

Université Ministry of Higher Education and Scientific Research

Djillali Liabes University

Faculty of exact sciences



THESIS

**For the fulfillment of the requirements for the
degree of Doctorate of Sciences In applied
mathematics**

Option : Probability and statistics

Presented by

Ramdane RAHAL

**Time series modeling and Forecasting using
genetic algorithms**

Advisor: Dr Chikr El Mezouar ZOUAOUI

co-Advisor: GheribAllah ABDELKADER

Thesis defence :

Committee in charge:

M. Last nameNAME	University	président
M. Last nameNAME	University	Thesis Director
M. Last nameNAME	University	Thesis co-Director
M. Last nameNAME	University	examiner
M. Last nameNAME	University	examiner
M. Last nameNAME	University	examiner
M. Last nameNAME	University	examiner

*This work is dedicated to my mother, my family and to the
many others, though unnamed, who helped me in the
completion of this task.*

Acknowledgements

First and foremost I want to thank my advisor Dr. Chikr Elmezouar Zouaoui. It has been an honor to be her Ph.D. student. On a personal level, I am thankful for his compassionate supervision and the confidence he placed in me. I appreciate all her contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating. The joy and enthusiasm he has for her research were contagious and motivational for me, even during tough times in the Ph.D. pursuit. I am also thankful for the excellent example he has provided as a successful man mathematical and professor.

I am intensely indebted to my associate supervisor, Professor Ghrib Allah Abdelkader. I am thankful for his compassionate supervision and his encouragement. I am deeply indebted to all teachers and administrators in department of probability and statistics, of Djilali Liabes university especially professor Mohammed Attouch and professor Samir Benaissa . I would also Lastly, I would like to thank my family for all their love and encouragement. For my patient wife whose faithful support during the final stages of this Ph.D. And most of all for my loving, supportive, encouraging, Thank you.

Ramdane Rahal
Djilali liabes university
March 2017.

Abstract

Abstract

The time series forecast is a very complex problem, consisting in predicting the behavior of a data series with only the information of the previous sequence. In this thesis, a highly comparative framework for time-series modeling and forecasting is developed to give a good models for time series to perform the prediction as well as linear predictors. For this, we are reduce errors of time series models, by used genetic algorithms GAs, one of robustness methods of optimization. GAs is inspired by natural evolution theories, apply operations of reproduction, crossover and mutation to candidate solutions according to their relative fitness scores in the successive populations of candidates Holland (1975). We choose the mean square errors MSE and Akaike criteria information AIC the objectives functions for optimization to calculate there minimums for select the optimal model. This method is based on the evolution of set of rules genetically codified, two types of coding GAs are used, the binary coded GAs (BCGA) and reel coded GAs (RCGA). In the first time we optimize MSE and AIC with BCGA, and in the second time with RCGA. By order to determine the best method, we compar between different methodologies and we contribute in this comparaive study by use a modification to errors of model given by GAs by normalized them, for have a better model and perform the forecast. The computer simulation results obtained demonstrate that GAs have the potential to become a powerful tool for time series modeling and forecasting and as well as when we use the advanced GAs. To illustrate our studies, we supports all chapters by some examples application on real time series data.

Keywords

Time series; Box-Jenkins method, Binary coded genetic algorithms(BCGA), Real coded genetic algorithms(RCGA).

Résumé

La prédiction des séries chronologiques est un problème très complexe, comment prédire le comportement d'une série de données avec uniquement les informations précédentes de la séquence. Dans cette thèse, un cadre hautement comparative des méthodes, pour modeler les séries chronologiques est performer la prédiction par ces modèles notamment les modèles linéaires. Pour le faire, on a réduit les erreurs des modèles par la méthode des algorithmes génétiques (AGs), une des méthodes robuste d'optimisation. AGs étaient inspirée de la théorie de l'évolution naturelle par Holland (1975), appliquent des opérations de reproduction, de croisement et de mutation sur les solutions candidates selon leurs scores de la fonction objective relative a la population des individus. Nous utilisons la moyenne quadratiques des erreurs (MSE), le critère d'information Akaike (AIC) et le critère de Bayes (BIC), comme des fonctions d'objectives à minimiser, puisque ils sont des critères de sélection du modèle optimal. Cette méthode est basée sur la codification, nous utilisons deux types de codage, le codage binaire (BCGA) et le codage réel (RCGA). Dans un premiers temps, on utilise la méthode BCGA pour l'optimisation et en deuxième temps, on fait l'optimisation par la méthode RCGA. Nous effectuons une étude comparative entre les différents types des AGs avec des modifications apportées sur les erreurs des modèles par la normalisation des erreurs pour avoir des modèles optimaux. Les résultats de simulation informatique obtenus démontrent que GAs ont le potentiel pour devenir un outil puissant pour les modèles des séries chronologiques et performer la prévision, notamment lorsque nous utilisons AGs avancées. Pour illustrer ces résultats d'étude nous renforçons les chapitres par des exemples d'applications sur des séries chronologiques réelles.

Mots-clefs

Séries chronologiques; methode de Box-Jenkins, génétique algorithmes en codage binaire(BCGA); génétique algorithmes en codage real(RCGA).

Contents

List of Figures	11
List of Tables	13
List of abbreviations	15
Introduction	17
1 Time series	19
1.1 Time series decomposition	20
1.2 White noise	20
1.3 Stationarity	21
1.3.1 Back-shift operator	21
1.3.2 Difference operator	21
1.3.3 Trend stationarity	22
1.3.4 Differences stationarity	22
Dickey-fuller test (DF)	23
Augmented Dickey-Fuller test (ADF)	23
1.4 Auto-covariance and Autocorrelation functions	23
1.4.1 Auto-covariance function (ACOVF)	23
1.4.2 Autocorrelation function (ACF)	24
1.4.3 Partial autocorrelation function PACF	24
1.4.4 Sample autocorrelation function SACF	24
1.5 Wold decomposition	25
1.6 Stationary time series Models	25
1.6.1 Autoregressive process AR(P)	25
Sample autocorrelation function of AR(P)	26
Autoregressive process MA(q)	28
1.6.2 Sample autocorrelation function of MA(q)	29
1.6.3 Mixed Autoregressive Moving Average models ARMA(p,q)	30
Autocorrelation properties	31
1.7 Models for non-stationary time series	31
1.7.1 Mixed Integrated Autoregressive Moving Average models ARIMA(p,q)	31
Guide lines for determining the order of the general non-seasonal	
models	32

1.7.2	Seasonal Autoregressive Integrated Moving Average models SARIMA($\mathbf{p,d,q}$)	33
1.8	Box and Jenkins methodology	33
1.8.1	Estimation	34
	Information criteria	34
	Akaike information criteria (AIC)	34
	Shwarz bayesian information criteria (BIC)	34
	Hannan-Quin information criteria (HQIC)	35
1.8.2	Forecasting in ARMA models	35
2	Genetic Algorithms	41
2.1	Definitions and terminology	41
2.1.1	Definitions	41
2.1.2	Objectif function and fitness function	42
2.2	Encoded	43
2.2.1	Binary-coded	43
2.2.2	Gray-coded	44
2.2.3	Real-coded RCGA	44
2.3	Genetic algorithms process	44
2.3.1	Initialization	45
2.3.2	Selection	46
	Probability for Selection	46
	Cumulative fitness distribution	47
	Roulette Wheel selection RWS	49
	Elitism selection ES	50
	Rank Selection	50
	linear ranking selection LRS	50
	Exponential ranking selection ERS	50
	Tournament selection TS	51
	Truncation selection TrS	51
	Boltzman selection BS	51
2.3.3	Crossover	52
	Single-point crossover SPC	52
	Two-point crossover TPC	52
2.3.4	Cut and splice crossover	53
	Uniform crossover and half Uniform crossover	53
2.3.5	Heuristic crossover HC	54
2.3.6	Mutation	54
2.3.7	Types of Mutation	55
2.3.8	Insertion mutation	55
2.3.9	Inversion mutation IM	55
2.3.10	Scramble mutation ScM	56
2.3.11	Swap mutation SwM	56
2.3.12	Flip mutation FIM	56
2.4	Example of application	57

