which the time series follows a VAR(1) and there exists a patch of two additive outliers at time indices $t = T, T+1$, with magnitudes $\omega_t = \omega$ for $t = T, T+1$. Suppose that the model parameters are known, then the expected values of the perturbations at time indices $t = T, T+1$ are given by

$$E(\hat{\omega}_T) = \omega_T + \mathbf{\Gamma i}_0{}^{-1}\mathbf{\Gamma i}_1\omega_{T+1} = (I_s + \mathbf{\Gamma i}_0{}^{-1}\mathbf{\Gamma i}_1)\omega,$$
$$E(\hat{\omega}_{T+1}) = \omega_{T+1} + \mathbf{\Gamma i}_0{}^{-1}\mathbf{\Gamma i}_{-1}\omega_T = (I_s + \mathbf{\Gamma i}_0{}^{-1}\mathbf{\Gamma i}_{-1})\omega.$$

We observe that they are biased. The bias depends on the inverse covariance matrices and it may cause the masking effect. The good performance achieved by the TPP in model 7 may depend on the peculiar parameters of the models. On the contrary in our methods the estimates of the magnitude of outliers are unbiased.

3.5.1 Real time series data

In this subsection we illustrate the performance of the meta-heuristic procedures by analysing a real example. The data are the well-known gas-furnace series of Box et al. (1994). This bivariate time series consists of an input gas rate in cubic feet per minute and the CO_2 concentration in the outlet gas as a percentage, both measured at 9–second time intervals. There are 296 observations. The TPP method finds additive multivariate outliers at positions 42, 54, 113, 199, 235, 264. All the other algorithms, based on 1,000,000 objective function (fitness) evaluations (T_0= 8.0, T_f= 0.05, SA_{iter} = 10,000, a = 0.95, gen=30,000, pop=30, N_t=100 and TA_{iter} = 10,000, g = 15, c = 8.2 and m = 6) converge to the solution with 4 outliers at positions: 42, 54, 199 and 264. Additional information may be derived by looking also at the sub-optimal solutions. Table 3.10 displays the outliers patterns corresponding to the best ten solutions found after 1,000,000 objective function evaluations. It suggests that additional time indices may be considered as candidates for the true outlier positions, giving additional insight about the probably outlying observations. It turns out that for this series the TPP method has not given the best solution, but the ten-th one in order of decreasing objective function.

Let I denote the number of evaluations of the objective function. In order to compare the convergence of the algorithms we calculate, for different values of I (100, 500, 1,000, 5,000, 10,000), the empirical distribution, based on 100 restarts, of the best obtained objective function. Table 3.11 reports some relevant statistics (mean, standard deviation, best value and 5-th percentile) about the empirical dis-

Table 3.5: Comparison of the algorithm performances: outliers at $t = 100, 150$ based on 10^4 iteration

	$N = 200$				$N = 400$			
	TA	SA	GA	TPP	TA	SA	GA	TPP
Model 1								
P_2	0.90	0.91	0.91	0.94	0.87	0.87	0.92	0.89
P_1	0.05	0.04	0.04	0.02	0.10	0.10	0.05	0.06
E	0.05	0.05	0.05	0.04	0.03	0.03	0.03	0.04
Model 2								
P_2	0.91	0.90	0.91	0.92	0.92	0.92	0.94	0.93
P_1	0.03	0.04	0.03	0.03	0.04	0.04	0.02	0.02
E	0.06	0.06	0.06	0.05	0.04	0.04	0.04	0.05
Model 3								
P_2	0.94	0.94	0.94	0.94	0.91	0.91	0.93	0.93
P_1	0.01	0.01	0.01	0.00	0.02	0.02	0.00	0.00
E	0.04	0.04	0.04	0.06	0.06	0.06	0.06	0.07
Model 4								
P_2	0.94	0.94	0.94	0.90	0.91	0.91	0.91	0.91
P_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E	0.06	0.06	0.06	0.10	0.09	0.09	0.09	0.09
Model 5								
P_2	0.90	0.90	0.90	0.93	0.94	0.94	0.94	0.94
P_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E	0.10	0.10	0.10	0.07	0.06	0.06	0.06	0.06
Model 6								
P_2	0.90	0.90	0.90	0.92	0.90	0.90	0.90	0.94
P_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E	0.10	0.10	0.10	0.08	0.10	0.10	0.10	0.06
Model 7								
P_2	0.95	0.94	0.95	0.94	0.90	0.90	0.90	0.93
P_1	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
E	0.05	0.05	0.05	0.06	0.01	0.10	0.10	0.07
Model 8								
P_2	0.94	0.94	0.94	0.92	0.96	0.96	0.96	0.96
P_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E	0.06	0.06	0.06	0.08	0.04	0.04	0.04	0.04

P_2= frequency of event 'exactly two outliers found at times 100 and 150'

P_1= frequency of event 'exactly one outlier found at time 100 or at time 150'

E= frequency of solutions with wrong identifications

Table 3.6: Comparison of the algorithm performances: outliers at $t = 100, 101$ based on 10^4 iteration

	$N = 200$				$N = 400$			
	TA	SA	GA	TPP	TA	SA	GA	TPP
Model 1								
P_2	0.72	0.71	0.72	0.23	0.55	0.56	0.58	0.19
P_1	0.05	0.06	0.05	0.08	0.07	0.06	0.05	0.07
E	0.11	0.11	0.11	0.18	0.13	0.13	0.12	0.14
Model 2								
P_2	0.74	0.74	0.75	0.22	0.68	0.67	0.69	0.21
P_1	0.10	0.10	0.10	0.37	0.15	0.14	0.12	0.40
E	0.13	0.13	0.12	0.25	0.10	0.10	0.10	0.25
Model 3								
P_2	0.83	0.83	0.84	0.34	0.74	0.75	0.78	0.43
P_1	0.03	0.03	0.03	0.06	0.05	0.05	0.04	0.05
E	0.07	0.07	0.06	0.23	0.12	0.11	0.09	0.21
Model 4								
P_2	0.52	0.52	0.54	0.00	0.40	0.41	0.42	0.01
P_1	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01
E	0.21	0.21	0.19	0.30	0.29	0.28	0.27	0.41
Model 5								
P_2	0.89	0.89	0.89	0.55	0.83	0.82	0.83	0.55
P_1	0.00	0.00	0.00	0.08	0.01	0.02	0.01	0.11
E	0.11	0.11	0.11	0.23	0.15	0.15	0.15	0.23
Model 6								
P_2	0.84	0.84	0.84	0.55	0.81	0.81	0.82	0.52
P_1	0.01	0.01	0.01	0.00	0.01	0.01	0.00	0.01
E	0.13	0.13	0.13	0.32	0.17	0.17	0.17	0.35
Model 7								
P_2	0.92	0.92	0.92	0.90	0.88	0.88	0.88	0.87
P_1	0.01	0.01	0.02	0.02	0.00	0.00	0.00	0.04
E	0.07	0.07	0.07	0.08	0.12	0.12	0.12	0.09
Model 8								
P_2	0.91	0.91	0.91	0.10	0.89	0.89	0.91	0.03
P_1	0.00	0.00	0.00	0.15	0.00	0.00	0.00	0.08
E	0.09	0.09	0.09	0.70	0.11	0.11	0.09	0.88

P_2= frequency of event 'exactly two outliers found at times 100 and 150'
P_1= frequency of event 'exactly one outlier found at time 100 or at time 150'
E= frequency of solutions with wrong identifications

Table 3.7: Comparison of the algorithm performances: outliers at $t = 40, 100, 101, 102, 150$ based on 10^4 iteration