3	Modeling linear Time Series with Binary Genetic Algorithm BCGA	61
3.1	BCGA on Autoregressive models AR(P)	61
3.1.1	Iterative process	66
3.2	BCGA on Autoregressive models MA(q)	67
3.3	BCGA on Seasonal Autoregressive Integrated Moving Average models <i>SARIMA</i> (p, d, q), application study	71
4	Real Genetic Algorithms RCGA and comparison study with BCGA models	73
4.1	Introduction	73
4.2	Crossover	73
4.2.1	Whole arithmetic crossover method AMXO	73
4.2.2	Local arithmetical crossover LAC	74
4.2.3	linear crossover LC	74
4.2.4	Extended line crossover ELC	74
4.2.5	linear BGA crossover (Schlierkamp-Veosen 1994)	74
4.2.6	Flat crossover FC	75
4.2.7	Blend Crossover <i>BLX</i> - α	75
4.2.8	<i>PBX</i> - α crossover	75
4.2.9	Heuristic crossover operator (HX)	75
4.2.10	Parent-centric crossover PCCO	75
4.2.11	laplace crossover operator LX	76
4.2.12	Multiple crossover MX	76
4.3	Mutation operators in real coded genetic algorithms RCGA	76
4.3.1	Power mutation operator PM	77
4.3.2	Boundary mutation BM	77
4.3.3	Random mutation RM	77
4.3.4	Makinen-Periaux and Toivanen mutation	77
4.3.5	Mühlen Bein's mutation	78
4.3.6	Discrete mutation (DX)	78
4.3.7	Continuous mutation CX	78
4.3.8	Creep mutation	78
4.3.9	Uniform mutation UM	78
4.3.10	Non-uniform mutation operator NUM	78
4.4	Comparison study between BCGA and RCGA models	82
5	Conclusion and perspective	83
	Bibliography	85
	General conclusion and perspective	91

List of Figures

1.1	Derives Energy Export For Algeria,1980 to 2011	20
1.2	Model decomposition	21
1.3	Export Derives Energy For Algeria Deffirenced ,1980 TO 2011	22
1.4	The sample autocorrelation function (SACF)	25
1.5	ACF and PACF for AR(1)(0.85)	28
1.6	ACF and PACF for AR(2)(0.3,0.54)	28
1.7	ACF and PACF for MA(1)(0.85)	30
1.8	ACF and PACF for MA(2)(0.5,0.7)	30
1.9	hotel data.train 1985 to 1995	33
1.10	traffic accidents in France	36
1.11	ddata of traffic accident .train residuals	36
1.12	Residuals diagnostic	37
1.13	ddata traffic accident forecast	37
1.14	hotel ddata.train 1985 TO 1995	38
1.15	Residuals diagnostic	38
1.16	forecast fit model	39
2.1	global minimum and local minimum	42
2.2	Diagram cdf of discrete distribution	47
2.3	Diagram cdf of continues distribution	48
2.4	Roulette Wheel	49
2.5	Rank Selection	51
2.6	Crossover in one point	53
2.7	Crossover in two points	53
2.8	Cut and splice crossover	53
2.9	Uniform crossover	54
2.10	Mutation example	55
2.11	Insertion mutation	55
2.12	Inversion mutation	55
2.13	Scramble mutation	56
2.14	Swap mutation	56
2.15	Bit flip mutation	56
2.16	Roulettewoolforexampple	58
2.17	Diagram of parents	59
2.18	crossover point probability	59

2.19	mutation probability	60
3.1	PRODUCTION DERIVE ENERGY FOR ALGERIA 1980 TO 2011	61
3.2	acf and pacf of derive oil production time series	62
3.3	Residuals diagnostic	64
3.4	roulette wheel selection	65
3.5	Single point Crossover	65
3.6	Binary Mutation	66
3.7	Comparison between for $AR(1)$ and $AR(2)$	66
3.8	Comparison between BCGA model and Box-Jenkins model	67
3.9	Hourly euro Index	67
3.10	acf and pacf of time series	68
3.11	diff euro index time series plot	68
3.12	acf and pacf of differentiate time series	69
3.13	Residuals diagnostic	70
3.14	qqnorm of residuals model	70
3.15	Comparison between Box-Jenkins and BCGA models	71
4.1	Analysis on the unimodal function	79
4.2	One point crossover	83
4.3	Two point crossover	83
4.4	Extended linear crossover	84
4.5	Laplace crossover	85
4.6	Arithmetic crossover	85
4.7	Makinen-Periaux and Toivanen mutation	85

List of Tables

1.1	Derives Energy Export For Algeria,1980 to 2011	19
1.2	32
1.3	Behavior of S-ACF and S-PACF of $ARIMA(p, q)$	32
1.4	Box-Jenkins approach	34
1.5	parameters estimation	36
3.1	parameters estimation	62
3.2	parameters estimation	63
3.3	Box and Jenkins methods	63
3.4	Fitness values of individuals	65
3.5	BCGAs method	66
3.6	comparison between genetic algorithms and Box-jenkins models	67
3.7	parameters estimation	69
3.8	Comparison between BCGAs model and Box-Jenkins model	71
4.1	BCGA application	83
4.2	RCGA application	84
4.3	RCGA with various crossover and mutation operators	86

List of abbreviations

TS : trend stationary

DS : differenced stationary

DF : Dickey-fuller test

i.i.d : independents and identically distributed

ADF : Augmented Dickey-Fuller test

AC0VF : Auto-covariance function

ACF : Autocorrelation function

PACF : Partial autocorrelation function

SACF : Sample autocorrelation function

AR(P): Autoregressive process

MA(q): Autoregressive process

ARMA(p, q): Mixed Autoregressive Moving Average models

ARIMA(p, d, q) : Mixed Integrated Autoregressive Moving Average models

SARIMA(p, d, q): Seasonal Autoregressive Integrated Moving Average models *SARIMA*(p, d, q)

AIC : Akaike information criteria

BIC : Shwarz bayesian information criteria

HQIC : Hannan-Quin information criteria

GAs : Genetic algorithms

Pop : population

BCGA : Binary-coded Genetic algorithms

CFD : Cumulative fitness distribution

\bar{f} : Average fitness

$\bar{\sigma}^2$: Fitness variance

$\bar{R}(f)$: Reproduction rate

P_d : Loss of diversity

RWS : Roulette Wheel selection

ES : Elitism selection

LRS : linear ranking selection

ERS : Exponential linear ranking selection

TS : Tournament selection

TrS : Truncation selection

BS : Boltzman selection

SPC : Single-point crossover

TPC : Two-point crossover

HC : Heuristic crossover

IM : Inversion mutation
ScM : Scramble mutation
SwM : Swap mutation
FIM : Flip mutation
InM : interchanging mutation
UM : uniform mutation
RCGA : real-coded Genetic algorithms
AMXO : Whole arithmetic crossover method
LAC : Local arithmetical crossover
SAC : Simple arithmetical crossover
LC : linear crossover
ELC : Extended line crossover
DC : Discrete crossover
FC : Flat crossover
BLX – α crossover
PBX – α crossover
PCCO : Parent-centric crossover
LX : laplace crossover operator
MX : Multiple crossover
NUM : Non-uniform mutation operator
PM : Power mutation operator
BM : Boundary mutation
RM : Random mutation
CX : Continuous mutation

Introduction

Time series refers to the branch of statistics where observations are collected sequentially in time. Time series modeling and forecasting have fundamental importance to various practical. Which deals with techniques developed for drawing the inference from time series(50). The primary goals of time series analysis are:

- 1) To set up a hypothetical statistical model to represent the series in order to obtain insights into mechanism that generate the data.
- 2) Once a static factory model has been formulated, to extrapolate from the in order anticipate (forecast) the future values of time series.