	$N = 200$				$N = 400$			
	TA	SA	GA	TPP	TA	SA	GA	TPP
Model 1								
P_2	0.60	0.58	0.63	0.32	0.32	0.32	0.37	0.24
P_1	0.28	0.30	0.25	0.39	0.48	0.48	0.44	0.46
E	0.12	0.12	0.12	0.29	0.20	0.20	0.19	0.30
Model 2								
P_2	0.75	0.00	0.00	0.29	0.68	0.00	0.00	0.27
P_1	0.13	0.00	0.00	0.45	0.20	0.00	0.00	0.50
E	0.12	0.00	0.00	0.26	0.12	0.00	0.00	0.23
Model 3								
P_2	0.72	0.75	0.76	0.28	0.47	0.47	0.49	0.35
P_1	0.15	0.13	0.12	0.29	0.24	0.24	0.23	0.25
E	0.13	0.12	0.12	0.43	0.29	0.29	0.28	0.40
Model 4								
P_2	0.23	0.22	0.26	0.01	0.20	0.21	0.23	0.00
P_1	0.31	0.32	0.31	0.22	0.21	0.20	0.20	0.19
E	0.46	0.46	0.43	0.77	0.59	0.59	0.57	0.81
Model 5								
P_2	0.84	0.84	0.85	0.55	0.72	0.71	0.72	0.54
P_1	0.03	0.03	0.02	0.13	0.08	0.09	0.08	0.15
E	0.13	0.13	0.13	0.32	0.20	0.20	0.20	0.31
Model 6								
P_2	0.95	0.95	0.95	0.41	0.90	0.90	0.90	0.40
P_1	0.02	0.02	0.02	0.04	0.02	0.02	0.02	0.03
E	0.03	0.03	0.03	0.55	0.08	0.08	0.08	0.57
Model 7								
P_2	1.00	1.00	1.00	0.91	1.00	1.00	1.00	0.90
P_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
E	0.00	0.00	0.00	0.09	0.00	0.00	0.00	0.09
Model 8								
P_2	0.57	0.58	0.60	0.00	0.66	0.65	0.68	0.01
P_1	0.11	0.10	0.08	0.35	0.03	0.04	0.03	0.28
E	0.32	0.32	0.32	0.65	0.31	0.31	0.29	0.71

P_5= frequency of event 'exactly five outliers found at times 40, 100, 101, 102, 150'

$P_{<5}$= frequency of event 'some of correct outliers are detected'

E= frequency of solutions with wrong identifications

Table 3.8: Comparison of the algorithm performances: outliers at $t = 100, 101$ based on 10^5 iteration

	$N = 200$				$N = 400$			
	TA	SA	GA	TPP	TA	SA	GA	TPP
Model 1								
P_2	0.73	0.73	0.73	0.23	0.61	0.61	0.61	0.19
P_1	0.05	0.05	0.05	0.08	0.05	0.05	0.05	0.07
E	0.10	0.10	0.10	0.18	0.09	0.09	0.09	0.14
Model 2								
P_2	0.75	0.75	0.75	0.22	0.72	0.72	0.72	0.21
P_1	0.10	0.10	0.10	0.37	0.11	0.11	0.11	0.40
E	0.12	0.12	0.12	0.25	0.10	0.10	0.10	0.25
Model 3								
P_2	0.84	0.84	0.84	0.34	0.83	0.83	0.83	0.43
P_1	0.03	0.03	0.03	0.06	0.03	0.03	0.03	0.05
E	0.06	0.06	0.06	0.23	0.05	0.05	0.05	0.21
Model 4								
P_2	0.60	0.60	0.60	0.00	0.64	0.64	0.64	0.01
P_1	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01
E	0.13	0.13	0.13	0.30	0.05	0.05	0.05	0.41
Model 5								
P_2	0.90	0.90	0.90	0.55	0.93	0.93	0.93	0.55
P_1	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.11
E	0.10	0.10	0.10	0.23	0.06	0.06	0.06	0.23
Model 6								
P_2	0.85	0.85	0.85	0.55	0.88	0.88	0.88	0.52
P_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
E	0.13	0.13	0.13	0.32	0.10	0.10	0.10	0.35
Model 7								
P_2	0.92	0.92	0.92	0.90	0.88	0.88	0.88	0.87
P_1	0.01	0.01	0.02	0.02	0.00	0.00	0.00	0.04
E	0.07	0.07	0.07	0.08	0.12	0.12	0.12	0.09
Model 8								
P_2	0.93	0.93	0.93	0.10	0.96	0.96	0.96	0.03
P_1	0.00	0.00	0.00	0.15	0.00	0.00	0.00	0.08
E	0.07	0.07	0.07	0.70	0.04	0.04	0.04	0.88

P_2= frequency of event 'exactly two outliers found at times 100 and 150'

P_1= frequency of event 'exactly one outlier found at time 100 or at time 150'

E= frequency of solutions with wrong identifications

Table 3.9: Comparison of the algorithm performances: outliers at $t = 40, 100, 101, 102, 150$ based on 10^5 iteration

	$N = 200$				$N = 400$			
	TA	SA	GA	TPP	TA	SA	GA	TPP
Model 1								
P_2	0.89	0.90	0.95	0.32	0.80	0.80	0.92	0.24
P_1	0.06	0.05	0.00	0.39	0.09	0.09	0.00	0.46
E	0.05	0.05	0.05	0.29	0.11	0.11	0.08	0.30
Model 2								
P_2	0.86	0.86	0.87	0.29	0.84	0.85	0.87	0.27
P_1	0.10	0.10	0.09	0.45	0.12	0.11	0.09	0.50
E	0.09	0.04	0.04	0.26	0.04	0.04	0.04	0.23
Model 3								
P_2	0.95	0.97	0.99	0.28	0.86	0.90	0.94	0.35
P_1	0.02	0.00	0.00	0.29	0.04	0.02	0.00	0.25
E	0.03	0.03	0.01	0.43	0.10	0.08	0.06	0.40
Model 4								
P_2	0.74	0.73	0.75	0.01	0.82	0.82	0.84	0.00
P_1	0.14	0.15	0.13	0.22	0.05	0.05	0.05	0.19
E	0.12	0.12	0.12	0.77	0.13	0.13	0.11	0.81
Model 5								
P_2	0.97	0.97	1.00	0.55	0.96	0.96	1.00	0.54
P_1	0.00	0.00	0.00	0.13	0.00	0.00	0.00	0.15
E	0.03	0.03	0.00	0.32	0.04	0.04	0.00	0.31
Model 6								
P_2	1.00	1.00	1.00	0.41	0.98	0.98	1.00	0.40
P_1	0.00	0.00	0.00	0.04	0.02	0.02	0.00	0.03
E	0.00	0.00	0.00	0.55	0.00	0.00	0.00	0.57
Model 7								
P_2	1.00	1.00	1.00	0.91	1.00	1.00	1.00	0.90
P_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
E	0.00	0.00	0.00	0.09	0.00	0.00	0.00	0.09
Model 8								
P_2	0.93	0.92	0.95	0.00	0.93	0.93	0.94	0.01
P_1	0.02	0.03	0.00	0.35	0.02	0.02	0.01	0.28
E	0.05	0.05	0.05	0.65	0.05	0.05	0.05	0.71

P_5= frequency of event 'exactly five outliers found at times 40, 100, 101, 102, 150'

$P_{<5}$= frequency of event 'some of correct outliers are detected'

E= frequency of solutions with wrong identifications

Table 3.10: Meta-heuristic algorithm solutions for the gas–furnace series

Solution	$f(x)$	Locations
S_1	-53.82	42 54 199 264
S_2	-53.29	43 54 199 264
S_3	-51.42	42 54 199 235 264
S_4	-50.89	43 54 199 235 264
S_5	-50.10	42 54 113 199 264
S_6	-49.57	43 54 113 199 264
S_7	-48.55	42 55 199 264
S_8	-48.02	43 55 199 264
S_9	-47.78	42 54 198 264
S_{10}	-47.70	42 54 113 199 235 264

Table 3.11: Statistics of empirical distributions for different values of I (based on 100 runs)

I	TA				SA			
	$\hat{\mu}$	$\hat{\sigma}$	$best$	$q_{0.05}$	$\hat{\mu}$	$\hat{\sigma}$	$best$	$q_{0.05}$
100	-19.50	14.77	-44.59	-39.42	-14.81	13.98	-40.32	-36.24
500	-42.54	6.43	-53.82	-53.82	-33.21	7.65	-45.54	-44.58
1,000	-48.68	4.71	-53.82	-53.82	-39.10	6.60	-53.82	-48.69
5,000	-52.83	1.87	-53.82	-53.82	-52.79	1.92	-53.82	-53.82
10,000	-53.16	1.17	-53.82	-53.82	-53.16	1.15	-53.82	-53.82
I	GA$_1$				GA$_2$			
	$\hat{\mu}$	$\hat{\sigma}$	$best$	$q_{0.05}$	$\hat{\mu}$	$\hat{\sigma}$	$best$	$q_{0.05}$
100	-31.69	6.91	-44.92	-44.02				
500	-44.59	6.86	-53.82	-53.82				
1,000	-49.19	4.53	-53.82	-53.82				
5,000	-51.71	2.92	-53.82	-53.82				
10,000	-53.01	1.17	-53.82	-53.82				

tributions along the guidelines suggested by Gilli & Winker (2009). As I increases, the distributions shift to the left ($\hat{\mu}$ decreases) and become less dispersed ($\hat{\sigma}$ decreases). The GA show a better initial performance due to the favourable way the initial population is chosen, but the SA and the TA have a faster convergence speed.

At the last iteration ($I = 10,000$), the best value ($f(x) = -53.82$) is found in 59 out of 100 runs for the SA, in 58 out of 100 runs for the TA, in 46 out of 100 for the GA.

Chapter 4

Conclusions and Further Developments

In chapter 2, a GAs-based procedure for identifying and estimating a MSETAR model with univariate or bivariate threshold variable is suggested. The procedure uses a special binary encoding composed of several fragments each of which represent a integer parameter of the MSETAR model. In spite of the relative complexity of the chromosome the genetic operators are suitable for simple implementation so that the computational burden is quite low. A simulation experiment demonstrated the validity of the GAs for implementing the identification and estimation procedure for building a nonlinear model in a multivariate setting. An application to real world data concerned with exchange rates of the United States dollar with four other countries currency between January 1980 and March 1984 proved the effectiveness of our procedure in empirical applications.

There are at least two issues that will possibly be interesting subject matters for future research. The first one is concerned with the consideration of subset VAR models in each regime. This may save considerable estimation effort, produces more stable coefficient estimates and would lead to the identification of a smaller size parameter set. On the other hand, the identification of subset models is known to constitute a difficult problem for which GAs have been suggested in the context of VAR models and univariate threshold models. The additional computational burden is a non negligible obstacle that requires both an appropriate encoding and a careful programming to be overcome. Next, consideration of more than two component series to be used as threshold variables for regime identification is an intricate matter that surely deserves further research. As before, it involves not only theoretical difficulties but the development of dedicated programming tools as well.

In chapter 3, three meta-heuristic methods for detecting additive outliers in multivariate time series are proposed. Meta-heuristic algorithms, unlike other methods in literature, do not identify and remove outliers one at a time, but examine several proposed outlier patterns, where all observations are simultaneously considered. This feature seems to be effective in handling masking (meaning that one outlier hides others) and swamping (when outliers make other clean observations to appear outliers as well) effects caused by multiple outliers. Furthermore, our methods do not require the specification of an adequate multivariate model, which is usually a difficult task, especially when the data are contaminated by outliers. The procedures are illustrated by analysing artificial and real data sets. The results obtained from the simulation experiments seem to support the idea that the meta-heuristic algorithms constitute a valid approach to detect the time points where potential outliers in vector time series are located. In our experiment the meta-heuristic methods provide better results than the TPP method to identify outlier patch, while the results are similar for the case of well separated outliers. The examination of the "gas-furnace" data of Box and Jenkins yields satisfactory results. Comparing the results obtained by the detection procedure of Tsay et al. (2000) with the best solution provided by meta-heuristic algorithms, we observe that they have in common four out of six outliers locations. Such small discrepancy is caused by the difference between the two identification procedures. The efficiency of the meta-heuristic methods proposed in this study, depends crucially on the choice of appropriate values for some control parameters. The simulation and the theoretical study used for determining the value of parameter c, allows us to control for the type I error α. For any given value of α there is a corresponding value for c that does not depend on the underlying model. It only depends on the number of components (s) and the length of the time series. In the case of real data, given a value of α, the corresponding value of c, as reported in Table 3.2, can be used.

The presence of partial outliers, i.e., anomalies that affect only some components of the multivariate series, may be an issue to be considered for future developments. Moreover, an interesting further problem is the outlier identificability, that is, studying how large should the outliers size to ensure that the correct outlier configuration has the maximum fitness.

Appendix A

Meta-heuristic methods

A.1 Introduction

Many optimisation problems do not satisfy the necessary conditions to guarantee the convergence of traditional numerical methods. For instance, in order to apply standard gradient methods to maximum likelihood estimation we need a globally convex likelihood function, however there are a number of relevant cases with non convex likelihood functions or functions with several local optima. Another class of *hard* problems is when the solution space is discrete and large. These problems are known as combinatorial problems. There is an objective function to be minimized, as usual; but the space over which that function is defined is not simply the n-dimensional space of n continuously variable parameters. Rather, it is a discrete, but very large, configuration space, like the set of possible orders of cities, or the set of possible allocations of silicon real estate blocks to circuit elements. We can consider a general statement of combinatorial optimization problem as:

$$\text{Minimize } f(x_1, x_2, \ldots, x_n) : \Omega \to \mathbb{R} \tag{A.1}$$

where the variables $x_1, x_2, \ldots, x_n$ take discrete values and $f(\cdot)$ represents the objective function, which has to be minimized over a discrete n-dimensional search space Ω (the collection of all feasible solutions). Of course, by replacing $f(\cdot)$ with $-f(\cdot)$, the algorithm can also be applied to maximization problems.

A simple approach for solving an instance of a combinatorial problem is to list all the feasible solutions, evaluate their objective function, and pick the best one. However, for a combinatorial problem of a reasonable size, the complete enumeration of its elements is not feasible, and most available searching algorithms are likely to yield some local optimum as a result ((Rayward-Smith et al. 1996)). Meta-heuristic algorithms are often used to solve this kind of problems. Heuristics typically start with a feasible solution and use an iterative procedure to search for improved solutions. For the minimization problem (equation A.1) with feasible search space Ω, an heuristic searches for a practical solution close to the optimal solution x^* where, for any $x \in \Omega$, $f(x^*) < f(x)$. These algorithm are call meta-heuristics because consist of general search principles organized in a general search strategy. The success of meta-heuristic methods is due to several factors: they do not rely on a set of strong assumptions about the optimisation problem, they are robust to changes in the characteristics of the problem, they do not produce a deterministic solution but a high quality stochastic approximation to the global optimum.