In many fields like it necessary to accurately predict future values of time series. The correct estimation of future of values is usually affected by complex processes like random fluctuations, sudden trend change, volatility, and noise.

Different models employed to estimate time series have been differentiated in two groups: linear and non-linear methods. The most popular linear method are based on the Box-Jenkins methodology [49]. In the early days, most of the models proposed were linear regression models. Their implementation is simple, yet they are quite limited in capabilities of interpreting time series. They are not capable of dealing with nonlinear and/or nonstationary behavioral patterns. They always assume the systems from which the time series has been measured to be linear and to operate under stationary conditions (Priestley 1981; Ljung 1987; Chatfield 1989; Box and Jenkins 1994).

The main aim of time series modeling is to carefully collect and rigorously study the past observations of a time series to develop an appropriate model which describes the inherent structure of the series.

It is obvious that a successful time series forecasting depends on an appropriate model fitting.

the mean of the problem for all researchers gives the optimal model fitted for data and reduce errors between observations and generate observations in order to have a good forecast. For seasonal time series forecasting, Box and Jenkins [6] had proposed a quite successful variation of ARIMA model, viz. the Seasonal ARIMA (SARIMA) [3, 6, 23]. The popularity of the ARIMA model is mainly due to its flexibility to represent several varieties of time series with simplicity as well as the associated Box-Jenkins methodology [3, 6, 8, 23] for optimal model building process.

Genetic algorithms are based on the principle of genetics and evolution stated Charles Darwin the theory of natural evolution, and In 1975, Holland described and developed this idea in his book Adaptation in natural and artificial systems and how to apply his principles to optimization problems.

Genetic algorithms are inspired by Darwin's theory of natural evolution. GAs is a method for solving both constrained and non-constrained optimization problems based on natural selection process that mimics biological evolution.

The problem resolution of optimization consists of exploring the space of research to minimize or maximize an objected function. GAs are optimization and robust searching technique that is used to find the best search result is composed mainly of three steps: recombination crossover and mutation. By maintaining a population of a solution, GAs can be viewed as implicitly modeling of solutions seen in the search process. The aim of this thesis is to present time series modeling using GAs, a comprehensive discussion and comparison of the three widely popular approaches for time series forecasting, binary coded GAs, real coded GAs, and Box-Jenkins approaches.

Many techniques in literature are proposed to develop this method. This thesis contains fours chapters, which are organized as follows: Chapter 1 gives an introduction to the basic concepts of time series modeling, together with some associated ideas such as stationarity, parsimony, etc.

Chapter 2 is designed to discuss the concepts of GAs, and there operators: selection, crossover, and mutation.

In Chapter 3 we have described the application of binary genetic algorithms BCGAs in time series modeling, for *AR*, *MA*, *ARMA* and *SARIMA* models and comparison with Box-Jenkins approach.

Chapter 4 presents the real coded genetic algorithms RCGAs approach and comparison between three approaches RCGAs, BCGAs, and Box-Jenkins.

After completion of these four chapters, we have given a brief conclusion of our work as well as the prospective future aim in this field.

Chapter 1

Time series

Définition 1.0.1. A time series is a sequence of observation of any random phenomenon measured at deferent points of time t and usually denoted $\{X_t\}_{t \in Z}$.

Example 1.0.2. The average years for Derives Energy Export For Algeria, from 1980 to 2011.

Table 1.1: Derives Energy Export For Algeria,1980 to 2011

YEAR	Export energy derived*10 ³	YEAR	Export energy derived*10 ³	YEAR	Export energy derived *10 ³
1980	56344	1991	85207	2002	123001
1981	54041	1992	87157	2003	132714
1982	54520	1993	86801	2004	136891
1983	60496	1994	83335	2005	145274
1984	64241	1995	88846	2006	140716
1985	68043	1996	95796	2007	138405
1986	68730	1997	104840	2008	134724
1987	73940	1998	109706	2009	121949
1988	72933	1999	118775	2010	119751
1989	79478	2000	124125	2011	114158
1990	83978	2001	119013		

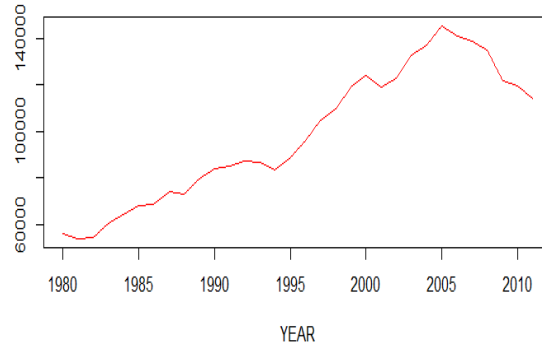


Figure 1.1: Derives Energy Export For Algeria,1980 to 2011

1.1 Time series decomposition

The time series can be decomposed into four elements:

- Trend (T_t): Long term movements in the mean;
- Seasonal effects (L_t): cyclical fluctuations related to the calendar;
- Cycles (C_t): Other cyclical fluctuations (such as a business cycle)
- Residuals (ϵ_t): Other random or systematic fluctuation.

And two models that allow as to do are:

- Additive decomposition model:

$$X_t = T_t + L_t + C_t + \epsilon_t \quad (1.1)$$

- Multiplicative decomposition model:

$$X_t = T_t \times L_t \times C_t \times \epsilon_t. \quad (1.2)$$

Example 1.1.1. Let us show figure of two time series:

The first has Additive decomposition model and the second has Multiplicative decomposition model.

1.2 White noise

Définition 1.2.1. The process $(\epsilon_t)_{t \in Z}$ is a strong white noise if:

- $(\epsilon_t)_{t \in Z}$ is a sequence of random variables i.i.d (independents and identically distributed).
- $\forall t \in Z, E(\epsilon_t) = 0$, and $E(\epsilon_t^2) = \sigma^2$

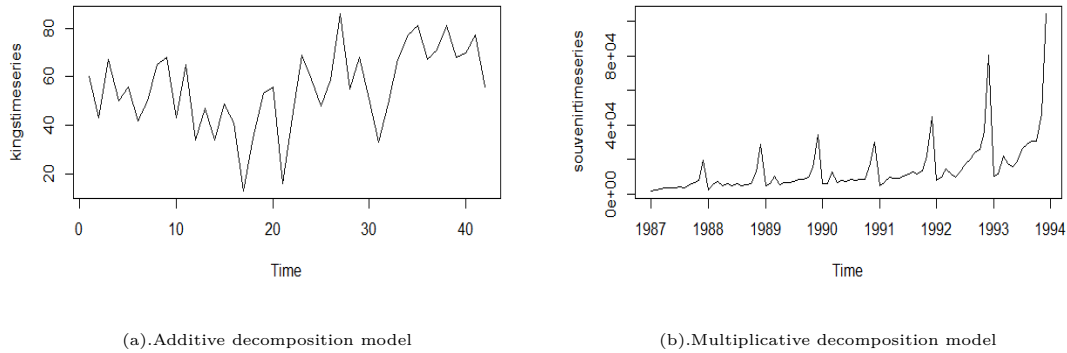


Figure 1.2: Model decomposition

Définition 1.2.2. The process $(\epsilon_t)_{t \in Z}$ is a weak white noise if:

- i) $(\epsilon_t)_{t \in Z}$ is a sequence of random variables i.d(identically distributed),
- ii) $\forall (t, t') \in Z^2: t \neq t', cov(\epsilon_t, \epsilon_{t'}) = 0$,
- iii) $\forall t \in Z, E(\epsilon_t) = 0$ and $E(\epsilon_t^2) = \sigma^2$.