In this thesis we are interested in the following meta-heuristic methods: simulated annealing, threshold accepting and genetic algorithms.

SA and TA are classified as *local search methods*. Classical local search algorithms are a class of methods in which the iterative procedure starts with a feasible solution ξ^c, and then at each iteration attempts to find a better solution by searching in a neighbourhood of the current solution ξ^c. This neighbourhood is a set of feasible solutions where the values of the variables are close to those of the current solution. Each time a new solution in the neighbourhood is an improvement, it is used to update the current solution. The iterative procedure ends based on pre-specified stopping criteria, such as when no further improvement is found or when the total number of iterations reaches a given limit. However, these algorithms may get stuck in local optima. To avoid this problem, the local search algorithms we adopt in this research may accept worse solutions than the current one.

Genetic algorithms were initially developed by Holland (1975) and are classified as *population based* methods, or *evolutionary algorithms*. They work on a whole set of solutions that is adapted simultaneously by imitating the evolutionary process of species that fit to the environment and reproduce.

We give a brief sketch of the three methods.

A.2 Simulated annealing

Simulated annealing (SA) is a random search technique based on an analogy to the physical process of annealing that occurs in thermodynamics, when a heated material cools down and changes its structure under a controlled temperature lowering schedule. At high temperatures, the molecules of a liquid move freely with respect to one another. If the liquid is cooled slowly, thermal mobility is lost. The atoms are often able to line themselves up and form a pure crystal that is completely ordered over a distance up to billions of times the size of an individual atom in all directions. This crystal is the state of minimum energy for this system. The amazing fact is that, for slowly cooled systems, nature is able to find this minimum energy state. In fact, if a liquid metal is cooled quickly or quenched, it does not reach this state but rather ends up in a polycrystalline or amorphous state having somewhat higher energy. So the essence of the process is slow cooling, allowing ample time for redistribution of the atoms as they lose mobility. This is the technical definition of annealing, and it is essential for ensuring that a low energy state will be achieved.

Metropolis et al. (1953) introduced a simple algorithm, known as Metropolis algorithm, to simulate the annealing process. In each step of this algorithm, an atom is given a small random displacement and the resulting change, ΔE, in the energy of the system is computed. If $\Delta E \leq 0$, the displacement is accepted, and the configuration with the displaced atom is used as the starting point of the next step. The case $\Delta E > 0$ is treated probabilistically: the probability that the configuration is accepted is $P(\Delta E) = exp(-\Delta E/kT)$. This choice of $P(\Delta E)$ has the consequence that the system evolves into a Boltzmann distribution.
Thirty years later, Kirkpatrick et al. (1983) proposed a method, based on Metropolis algorithm, for finding the global minimum of a objective function that may possess several local minimal. This method, called simulated annealing, used the objective function in place of the energy, configurations are feasible solutions of the problem and the change of configuration corresponds to neighbouring solutions.

In analogy with the Metropolis algorithm, simulated annealing is characterised by the presence of a control parameter T called *temperature*, an annealing schedule which tells how it is lowered from high to low values, an acceptance probability and a stopping rule. Temperature T is a non-increasing function of time; it is designed to exclude almost all *bad moves* at the end. In a classical schedule starting from T_0, the temperature is maintained constant for SA_{iter} consecutive steps. Then, after each series of SA_{iter} steps, it is decreased through multiplication by a fixed factor α $(0 < \alpha < 1)$. This implies the setting of three parameters, T_0, α and SA_{iter}, which will be respectively referred to as initial temperature, cooling rate and length of plateau. Different cooling schedules are suggested in the literature. On the analogy of thermodynamics, a Boltzmann-like distribution is usually chosen as acceptance probability. The stopping criteria can either be a suitably low temperature or when the system is *frozen* at the current temperature (i.e. no better or worse moves are being accepted).

SA algorithm is an iterative procedure that extends the local search method, described above, to allow for a new solution at some iterations to be worse than the current solution, rather than an improvement. This extension helps to avoid getting trapped in a local optimum. By accepting worse solutions in some neighborhoods, the heuristic searches more widely within the feasible search space, so that it is more likely to escape a local optimum and move to the global optimum.

In terms of the minimization problem given by equation (A.1), the algorithm for a simulated annealing heuristic consists of the steps reported in algorithm (1).

Algorithm 1 Pseudocode for simulated annealing.

```
Initialise $T_0$, $T_f$, $a$ and $SA_{iter}$
Generate initial solution $\xi^c$
$T = T_0$
while $T > T_f$ do
    for $r = 1$ to $SA_{iter}$ do
        Compute $\xi^n \in N(\xi^c)$ (neighbour to current solution)
        Compute $\Delta = f(\xi^n) - f(\xi^c)$ and generate $u$ from a uniform random variable
        between 0 and 1
        if $\Delta < 0$ or $e^{-\Delta/T} > u$ then
            $\xi^c = \xi^n$
        end if
    end for
    $T \leftarrow aT$
end while
```

Like the local search method, the simulated annealing heuristic searches for a new solution ξ^n at each iteration in the neighborhood of the current solution ξ^c. If the new solution is an improvement($f(\xi^n) < f(\xi^c)$), it is accepted as the update to the current solution, just as in the local search method. In addition, if the new solution is worse to the current solution ($f(\xi^n) > f(\xi^c)$), the new solution is sometimes accepted, with a given probability that depends on the difference between the values of objective function for the new and current solutions. The bigger this difference, the smaller the probability that the new (worse) solution is accepted as the update to the current solution. The acceptance probability is determined by whether a random number u generated between 0 and 1 is less than or greater than the function $e^{-\Delta/T}$, where Δ is the difference between f(ξ^n) and f(ξ^c), and T is a temperature parameter. The temperature is initially set at a high value, in order to accept worse solutions frequently. In this way, in the initial stage of research, the algorithm is able to overcome the local optima, and the space of the solutions may be explored more uniformly. It is then gradually lowered as the iterative procedure progresses to allow fewer and fewer worse solutions, that is, the algorithm becomes more and more selective in accepting new solutions. By the end, only moves that improve $f(\xi)$ are accepted in practice. The algorithm then coincides, for low temperatures, with a local search algorithm.

The total number of iterationI_{tot}^{SA} is obtained as the number of different temperatures $N_{temperature}$ (function of T_0, T_f, a) times the number of steps SA_{iter}.

Recent applications of the simulated annealing algorithm are discussed by Vera & Díaz-García (2008), Depril et al. (2008), Duczmal & Assunção (2004) and Angelis et al. (2001).

A.3 Threshold accepting

Threshold accepting (TA) was introduced by Dueck & Scheuer (1990) as a deterministic analog to simulated annealing. They applied the algorithm to a Travelling Salesman Problem and argued that their algorithm is superior to classical simulated annealing. It is a refined local search procedure which escapes local optima by accepting solutions which are worse,but no more than a given threshold. The algorithm is deterministic as it uses a deterministic acceptance criterion instead of the probabilistic one used in simulated annealing for accepting worse solutions. The number of steps where we explore the neighborhood for improving the solution is fixed. The threshold is decreased iteratively and reaches the value of zero after a given number of steps. The TA algorithm has an easy parameterization, it is robust to changes in problem characteristics and works well for many problem instances. . Threshold accepting has been successfully applied to different areas of statistics and econometrics (Winker & Fang (1997), Fang et al. (2000), Winker (2000), Winker (2001), Gilli & Winker (2004), Maringer & Winker (2009), Lin et al. (2010), Lyra et al. (2010), Winker et al. (2011)). An extensive introduction to TA is given in Winker (2001).

Algorithm (2) provides the pseudo-code for a prototype threshold accepting implementation for a minimization problem.

Comparing SA and TA algorithm we can see that, first, the sequence of temperatures T is replaced by a sequence of N_t thresholds τ_h with $h = 1, \dots, N_t$ and, the most important, the statement 8 of algorithm (1) is replaced by:

Algorithm 2 Pseudocode for Threshold Accepting.

```
Initialise $N_t$, $TA_iter$,
Generate the sequence $\tau_h$, $h = 1, \ldots, N_t$
Generate initial solution $\xi^c$
for $h = 1$ to $N_t$ do
    for $r = 1$ to $TA_{iter}$ do
        Compute $\xi^n \in N(\xi^c)$ (neighbour to current solution)
        Compute $\Delta = f(\xi^n) - f(\xi^c)$ and generate $u$ from a uniform random variable between 0 and 1
        if $\Delta < 0$ or $\Delta < \tau_h$ then
            $\xi^c = \xi^n$
        end if
    end for
end for
```

$$\textbf{if} \quad \Delta < \tau_h \quad \textbf{then} \quad \xi^c = \xi^n.$$

In this case the total number of iteration I_{tot}^{TA} is obtained as the product of the number of different thresholds N_t and the number of times each thresholds is used, TA_{iter}.

A crucial element of TA is its threshold sequence since it determines TA's ability to overcome local optima. Basically, the idea is to accept ξ^n if its objective function value is better or if it is not much worse than that of ξ^c where not much worse means the deterioration may not exceed some threshold τ defined by the threshold sequence. In extreme cases of threshold settings, the algorithm behaves like a classical local search algorithm (if all threshold values are set equal to zero) or like a random walk (if all values of the threshold sequence are set to a very large value). Althöfer & Koschnick (1991) demonstrated the convergence of the TA algorithm under the hypothesis that an appropriate threshold sequence exists. But in their proof they do not provide a way to construct an appropriate sequence. Consequently, the threshold sequence is often chosen in a rather ad hoc approach. Two simple procedures can be used to generate the sequence of thresholds. In the first place, one could use a linear sequence decreasing to zero. The advantage of a linear threshold sequence consists in the fact, that for tuning purposes only the first value of the sequence has to be selected as it fixes the whole sequence. Alternatively, we can generate a sequence of selected thresholds using the a data driven method suggested in Winker & Fang (1997). This procedure is detailed in algorithm (3).

Algorithm 3 Pseudocode for generating threshold sequence.

```
Initialise $N_t$ and $M$
for $r = 1$ to $M$ do
    Randomly choose solution $\xi_r^c$
    Randomly choose neighbour solution $\xi_r^n \in N(\xi_r^c)$
    Compute $\Delta_r =\mid f(\xi_r^c) - f(\xi_r^n) \mid$
end for
Compute the cumulative distribution function F of $\Delta_r$, $r = 1, \ldots, M$
Compute the sequence of thresholds $\tau_i = F^{-1}(\frac{N_t-1}{N_t}), i = 1, \ldots, N_t$
```

This method uses a two step process to construct the threshold sequence. For the first step a large number (M) of possible solutions ξ^c is generated at random. Then, we compute the distances between the values of the objective function at random point ξ_r^c and its neighbour ξ_r^n, $\Delta_r =\mid f(\xi_r^c) - f(\xi_r^n) \mid, r = 1, 2, \ldots, M$. In the second step the cumulative empirical distribution F of the distances Δ_r is computed. This distribution is an approximation of the distribution of local relative changes of the objective function. The thresholds τ_i are computed as the quantiles Q_i corresponding to percentiles $P_i = \frac{N_t-i}{N_t}, i = 1, \ldots, N_t$. The threshold sequence will be monotonically decreasing to zero.

A.4 Genetic algorithms

Genetic algorithms (GAs) are global stochastic optimization techniques that are based on the adaptive mechanics of natural selection evolution. They were introduced in Holland (1975), and subsequently made widely popular by Goldberg (1989). The statistical applications of the GAs have been discussed by Chatterjee et al. (1996) and Chatterjee & Laudato (1997). GAs use two basic processes from evolution: inheritance, or the passing of features from one generation to the next, and competition, or survival of the fittest. Through these processes individuals which are most successful in surviving will have relatively larger numbers of offspring. Poorly performing individuals will produce few of even no offspring at all. This means that the genes from the highly adapted, or fit individuals will spread to an increasing number of individuals in each successive generation. The combination of good characteristics from different parents can sometimes produce highly fit offsprings, whose fitness is greater than that of either parent. In this way, species evolve to become more and more well suited to their environment.

The general structure of genetic algorithms is shown in algorithm (4).

Algorithm 4 Pseudocode for genetic algorithms.

```
Set population size ($pop$), probability of crossover ($pcross$), probability of mutation ($pmut$), number of generations ($gen$)
Generate initial population $P$ of solutions
for $i = 1$ to $gen$ do
    Evaluate each individual's fitness
    Initialise $P' = \emptyset$ (set of children)
    for j =1 to $\frac{pop}{2}$ do
        Select individuals $x_a$ and $x_b$ from $P$ with probability proportional to their fitness
        Generate $p_1$ and $p_2$ from a uniform random variable $U(0,1)$
        if $p_1 > pcross$ then
            Apply crossover to $x_a$ and $x_b$ to produce $x_a^{child}$ and $x_b^{child}$
        else
            $x_a^{child} = x_a$ and $x_b^{child} = x_b$
        end if
        if $p_2 > pmut$ then
            Apply mutation to $x_a^{child}$ and $x_b^{child}$
        end if
        $P' = P' \cup \{x_a^{child}, x_b^{child}\}$
    end for
    $P = P'$
end for
```

A genetic algorithm maintains a population of solution candidates and works as an iteration loop. First, an initial population is generated randomly. Each individual in the population is an encoded form of a solution to the problem under consideration, called a chromosome which is usually a string of characters or symbols, e.g., a string of 0's and 1's (a binary string). The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated by a fitness evaluation function, $g(\cdot)$, and selected according to the fitness values using a selection mechanism, e.g., fitness-proportionate selection, so that fitter chromosomes have higher probabilities of being selected. New chromosomes, called offspring, are formed by either merging two selected chromosomes from the current generation using a crossover operator, or modifying a chromosome using a mutation operator. Crossover results in the exchange of genetic material between relatively fit members of the population, potentially leading to a better pool of solutions. Mutation randomly introduces new features into the population to ensure a more thorough exploration of the search space. A whole new population of possible solutions is thus produced by selecting the best individuals from the current generation, and mating them to produce a new set of individuals. This new generation contains a higher proportion of the characteristics possessed by the good members of the previous generation. In this way, over many generations, good characteristics are spread throughout the population, being mixed and exchanged with other good characteristics as they go. By favouring the mating of the more fit individuals the population's average fitness will improve and most promising areas of the search space are explored. If the GA has been designed well, the population will converge to a best chromosome approaching the optimal or near-optimal solution.

To use genetic algorithms, each of the following must be developed:

Encoding scheme. In GAs, a population of candidate solutions is maintained and manipulated by genetic operators. The solutions are encoded as chromosomes (usually strings of characters or symbols, e.g., binary strings, real number strings, or symbol strings) to which genetic operators can be applied. An encoding scheme is needed to map candidate solutions into coded strings.