1.3 Stationarity

Définition 1.3.1. A time series $X_t, t \in Z$ is strictly stationary or strongly stationary if: for $k \geq 1$ the vectors $(X_{t_1}, X_{t_2}, \dots, X_{t_k})$ and $(X_{t_1+h}, X_{t_2+h}, \dots, X_{t_k+h})$ have identical law and joint probability distribution. i.e:

Définition 1.3.2. A time series X_t is weakly stationary or second order stationary if :

- i) $E(X_t) = \mu; \forall t \in Z$; (constant)
- ii) $var(X_t) = \sigma^2; \forall t \in Z$; (constant)
- iii) $\forall r, s, t \in Z : cov(X_{r+t}, X_{s+t}) = cov(X_r, X_s)$.

Example 1.3.3. • Strong white noise is strictly stationary.

- Weak white noise is second order stationary.

1.3.1 Back-shift operator

The back shift operator denoted by B , such as $B(y_t) = y_{t-1}$.

Remark 1.3.4. • $B^2(y_t) = B(B(y_t)) = y_{t-2}$, and $B^k(y_t) = y_{t-k}$,

- $B(ay_t + b) = aB(y_t) + b = ay_{t-1} + b$, for any constant a, b .

1.3.2 Difference operator

Définition 1.3.5. The Difference operator ∇ is defined for X_t process by :

$$\nabla(X_t) = X_t - X_{t-1} = (I - B)X_t. \quad (1.3)$$

Where I is identic operator and B is Back-shift operator.

Remark 1.3.6. - $\nabla^d = (I - B)^d, d \in N$.

- for $d = 2, \nabla^2 = (I - B)^2 = I - 2B + B^2$

- If $X_t = at + b + \epsilon_t$, where ϵ_t is stationary, then $\nabla(X_t) = a + (\epsilon_t - \epsilon_{t-1})$ (remove the trend);

- In general, the operator ∇^d remove the polynomials of d degree.

1.3.3 Trend stationarity

Définition 1.3.7. A time series X_t is said to be No-stationary (TS) or trend stationary if : $X_t = f(t) + y_t$, when f is a determinists function, and $(y_t)_{t \in Z}$ is a stationary process .

Example 1.3.8.

$$X_t = \alpha + \beta t + \epsilon_t,$$

X_t is trend stationary (TS).

$$E(X_t) = \alpha + \beta t,$$

depend on t.

1.3.4 Differences stationarity

Définition 1.3.9. A time series X_t is No-stationary (DS) or differenciated stationary if X_t is stationary after d differentiations:

$$\nabla^d X_t = (1 - B)^d X_t. \quad (1.4)$$

Example 1.3.10. Plot for time series 1.1 after differenciated:

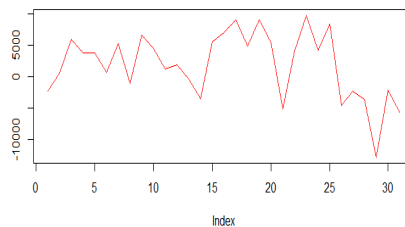


Figure 1.3: Export Derives Energy For Algeria Deffired ,1980 TO 2011

Proposition 1.3.11. If $X_t, t \in Z$ is a stationary process, and $a_i, i \in Z$, is a real sequence absolutely convergent i.e $\sum_{i \in Z} |a_i| < +\infty$, then the process y_t defined by: $y_t = \sum_{i \in Z} a_i X_{t-i}$, for $t \in Z$, is a stationary process.

Proof. See [10]. □

Remark 1.3.12. For make data stationary, we may do any of the following:

- i) Re-scale it (for instance, by a logarithmic or exponential transformation).
- ii) Remove deterministic components.
- iii) Difference it. That is, take $\nabla^d(X_t)$ until stationary.

So we need the test of stationary. We can use Dickey-fuller tests (DF) and Augmented Dickey-Fuller tests (ADF)

Dickey-fuller test (DF)

Tests of Dickey-Fuller is the unit root tests when the null hypothesis is no stationarity carried out the process autoregressive one $AR(1)$ such as ε_t is i.i.d, $(0, \sigma_\varepsilon^2)$ for the models:

- i) $X_t = \beta_1 X_{t-1} + c + \alpha \times t + \varepsilon_t$
- ii) $X_t = \beta_1 X_{t-1} + c + \varepsilon_t$
- iii) $X_t = \beta_1 X_{t-1} + \varepsilon_t$

Augmented Dickey-Fuller test (ADF)

When ε_t is not i.i.d, $(0, \sigma_\varepsilon^2)$ and $p \geq 2$, we need use (ADF), is generalized for (DF) and same threshold for signification to this tree models :

- i) $X_t = \beta_1 X_{t-1} + \sum_{j=1}^p \xi_j \Delta x_{t-j} + c + \alpha t + \varepsilon_t$
- ii) $X_t = \beta_1 X_{t-1} + \sum_{j=1}^p \xi_j \Delta x_{t-j} + c + \varepsilon_t$
- iii) $X_t = \beta_1 X_{t-1} + \sum_{j=1}^p \xi_j \Delta x_{t-j} + \varepsilon_t$

we testing by (ADF) and (DF), unit root hypothesis :

$$H_0 : \beta_1 = 1 ; H_1 : |\beta_1| < 1$$

Remark 1.3.13. But the strategic of the tests (ADF) and (DF) is testing the general model with trend and constant model (i), if the test statics is less than critical-value and p-value less 0.05, then we can reject null hypothesis and accept alternative hypothesis, but if the test statics is more than critical-value and p-value more 0.05, then we can not reject null hypothesis and series is non-stationary.

1.4 Auto-covariance and Autocorrelation functions**1.4.1 Auto-covariance function (ACOVF)**

Définition 1.4.1. The sequence of covariance between X_t and X_{t-k} is called the auto-covariance function denoted by γ_k , $k \in Z$:

$$\gamma_k = cov(X_t, X_{t-k}). \tag{1.5}$$

Remark 1.4.2. For $k = 0$, $\gamma_0 = var(X_t)$.

properties 1.4.3. The Auto-covariance function verifies the next properties:

- $\gamma_0 \geq 0$
- $|\gamma_k| \leq \gamma_0$,
- γ_k is a symmetry function: $\gamma_{-k} = \gamma_k$

1.4.2 Autocorrelation function (ACF)

Définition 1.4.4. The sequence denoted $\rho_k = \frac{\gamma_k}{\gamma_0} = \text{corr}(X_t, X_{t-k})$ is called autocorrelation function (ACF).

properties 1.4.5. • $\rho_0 = 1$

- $\rho_{-k} = \rho_k$
- $|\rho_k| \leq 1$

Example 1.4.6. For white noise, $E(\epsilon_t) = 0$, $\gamma_0 = \sigma^2$, for all t ;

$$\gamma_k = \text{corr}(\epsilon_t, \epsilon_{t-k}) = \begin{cases} \sigma_t^2, & k=0; \\ 0, & k \neq 0. \end{cases}$$

$$\rho_k = \frac{\gamma_k}{\gamma_0} = \begin{cases} 1, & k=0; \\ 0, & k \neq 0. \end{cases}$$

1.4.3 Partial autocorrelation function PACF

Définition 1.4.7. We call the partial autocorrelation function of X_t at lag k , the function :

$$\begin{cases} r(k) = \text{corr}(X_t, X_{t-k} \setminus X_{t-1}, X_{t-2}, \dots, X_{t-k+1}), & \text{for } k \in N^*; \\ r(1) = \rho(1), & \text{for } k = 1. \end{cases} \quad (1.6)$$

1.4.4 Sample autocorrelation function SACF

Most of the analysts using sampled data, such if we suppose X_t is stationary, for $t \in N$, we can use sample autocorrelation function (SACF) denoted $\hat{\rho}_k$ by substituted $\hat{\gamma}_k$ the estimate of auto-covariance function γ_k in ρ_k formula where:

$$\hat{\gamma}_k = \frac{1}{n} \sum_{t=1}^{n-k} (x_t - \bar{x})(x_{t-k} - \bar{x}), t \in N \quad (1.7)$$

Définition 1.4.8. The sample autocorrelation function (SACF) is defined as:

$$\hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0}. \quad (1.8)$$

Remark 1.4.9. 1) In the SACF of the time series values X_1, X_2, \dots, X_n is either cuts off fairly quickly or dies down fairly quickly, then the time series values should be considered stationary.