Initialization of population. The initialization is usually done randomly to sample the search space uniformly without bias. A well-initialized population can improve the algorithm's robustness and effectiveness in finding an optimal solution, while a poorly-initialized population may trap the algorithm in local optima and make it hard to reach the global optimum.

Evaluation function. During the operation of genetic algorithms, all chromosomes are evaluated to see how fit they are as solutions to the problem. An evaluation function is required to assign a fitness value to each chromosome.

Selection. The key principle of Darwinian natural evolution theory is that fitter individuals have a greater chance to reproduce offspring, and it is by this principle of *survival of the fittest* that species evolve into better forms. In genetic algorithms, the bias towards fitter individuals is achieved through selection. The objective of any selection scheme is to statistically guarantee that fitter individuals have a higher probability of selection for reproduction. In a GA, selection is carried out in two different stages: parent selection and generational selection. Parent selection is the step in which individuals from the parent generation are selected as parents to create offspring. Generational selection is carried out after a specified number of offspring are generated. In general, the new generation is created by selecting individuals from both the parent generation and the offspring generation. Most selection schemes belong to the following two categories: stochastic selection and deterministic selection. For parent selection, stochastic selections are usually applied, and for generational selection, deterministic selections are usually used. Fitness proportionate selection (roulette wheel and stochastic universal) and tournament selection are two of the most popular stochastic selection algorithms. Proportionate selection methods assign probability to an individual according to its fitness, and this can be problematic. Indeed, if the fitness range is too large, then only a few good individuals will be selected. This will tend to fill the entire population with similar chromosomes and will limit the ability of the GA to explore the search space. On the other hand, if the fitness values are too close to each other, then the GA will tend to select one copy of each individual, with only random variations in selection. Consequently, it will not be guided by small fitness variations and will be reduced to random search. Fitness scaling and Rank-based selection are two alternative methods that have been proposed to compensate for these issues. Using fitness scaling, the fitness of all parents can be scaled relative to some reference value, and proportionate selection then assigns selection probability according to the scaled fitness values. Several scaling mechanisms have been proposed. In general, the scaled fitness $g_k^{'}$ derived from the raw fitness g_k for chromosome k can be expressed as $g_k^{'} = G(g_k)$: where the mapping function $G(\cdot)$ transforms the raw fitness into scaled fitness. The function $G(\cdot)$ may take different forms to yield different scaling methods, such as linear scaling, sigma truncation, power law scaling, etc. For example, the 'sigma truncation scaling' (e.g., Goldberg 1989) consists in applying the normalization transform

$$\mathrm{g_k}' = \mathrm{g_k} - (\bar{g} - c\sigma)\,,$$

where $\bar{g}$ is the population mean, c is a suitable real positive constant and σ

the standard deviation, and in excluding the individuals with zero or negative fitness from selection. For detailed description of scaling methods, (see Gen & Cheng (1997)).

Rank-based selection methods utilize the indices of individuals when ordered according to fitness to calculate the corresponding selection probabilities, rather than using absolute fitness values (Baker 1987)).

Deterministic selection schemes are usually used in generational selection to select individuals from both the parent generation and offspring generation to create the next generation. Most GA implementation are based on the generational replacement where the entire parent generation is replaced by their offspring (i.e., the offspring generation is taken as the new generation, and the parent generation is discarded after the offspring generation is created).

Crossover. Once two chromosomes are selected, the crossover exchanges parts of their genes and generates two new strings that share characteristics of both original chromosomes. Crossover is the most important genetic operator for a GA, and it is the driving force for exploration of the search space. The performance of the GA depends to a great extent on the performance of the crossover operator used (Holland 1975). Crossover operator is not typically applied for all parents but it is applied with probability *pcross* which is normally set equal to a value in [0.6,1]. During the last decades, a number of different crossover operators have been successfully designed: single-point crossover, two-point crossover, uniform crossover, non-geometric crossover etc. A comparison of different binary crossover operators was undertaken in Eshelman et al. (1989), both theoretically and empirically. It was found that none of them is the consistent winner, and there was not more than 20% difference in speed among the techniques.

Mutation. After new individuals are generated through crossover, mutation is applied with a low probability, *pmut*, to introduce random changes into the population. In a binary-coded GA, mutation means that, with a given probability *pmut*, each bit (gene) of each string (chromosome) may change its value from 0 to 1 or vice versa, while in a nonbinary-coded GA, mutation involves randomly generating a new value in a specified position in the chromosome. In GAs, mutation serves the crucial roles of replacing gene values lost from the population during the selection process so that they can be tried in a new context, and of providing gene values that were not present in the initial population. By introducing random changes into the population, more regions of the search space can be evaluated, and premature convergence can be avoided. A variety of mutation operators have been proposed in the literature: Flip Bit, uniform, non-uniform, Gausssian etc.

Bibliography

Akaike, H. (1974), 'A new look at the statistical identification model', *IEEE Transactions on Automatic Control* **19**, 716–723.

Althöfer, I. & Koschnick, K. (1991), 'On the convergence of threshold accepting', *Applied Mathematics and Optimization* **24**, 183–195.

Angelis, L., Bora-Senta, E. & Moyssiadis, C. (2001), 'Optimal exact experimental designs with correlated errors through a simulated annealing algorithm', *Computational Statistics and Data Analysis* **37**, 275–296.

Arnold, M. & Gunther, R. (2001), 'Adaptive parameter estimation in multivariate self-exciting threshold autoregressive models', *Communications in Statistics - Simulation and Computation* **30**, 257–275.

Aytug, H. & Koehler, G. J. (2000), 'New stopping criterion for genetic algorithms', *European Journal of Operational Research* **126**, 662–674.

Back, T., Fogel, D. B. & Michalewicz, Z. (1997), *Handbook of Evolutionary Computation*, Taylor and Francis, New York London.

Bai, J. & Perron, P. (2003), 'Computation and analysis of multiple structural change models', *Journal of Applied Econometrics* **18**, 1–22.

Baker, J. E. (1987), Reducing bias and inefficiency in the selection algorithm, *in* 'Proceedings of the Second International Conference on Genetic Algorithms', Lawrence Erlbaum Associates, Inc. Mahwah, NJ, USA, pp. 14–21.

Baragona, R. & Battaglia, F. (1989), Identificazione e stima di dati anomali in serie temporali per mezzo di interpolatori lineari, Technical Report 19, Department of Statistical Sciences, University of Roma La Sapienza, Italy.

Baragona, R. & Battaglia, F. (2007), 'Outliers detection in multivariate time series by independent component analysis', *Neural Computation* **19**, 1962–1984.

Baragona, R., Battaglia, F. & Calzini, C. (2001), ‘Genetic algorithms for the identification of additive and innovational outliers in time series’, *Computational Statistics and Data Analysis* **37**, 1–12.

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

Baragona, R., Battaglia, F. & Poli, I. (2011), *Evolutionary statistical procedures*, Springer-Verlag, Berlin.

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

Barbieri, M. (1991), ‘Outliers in serie temporali multivariate’, *Quaderni di Statistica e Econometria* **13**, 1–11.

Barnett, G., Kohn, R. & Sheather, S. (1997), ‘Robust bayesian estimation of autoregressive-moving average models’, *Journal of Time Series Analysis* **18**, 11–28.

Battaglia, F. (1983), ‘Inverse autocovariances and a measure of linear determinism for a stationary process’, *Journal of Time Series Analysis* **4**, 79–87.

Battaglia, F. (1984), ‘Inverse covariances of a multivariate time series’, *Metron* **42**, 117–129.

Battaglia, F. (2007), *Metodi di Previsione Statistica*, Springer-Verlag Italia, Milano.

Battaglia, F. & Baragona, R. (1992), ‘Linear interpolators and the outlier problem in time series’, *Metron* **50**, 79–97.

Battaglia, F. & Protopapas, M. K. (2011), ‘Time-varying multi-regime models fitting by genetic algorithms’, *Journal of Time Series Analysis* **32**, 237–252.

Battaglia, F. & Protopapas, M. K. (2012), ‘Multi-regime models for nonlinear nonstationary time series’, *Computational Statistics* **27**, 319–341.

Bhansali, R. J. (1980), ‘Autoregressive and window estimates of the inverse correlation function’, *Biometrika* **67**, 551–566.

Bhansali, R. J. & Downham, D. Y. (1977), ‘Some properties of the order of an autoregressive model selected by a generalization of Akaike’s FPE criterion’, *Biometrika* **64**, 547–551.

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.

Box, G. & Jenkins, G. (1970), *Time series analysis: Forecasting and control*, Holden-Day, San Francisco.

Brillinger, D. (1975), *Time Series: Data Analysis and Theory*, Holt, New York.

Bruce, A. G. & Martin, D. (1989), 'Leave-k-out diagnostics for time series (with discussion)', *Journal of the Royal Statistical Society: Series B* **51**, 363–424.

Chang, I. & Tiao, G. C. (1983), Estimation of time series parameters in the presence of outliers, Technical Report 8, University of Chicago, Statistics Research Center.

Chang, I., Tiao, G. C. & Chen, C. (1988), 'Estimation of time series parameters in the presence of outliers', *Technometrics* **30**, 193–204.

Chappell, D., Padmore, J., Mistry, P. & Ellis, C. (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 - Simulation and Computation* **26**, 1617–1630.

Chatterjee, S., Laudato, M. & Lynch, L. (1996), 'Genetic algorithms and their statistical applications: an introduction', *Computational Statistics and Data Analysis* **22**, 633–651.

Chen, C. & Liu, L. (1993), 'Joint estimation of model parameters and outlier effects in time series', *Journal of the American Statistical Association* **88**, 284–297.

da Graça Lobo, F. (2000), *The parameter-less genetic algorithm: rational and automated parameter selection for simplified genetic algorithm operation*, PhD thesis, Universidade Nova de Lisboa.

de Jong, K. A. (1975), An Analysis of the Behavior of a Class of Genetic Adaptive Systems, Phd thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI.

Depril, D., Van Mechelen, I. & Mirkin, B. (2008), 'Algorithms for additive clustering of rectangular data tables', *Computational Statistics and Data Analysis* **52**, 4923–4938.

Dickey, D. A. & W.A., F. (1979), 'Distribution of the estimators for autoregressive time series with a unit root', *Journal of the American Statistical Association* **74**, 421– 431.

Dickey, D. A. & W.A., F. (1981), 'Likelihood ratio statistics for autoregressive time series with a unit root', *Econometrica* **49**, 1057–1072.

Duczmal, L. & Assunção, R. (2004), 'A simulated annealing strategy for the detection of arbitrarily shaped spatial clusters', *Computational Statistics and Data Analysis* **45**, 269–286.

Dueck, G. & Scheuer, T. (1990), 'Threshold accepting: A general purpose algorithm appearing superior to simulated annealing', *Journal of Computational Physics* **90**, 161–175.

Eglese, R. (1990), 'Simulated annealing: a tool for operational research', *European Journal of Operational Research* **46**, 271–281.

Eiben, A., Hinterding, R. & Michalewicz, Z. (1999), 'Parameter control in evolutionary algorithms', *IEEE Trans. on Evolutionary Computation* **3**, 124–141.

Engle, R. (1982), 'Autoregressive conditional heteroscedasticity with estimates of the variance of of united kingdom inflation', *Econometrica* **50**, 987–1008.

Eshelman, L., Caruana, R. & Schaffer, D. (1989), Biases in the crossover landscape, *in* 'Proceedings of the Third International Conference on Genetic Algorithms', Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, pp. 10–19.

Fan, J. & Yao, Q. (2003), *Nonlinear Time Series. Nonparametric and Parametric Methods*, Springer-Verlag, Berlin.

Fang, K., Lin, D. K. J., Winker, P. & Zhang, Y. (2000), 'Uniform design:theory and application', *Technometrics* **42**, 237–248.

Fox, A. J. (1972), 'Outliers in time series', *Journal of the Royal Statistical Society:Series B* **34**, 350–63.

Galeano, P., Pena, D. & Tsay, R. S. (2006), 'Outlier detection in multivariate time series by projection pursuit', *Journal of the American Statistical Association* **101**, 654–669.

Gao, Y. (2003), Population size and sampling complexity in genetic algorithms, *in* 'GECCO 2003 Workshop on Learning, Adaptation, and Approximation in Evolutionary Computation', Springer, Berlin, pp. 178–181.

Gen, M. & Cheng, R. (1997), *Genetic Algorithms and Engineering Design*, Wiley, New York Chichester.

Ghos, D. & Dutt, S. (2008), 'Nonstationarity and nonlinearity in the us unemployment rate: A re-examination', *Journal for Economic Educators* **8**, 43–53.

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

Gilli, M. & Winker, P. (2009), Heuristic optimization methods in econometrics, *in* E. Kontoghiorghes & D. Belsley, eds, 'Handbook on computational econometrics', Wiley, Chichester, chapter 3, pp. 81–119.

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

Gómez, V., Maravall, A. & Peña, D. (1993), Computing missing values in time series, Working Paper 93-27 Statistics and Econometric Series 21, Departamento de Estadistica y Econometria Universidad Carlos III de Madrid.

Granger, C. & Andersen, A. (1978), *An Introduction to Bilinear Time Series Models*, Vanderhueck and Ruprecht, Gottingen.

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

Grefenstette, J. (1986), 'Optimization of control parameters for genetic algorithms', *IEEE Systems, Man, and Cybernetics Society* **16**, 122–128.

Hamilton, J. D. (1994), *Time Series Analysis*, Princeton University Press, Princeton.

Hannan, E. (1970), *Multiple Time Series*, John Wiley, New York.

Harvill, J. L. & Ray, B. K. (1999), 'A note on tests for nonlinearity in a vector time series', *Biometrika* **86**, 728–734.

Haupt, R. L. & Haupt, S. E. (2004), *Practical genetic algorithms (2nd edition)*, Wiley, New York Chichester.

Holland, J. (1975), *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and AI.*, The University of Michigan, Ann Arbor, MI.

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.

Jones, D. (1978), ‘Nonlinear autoregressive processes’, *Proc. R. Soc. London* **A360**, 71–95.

Justel, A., Peña, T. & Tsay, R. (2001), ‘Detection of outlier patches in autoregressive time series.’, *Statistica Sinica* **11**, 651–673.

Khattree, R. & Naik, D. (1987), ‘Detection of outliers in bivariate time series data’, *Communications in Statistics - Theory and Methods* **16(12)**, 3701–3714.

Kirkpatrick, S., Gelat, C. D. & Vecchi, M. P. (1983), ‘Optimization by simulated annealing’, *Science* **220**, 671–680.

Kolmogorov, A. (1941), ‘Interpolation and extrapolation’, *Bulletin de l’Academie des Sciences de U.S.S.R., Series Mathematics* **5**, 3–14.

Kwiatkowski, D., Phillips, P. C. B., Schmidt, P. & Shin, Y. (1992), ‘Testing the null hypothesis of stationarity against the alternative of a unit root’, *Journal of Econometrics* **54**, 159–178.