2) If the SACF of the time series values X_1, X_2, \dots, X_n dies down extremely slowly, then the time series values should be considered non-stationary.

Example 1.4.10. Example of the SACF two time series:

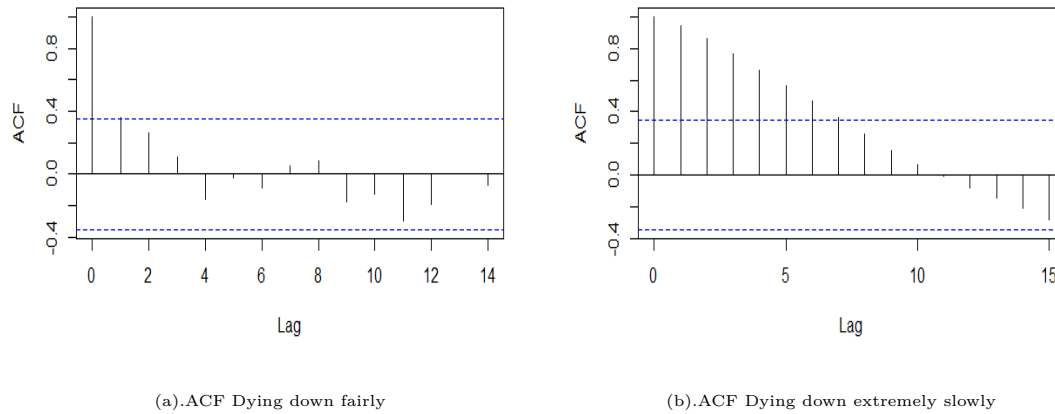


Figure 1.4: The sample autocorrelation function (SACF)

1.5 Wold decomposition

Théorème 1.5.1. *Any mean zero covariance stationary process X_t can be represented in the form:*

$$X_t = \sum_{j=0}^{\infty} \theta_j \epsilon_{t-j} + \eta_t \quad (1.9)$$

Where:

- 1) $\theta_0 = 1$, and $\sum_{j=0}^{\infty} \theta_j^2 < \infty$,
- 2) $\epsilon_t \rightsquigarrow N(0, \sigma_t^2)$,
- 3) $E(\epsilon_t, \eta_t) = 0, \forall s, t > 0$,
- 4) ϵ_t is the error in forecasting X_t on the basis of linear of lagged n :
 $\epsilon_t = X_t - E(X_t | X_{t-1}, X_{t-2}, \dots, X_{t-n})$,
- 5) η_t is a deterministic process, and it can be predicted from a linear function of lagged n .

Proof. See [25]. □

Remark 1.5.2. Wold decomposition says that any covariance stationary process has a linear presentation: linear deterministic components (η_t) and linear indeterminate components (ϵ_t).

1.6 Stationary time series Models

1.6.1 Autoregressive process AR(P)

Définition 1.6.1. The process $\{X_t\}_{t \in Z}$ is said to be an autoregressive process of order p , denoted $AR(p)$, if it is stationary and

$$X_t = \sum_{i=1}^p \theta_i X_{t-i} + \epsilon_t. \quad (1.10)$$

For all $t \in Z$ and $\theta_p \neq 0$. Where θ_i are reals and ϵ_t is white noise, his variance σ^2 .

Remark 1.6.2. We can rewrite equality :

$$\Phi(B)X_t = \epsilon_t \quad (1.11)$$

Where the polynomial $\Phi(B) = I - \theta_1 B - \theta_2 B^2 - \dots - \theta_p B^p$.

In general the forme of $AR(p)$ with constant is writen:

$$\Phi(B)X_t = \mu + \epsilon_t, \quad (1.12)$$

μ is a constant.

Example 1.6.3. - For $p = 1$:

$$AR(1) : X_t = \theta_1 X_{t-1} + \epsilon_t, t \in Z \quad (1.13)$$

- For $p = 2$:

$$AR(2) : X_t = \theta_1 X_{t-1} + \theta_2 X_{t-2} + \epsilon_t, t \in Z \quad (1.14)$$

Sample autocorrelation function of AR(P)

Proposition 1.6.4. *If $\{X_t\}_{t \in Z}$ is the autoregressive process $AR(p)$:*

$X_t = \sum_{i=1}^p \theta_i X_{t-i} + \epsilon_t$, where $t \in Z$, $\theta_i \in R$, $i = \overline{1, p}$ and ϵ_t is white noise, then:

$$\gamma_0 = \frac{\sigma^2}{1 - \sum_{i=1}^p \theta_i \rho_i} \quad (1.15)$$

and

$$\rho_h = \sum_{i=1}^p \theta_i \rho_{h-i}. \quad (1.16)$$

Proof. We have :

$$\begin{aligned} \gamma_0 &= cov(X_t, X_t), \\ &= cov(X_t, \sum_{i=1}^p \theta_i X_{t-i} + \epsilon_t), \\ &= \sum_{i=1}^p \theta_i cov(X_t, X_{t-i}) + cov(X_t, \epsilon_t), \\ &= \sum_{i=1}^p \theta_i \gamma_i + cov(\sum_{i=1}^p \theta_i X_{t-i} + \epsilon_t, \epsilon_t), \\ &= \sum_{i=1}^p \theta_i \gamma_i + cov(\epsilon_t, \epsilon_t), \\ &= \sum_{i=1}^p \theta_i \gamma_0 \rho_i + \sigma^2. \end{aligned}$$

Then,

$$\gamma_0 = \frac{\sigma^2}{1 - \sum_{i=1}^p \theta_i \rho_i}.$$

In the same : $\forall h \in N^*$,

$$\begin{aligned}
\gamma_h &= \text{cov}(X_t, X_{t-h}), \\
&= \text{cov}\left(\sum_{i=1}^p \theta_i X_{t-i} + \epsilon_t, X_{t-h}\right), \\
&= \sum_{i=1}^p \theta_i \text{cov}(X_{t-i}, X_{t-h}) + \text{cov}(X_t, X_{t-h}), \\
&= \sum_{i=1}^p \theta_i \gamma_{h-i}.
\end{aligned}$$

□

Remark 1.6.5. We obtain the system called the Walker-equations:

$$\gamma_0 = \frac{\sigma^2}{1 - \sum_{i=1}^p \theta_i \rho_i}.$$

$$\begin{pmatrix} \rho_1 \\ \rho_2 \\ \cdot \\ \cdot \\ \cdot \\ \rho_p \end{pmatrix} = \begin{pmatrix} 1 & \rho_1 & \rho_2 & \cdot & \cdot & \rho_p \\ \rho_1 & 1 & \cdot & \cdot & \cdot & \cdot \\ \rho_2 & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ \rho_p & \cdot & \cdot & \cdot & \cdot & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \cdot \\ \cdot \\ \cdot \\ \theta_p \end{pmatrix} = R_p \theta$$

Where R_p is the matrices of correlation and $\theta = (\theta_1, \theta_2, \dots, \theta_p)$

Remark 1.6.6. If the roots of characteristic polynomial $\Phi(z)$; $z_i = \frac{1}{\lambda_i}$, $i \in \{1, 2, \dots, p\}$ are reals and distinct, we obtain:

$$\rho_h = \sum_{i=1}^p c_i \lambda_i^h. \quad (1.17)$$

Example 1.6.7. For $AR(1)$:

$$\begin{cases} \gamma_0 = \frac{\sigma^2}{1-\theta_1^2} \\ \gamma_k = \frac{\theta_1^k \sigma^2}{1-\theta_1^2} \\ \rho_k = \theta_1^k \end{cases} \quad (1.18)$$