Li, T. & Yorke, J. (1975), ‘Period three implies chaos’, *The American Mathematical Monthly* **82**, 985–992.

Lin, D., Sharpe, C. & Winker, P. (2010), ‘Optimized u-type designs on flexible regions’, *Computational Statistics and Data Analysis* **54**, 1505–1515.

Ljung, G. M. (1989), ‘A note on the estimation of missing values in time series’, *Communications in Statistics - Simulation and Computation* **18(2)**, 459–465.

Ljung, G. M. (1993), ‘On outlier detection in time series’, *Journal of the Royal Statistical Society:Series B* **55**, 559–567.

Lorenz, N. (1963), ‘Deterministic nonperiodic flow’, *J. Atmos. Sci.* **20**, 130–141.

Lütkepohl, H. (1993), *Introduction to multiple time series analysis (2nd edition)*, Spinger-Verlag, Berlin.

Lyra, M., Paha, J., Paterlini, S. & Winker, P. (2010), ‘Optimization heuristics for determining internal rating grading scales’, *Computational Statistics and Data Analysis* **54**, 2693–2706.

Maringer, D. & Winker, P. (2009), 'The convergence of estimators based on heuristics: theory and application to a garch model', *Computational Statistics* **24**, 533–550.

Masani, P. (1960), 'prediction theory of multivariate stochastic processes, iii', *Acta Math.* **104**, 141–162.

May, R. (1976), 'Simple mathematical models with very complicated dynamics', *Nature* **261**, 459–467.

McCulloch, R. E. & Tsay, R. S. (1994), 'Bayesian analysis of autoregressive time series via the Gibbs sampler', *Journal of Time Series Analysis* **15**, 235–250.

Medeiros, M., Veiga, A. & Resende, M. (2002), 'A combinatorial approach to piecewise linear time series analysis', *Journal of Computational and Graphical Statistics* **11**, 236–258.

Metropolis, N., Rosenbluth, A. W., Teller, A. H. & Teller, E. (1953), 'Equation of state calculation by fast computing machines', *Journal of Chemical Physics* **21**, 1087–1091.

Michalewicz, Z. (1996), *Genetic Algorithms + Data Structures = Evolution Programs (3rd edition)*, Springer, Berlin Heidelberg.

Mitchell, M. (1996), *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, Massachusetts.

Moran, J. (1953*a*), 'The statistical analysis of the canadian lynx cycle. i: Structure and prediction', *Australian Journal of Zoology* **1**, 163–173.

Moran, J. (1953*b*), 'The statistical analysis of the canadian lynx cycle, ii. synchronization and meterology', *Australian Journal of Zoology* **1**, 291–298.

Ozaki, T. (1982), 'The statistical analysis of perturbed limit cycles using nonlinear time series models', *Journal of Time Series Analysis* **3**, 29–41.

Park, M. & Kim, Y. (1998), 'A systematic procedure for setting parameters in simulated annealing algorithm', *Computers and Operations Research* **25**, 207–217.

Peña, D. (1990), 'Influential observations in time series', *Journal of Business and Economic Statistics* **8**, 235–241.

Phillips, P. & Perron, P. (1988), 'Testing for a unit root in time series regression', *Biometrika* **75**, 335–346.

Priestley, M. B. (1981), *Spectral Analysis and Time Series*, Academic Press, New York.

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

Quenouille, M. (1957), *The Analysis of Multiple Time Series*, Griffin, London.

Rayward-Smith, V., Osman, I., Reeves, C. & Smith, G. (1996), *Modern Heuristic Search Methods*, Wiley, Chichester, New York.

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

Reinsel, G. (1993), *Elements of multivariate time series*, Springer-Verlag, New York.

Rozanov, Y. (1957), *Stationary random processes*, Holden-Day, San Francisco.

Rudolph, G. (1994), 'Convergence analysis of canonical genetic algorithms', *IEEE Transactions on Neural Networks* **5**, 96–101.

Rudolph, G. (1997), *Convergence properties of evolutionary algorithms*, Verlag Dr. Kovač, Hamburg.

Sayyareha, A., Obeidia, R. & Bar-Henbc, A. (2011), 'Empirical comparison between some model selection criteria', *Communications in Statistics - Simulation and Computation* **40**, 72–86.

Schaffer, J., Caruana, R., Eshelman, L. & Das, R. (1989), A study of control parameters affecting online performance of genetic algorithms for function optimization, *in* 'Proceedings of the 3rd International Conference on Genetic Algorithm', Morgan Kaufmann, San Mateo, CA, pp. 51–60.

Shaman, P. (1976), 'Approximations for stationary covariance matrices and their inverses with applications to arima models', *Annals of Statististics* **4**, 292–301.

South, M., Wetherill, G. & Tham, M. (1993), 'Hitch-hiker's guide to genetic algorithms', *Journal of Applied Statistics* **20**, 153–175.

Subba, R. T. (1981), 'On the theory of bilinear time series models', *Journal of the Royal Statistical Society: Series B* **43**, 224–255.

Tiao, G. C. & Tsay, R. S. (1994), 'Non linear and adaptive modelling in time series', *Journal of Forecasting* **13**, 338–344.

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

Tong, H. & Lim, K. (1980), 'Threshold autoregression, limit cycles and cyclical data (with discussion)', *Journal of the Royal Statistical Society: Series B* **42**, 245–292.

Tong, H., Thanoon, B. & Gudmundsson, G. (1985), 'Threshold time series modelling of two icelandic riverflow systems', *Water Res. Bull.* **21**, 651–661.

Tsay, R., Peña, D. & Pankratz, A. (2000), 'Outliers in multivariate time series', *Biometrika* **87**, 789–804.

Tsay, R. S. (1986), 'Time series model specification in the presence of outliers', *Journal of the American Statistical Association* **81**, 132–141.

Tsay, R. S. (1988), 'Outliers, level shifts, and variance changes in time series', *Journal of Forecasting* **7**, 1–20.

Tsay, R. S. (1998), 'Testing and modeling multivariate threshold models', *Journal of the American Statistical Association* **93**, 231–240.

Vera, J. & Díaz-García, J. (2008), 'A global simulated annealing heuristic for the three-parameter lognormal maximum likelihood estimation', *Computational Statistics and Data Analysis* **52**, 5055–5065.

Vitale, C. (1984), 'Definizione ed uso delle matrici di autocross covarianze inverse', *Statistica* **3**, 395–405.

Whittle, P. (1954), 'The statistical analysis of a seiche record', *Sears Foun. J. Mar. Res.* **13**, 76–100.

Whittle, P. (1963), *Prediction and regulation by linear least-square methods*, English Universities Press, London.

Winker, P. (2000), 'Optimized multivariate lag structure selection', *Computational Economics* **16**, 87–103.

Winker, P. (2001), *Optimization Heuristics in Econometrics and Statistics: A simple approach for complex problems with Threshold Accepting*, Wiley, New York.

Winker, P. & Fang, K. (1997), 'Application of threshold accepting to the evaluation of the discrepancy of a set of points', *SIAM Journal on Numerical Analysis* **34**, 2028–2042.

Winker, P., Lyra, M. & Sharpe, C. (2011), ‘Least median of squares estimation by optimization heuristics with an application to the capm and a multi-factor model’, *Computational Management Science* **8**, 103–123.

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.

Yule, G. (1927), ‘On a method of investigating periodicities in disturbed series, with special reference to wolfer’s sunspot numbers’, *Philosophical Transactions of the Royal Society* **226**, 267–298.