$k = 1, 2, \dots$

For $AR(2)$:

$$\begin{cases} \gamma_0 = \frac{\sigma^2}{1-\theta_1\rho_1-\theta_2\rho_2} \\ \rho_1 = \frac{\theta_1}{1-\theta_2} \\ \rho_2 = \frac{\theta_2-\theta_2^2+\theta_1^2}{1-\theta_2} \end{cases} \quad (1.19)$$

Example 1.6.8. $AR(1)$: $X_t = 0.85X_{t-1} + \epsilon_t$, $t \in Z$

$AR(2)$: $X_t = 0.3X_{t-1} + 0.54X_{t-2} + \epsilon_t$, $t \in Z$

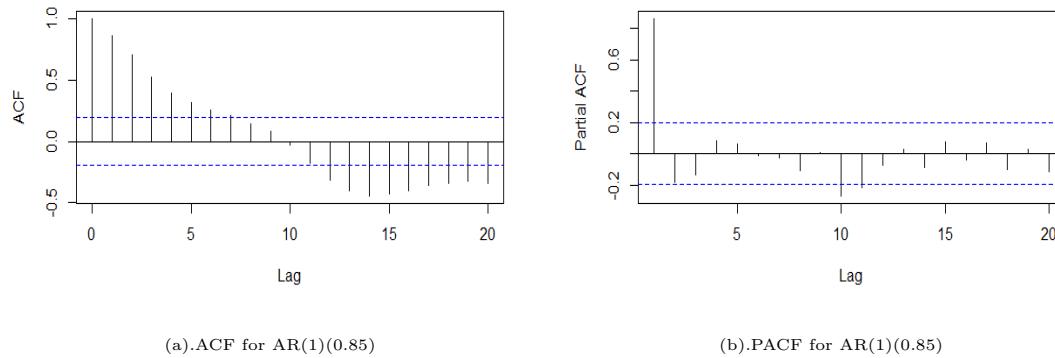


Figure 1.5: ACF and PACF for AR(1)(0.85)

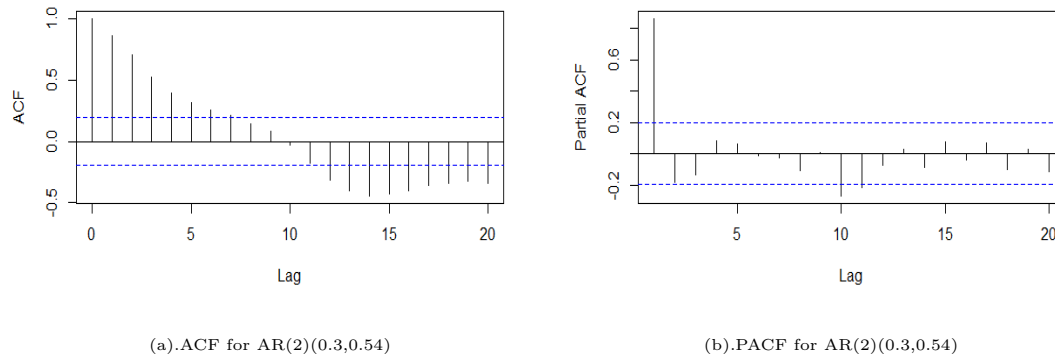


Figure 1.6: ACF and PACF for AR(2)(0.3,0.54)

Remark 1.6.9. We can remark, for the $AR(p)$ process, the PACF is zero for all lags beyond p , i.e $r(k) = 0$

Autoregressive process MA(q)

Définition 1.6.10. The process $\{X_t\}_{t \in \mathbb{Z}}$ is said to be moving average process $MA(q)$ of order q if:

$$X_t = \epsilon_t + \sum_{i=1}^q \varphi_i \epsilon_{t-i}, \quad (1.20)$$

$\forall t \in \mathbb{Z}$

where ϵ_t is white noise and $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_q)$, $\varphi_q \neq 0$

Remark 1.6.11. - We can use notation

$$X_t = \Theta(B)\epsilon_t \quad (1.21)$$

where $\Theta(B) = I + \sum_{i=1}^q \varphi_i B^i$.

- The moving average process is stationary.

Example 1.6.12. For $q=1$ the model $MA(1)$:

$$X_t = \epsilon_t + \varphi_1 \epsilon_{t-1} \quad (1.22)$$

For $q=2$ the model $MA(2)$:

$$X_t = \epsilon_t + \varphi_1 \epsilon_{t-1} + \varphi_2 \epsilon_{t-2} \quad (1.23)$$

1.6.2 Sample autocorrelation function of $MA(q)$

Proposition 1.6.13. If $\{X_t\}_{t \in Z}$ is the moving average process $MA(q)$:

$X_t = \epsilon_t + \sum_{i=1}^q \varphi_i \epsilon_{t-i}$, $\forall t \in Z$, where ϵ_t is white noise and $\varphi_i \in R$, $i = \overline{1, q}$, then:

$$\gamma_0 = \sigma^2 \left(1 + \sum_{i=1}^q \varphi_i^2\right), \quad (1.24)$$

$$\gamma_h = \begin{cases} \varphi_h + \sum_{i=1}^q \varphi_i \varphi_{i-h} & \text{if } h \in \{1, 2, \dots, q\} \\ 0 & \text{if } h > q \end{cases} \quad (1.25)$$

Proof. Let us

$$\begin{aligned} \gamma_0 &= \text{var}\left(\epsilon_t + \sum_{i=1}^q \varphi_i \epsilon_{t-i}\right) \\ &= \text{cov}\left(\sum_{i=1}^q \theta_i X_{t-i} + \epsilon_t, X_{t-h}\right), \\ &= \sigma^2 \left(1 + \sum_{i=1}^q \varphi_i^2\right) \end{aligned}$$

$\forall h \in N$:

$$\begin{aligned} \gamma_h &= \text{cov}(X_t, X_{t-h}) \\ &= \text{cov}\left(\epsilon_t + \sum_{i=1}^q \varphi_i \epsilon_{t-i}, \epsilon_{t-h} + \sum_{j=1}^q \varphi_j \epsilon_{t-h-j}\right) \\ &= \begin{cases} \varphi_h + \sum_{i=1}^q \varphi_i \varphi_{i-h} & \text{if } h \in \{1, 2, \dots, q\} \\ 0 & \text{if } h > q \end{cases} \end{aligned}$$

□

Remark 1.6.14. If $\{X_t\}_{t \in Z}$ is the moving average process of order q ($MA(q)$), then his S-ACF are Zero for all lags beyond q i.e:

$$\begin{cases} \rho(h) \neq 0 & h = q \\ \rho(h) = 0 & \forall h \geq q + 1. \end{cases} \quad (1.26)$$

Example 1.6.15. For $q = 1$, $MA(1)$:

$$\gamma_k = \begin{cases} \sigma^2(1 + \varphi_1^2) & \text{for } k = 0, \\ -\varphi_1 \sigma^2 & \text{for } k = 1, \\ 0 & \text{for } k \geq 2. \end{cases} \quad (1.27)$$

$$\rho(k) = \begin{cases} \frac{-\varphi_1}{1+\varphi_1^2} & \text{for } k = 1 \\ 0 & \text{for } k \geq 2 \end{cases} \quad (1.28)$$

For $q = 2$, $MA(2)$:

$$\rho(k) = \begin{cases} \frac{-\varphi_1 + \varphi_1\varphi_2}{1 + \varphi_1^2 + \varphi_2^2} & \text{for } k = 1, \\ \frac{-\varphi_1}{1 + \varphi_1^2 + \varphi_2^2} & \text{for } k = 2, \\ 0 & \text{for } k \geq 3. \end{cases} \quad (1.29)$$

Example 1.6.16. $MA(0.2)$: $X_t = \epsilon_t + 0.2\epsilon_{t-1}$,
 $MA(0.5, 0.7)$: $X_t = \epsilon_t + 0.5\epsilon_{t-1} + 0.7\epsilon_{t-1}$.

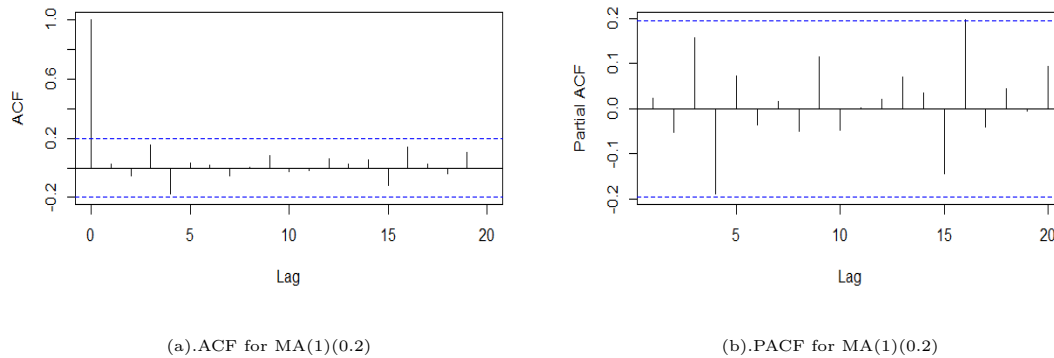


Figure 1.7: ACF and PACF for MA(1)(0.85)

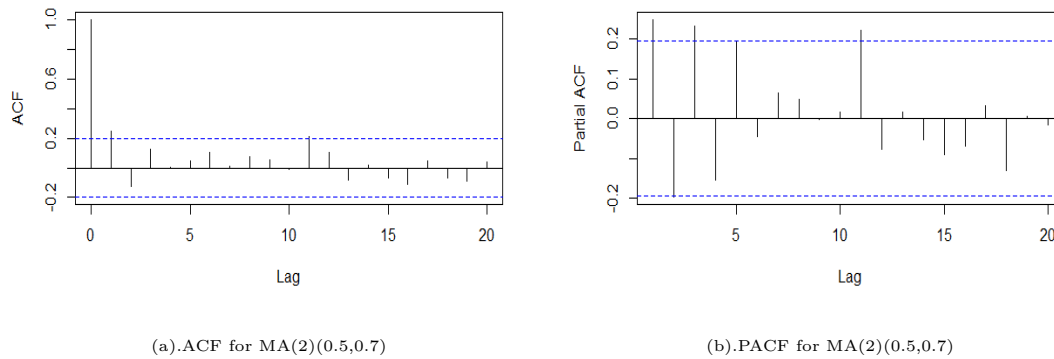


Figure 1.8: ACF and PACF for MA(2)(0.5,0.7)

1.6.3 Mixed Autoregressive Moving Average models ARMA(p,q)

Définition 1.6.17. We say that $\{X_t\}_{t \in \mathbb{Z}}$ is a mixed autoregressive moving average, process of order (p, q) denoted ARMA(p,q) if it satisfies:

- $\{X_t\}_{t \in \mathbb{Z}}$ is stationary,

- It exist $\theta = (\theta_1, \theta_2, \dots, \theta_p) \in R^p$, $\theta_p \neq 0$ and $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_q) \in R^q$, $\varphi_q \neq 0$; then for all $t \in Z$:

$$X_t - \sum_{i=1}^p \theta_i X_{t-i} = \epsilon_t + \sum_{i=1}^q \varphi_i \epsilon_{t-i}, \quad (1.30)$$

Remark 1.6.18. Using the shift operator B equation 1.30 may be rewritten as:

$$\Phi(B)X_t = \Theta(B)\epsilon_t \quad (1.31)$$

where $\Phi(B) = I - \sum_{i=1}^p \theta_i B^i$ and $\Theta(B) = I + \sum_{i=1}^q \varphi_i B^i$.

- $AR(p)$ is an $ARMA(p, 0)$ and $MA(q)$ is an $ARMA(0, q)$.
- The process is stationary if all roots of the characteristic equation

$$y^p - \theta_1 y^{p-1} - \theta_2 y^{p-2} - \dots - \theta_p = 0$$

lie outside the unit circle.

- The representation $\Phi(B)X_t = \Theta(B)\epsilon_t$ is:

1. Minimum: if Φ and Θ have not a commune factors,
2. Causal: if all roots of Φ are outside the unit circle,
3. Invertible: if all roots of Θ are outside the unit circle,
4. Canonic: if the representation is causal and invertible.

Example 1.6.19. Let us example of $ARMA(1, 1)$ and $ARMA(1, 1)$:

$$\begin{aligned} ARMA(1, 1) \text{ model: } X_t &= 0.54X_{t-1} + \epsilon_t + 0.85\epsilon_{t-1} \\ ARMA(2, 1) \text{ model: } X_t &= -0.25X_{t-1} + 0.60X_{t-2} + \epsilon_t + 0.35\epsilon_{t-1} \end{aligned}$$

Autocorrelation properties

Proposition 1.6.20. Let us $\{X_t\}_{t \in Z}$ is $ARMA(p, q)$ process, then auto-covariances function γ_h satisfies:

1. $\gamma_h - \sum_{i=1}^p \theta_i \gamma_{h-i} = 0$ for $h \geq q + 1$,
2. $\gamma_h - \sum_{i=1}^p \theta_i \gamma_{h-i} = \sigma^2[\varphi_h + h_1 \varphi_{h+1} + \dots + h_{q-h} \varphi_q]$ for $0 \leq h \leq q$,

Proof. See [8] □

Remark 1.6.21. The *PACF* of an invertible $ARMA$ model will not cut off. The following table summarized the behavior of *PACF* of the causal and invertible $ARMA$ models .

1.7 Models for non-stationary time series

1.7.1 Mixed Integrated Autoregressive Moving Average models ARIMA(p,q)

Définition 1.7.1. A process $\{X_t\}$ is said an integrated auto-regressive moving average model if d^{th} difference, $Z_t = \nabla^d X_t$ is stationary $ARMA$ process.

Table 1.2

.	$AR(p)$	$MA(q)$	$ARMA(p,q)$
ACF	Tails off	cuts off after lag q	Tails off
$PACF$	cuts off after lag p	Tails off	Tails off

Remark 1.7.2. More generally, the $ARIMA(p, d, q)$ model for $\{X_t\}_{t \in Z}$ is written as:

$$\Phi(B)(1 - B)^d X_t = \mu + \Theta(B)\epsilon_t \quad (1.32)$$

where $\Phi(B) = I - \sum_{i=1}^p \theta_i B^i$, $\Theta(B) = I + \sum_{i=1}^q \varphi_i B^i$ and μ is a constant.

Guide lines for determining the order of the general non-seasonal models

Table 1.3: Behavior of S-ACF and S-PACF of $ARIMA(p, q)$

Behavior of S-ACF and S-PACF	Determination of orders
The S-ACF dies extremely slowly and S-ACF for the first differential, data either cuts off fairly quickly or dies down fairly quickly	$d = 1$
The S-ACF for the first differential, data dies down extremely slowly and the second differential data either cuts off fairly quickly or dies down fairly quickly	$d = 2$
The S-ACF has spikes at lags 1,2,3,...,k, and cuts off after lags, and the S-PACF dies down	$q = k; p = 0$
The S-ACF dies down and the S-PACF has spikes at lags 1,2,...,l, and cuts off after lag l	$p = l; q = 0$
The S-ACF has spikes at lags 1,2,...,k, and cuts off after lag k and the S-PACF has spikes at lags 1,2,...,l, and cuts off after lag l	$q=k$ or $p=l$ and choose best model
The S-ACF dies down and the S-PACF dies down	small values for both p and q
The S-ACF contains small sample correlation and the S-PACF contains small sample correlation.	white noise no seasonal model

1.7.2 Seasonal Autoregressive Integrated Moving Average models SARIMA(p,d,q)

Définition 1.7.3. Since, s_1, s_1, \dots, s_n are n integers, the process $\{X_t\}$ is the SARIMA(p,d,q) if satisfies the relation:

$$\Phi(B)(1 - B^{s_1})(1 - B^{s_2})\dots(1 - B^{s_n})X_t = \Theta(B)\epsilon_t, \quad (1.33)$$

for all $t \geq 0$.

where $\Phi(B) = I - \sum_{i=1}^p \theta_i B^i$ and $\Theta(B) = I + \sum_{i=1}^q \varphi_i B^i$ with $\theta_p \neq 0$, $\varphi_p \neq 0$ and ϵ_t is white noise with σ^2 variance.

Remark 1.7.4. - If we take $n = d, s_1 = s_2 = \dots, s_n = 1$ the SARIMA(p,d,q) model is an ARIMA(p,d,q) model,i.e SARIMA models are generalization of ARIMA models include seasonal part.

- SARIMA(p,d,q) = ARIMA(p,d,q)(P,D,Q)S, where (p,d,q)(non-seasonal part of the model), (P,D,Q)(seasonal part of the model) and S is the number of seasons in year.

- The forms, more used are :

$$\Phi(B)(1 - B^s)X_t = \Theta(B)\epsilon_t, \quad (1.34)$$

or

$$\Phi(B)(1 - B^s)(1 - B^d)X_t = \Theta(B)\epsilon_t. \quad (1.35)$$

Example 1.7.5. Monthly hotel room averages time series in Kuwait for 1985-1995(see 26), here $S = 12$:

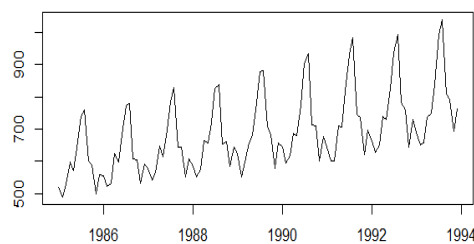


Figure 1.9: hotel data.train 1985 to 1995

1.8 Box and Jenkins methodology

In general finding models for time series is non trivial. The approach proposed by Box-Jenkins come to be known us Box-Jenkins methodology to ARIMA models, when this method became highly popular in 1970 among academics. It has tree part: identification, estimation, and verification(diagnositics).

Table schematic representation of the Box-Jenkins approach source Maenadic and M.Hibou 95/45/TM. Revised version 95/33/TM.

Table 1.4: Box-Jenkins approach

phase 1)		postulate general class of models
	→	
Identification		identify model to be tentatively entertained
phase 2)		estimate parameters tentatively entertained model
	→	
Estimation and testing		diagnostic checking is the model adequate no or yes
phase 3)	→	use model for forecast

1.8.1 Estimation

The first determining appropriate the order of the ARMA model by examining the autocorrelation and partial autocorrelation of the stationary series. And estimate the parameters of models and after select best model with many of criteria for example:

Information criteria

The goodness of fit of the model can be assessed with the residuals variance:

$$\hat{\sigma}^2(k) = \frac{1}{n} \sum_{t=1}^n \hat{\epsilon}_t^2 \quad (1.36)$$

where n is the number of observation used for estimate k is total number of parameters estimated (ex: $k = p + q + 2$), $\sum_{t=1}^n \hat{\epsilon}_t^2$ is the adjusted residuals at time t .

Akaike information criteria (AIC)

Définition 1.8.1. The Akaike information criteria (AIC) is defined to be :

$$AIC(p, q) = n \times \log(\hat{\epsilon}_k^2) + 2 \times (k) \quad (1.37)$$

where k is the total number of parameters estimated

Shwarz bayesian information criteria (BIC)

Définition 1.8.2. Schwarz bayesian information criteria is defined to be:

$$BIC(p, q) = n \times \log(\hat{\epsilon}_k^2) + k \times (1 + \log(n)) \quad (1.38)$$

where k is the total number of parameters estimated.

Hannan-Quin information criteria (HQIC)

Hannan-Quin information criteria is defined to be:

$$HQIC(p, q) = \log(\hat{\epsilon}_t^2) + \frac{2k}{n} \log(\log(n)). \quad (1.39)$$

where k is the total number of parameters estimated.

Remark 1.8.3. - The objective is thus to choose the number of parameters which minimizes the value of the information criteria.

- Diagnostic testing in the Box-Jenkins methodology involves the statistical properties of the errors terms (normality assumption, weak white noise assumption).

1.8.2 Forecasting in ARMA models

Now suppose we interested in forecasting X_{n+k} from observations $\{X_t, t \leq n\}$. Working with the Wold representation we may consider forecasts of from:

$$\hat{X}_{n,k} = \sum_{j=0}^{\infty} C_{k,j} \epsilon_{n-j} \quad (1.40)$$

Comparing 1.40 with the Wold representation for X_{n+k} , we introduce:

$$X_{n+k} - \hat{X}_{n,k} = \sum_{j=0}^{k-1} C_j \epsilon_{n+k-j} + \sum_{j=0}^{\infty} C_{j+k} (C_{j+k} - C_{k,j} C_{k,j}) \epsilon_{n-j}.$$

Hence

$$E(X_{n+k} - \hat{X}_{n,k})^2 = \{\sum_{j=0}^{k-1} C_j^2 + \sum_{j=0}^{\infty} (C_{j+k} - C_{k,j} C_{k,j})^2\} \sigma_\epsilon^2$$

This expression may be minimized by setting : $C_{j+k} = C_{k,j}$ for all $j \geq 0, k \geq 0$, and then give rise to then mean squared prediction error.

Application on ARIMA Model

Example 1.8.4. In this example, we show us the application of Box-Jenkins methodology to give ARIMA model for traffic accidents time series in France:

1) Visualize the Time Series

We visualize the times series by plotting it see figure 1.10:

2) Stationarity the Series

We can check if the series is stationary or not. Dickey Fuller is one of the popular test used.

Augmented Dickey-Fuller Test

data: traffic accidents.train

Dickey-Fuller = -2.2719, Lag order = 0, p-value = 0.4674

alternative hypothesis: stationary

From this test, p-value= 0.4674, then the time series is non-stationary (i.e. explosive), we need to stationeries the series (by taking difference transformation).

data: ddataaccidts.train

Dickey-Fuller = -5.3221, Lag order = 0, p-value = 0.01

alternative hypothesis: stationary

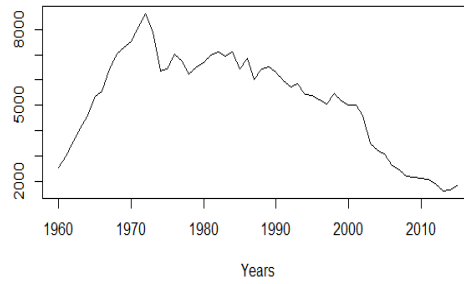


Figure 1.10: traffic accidents in France

Now the time series is stationary

3) Find Optimal Parameters of ARIMA Model

The value found in the previous section might be an approximate estimate and we need to explore more (p,d,q) combinations. The one with the lowest AIC should be our choice. `arima(x = ddata traffic accidents.train, order = c(0, 0, 1))` :

Coefficients: σ^2 estimated as 250332: log likelihood = -267.22, aic = 540.44

Table 1.5: parameters estimation

	ma1	intercept
	0.2032	83.1014
s.e.	0.1527	101.2803

4) Residuals diagnostic

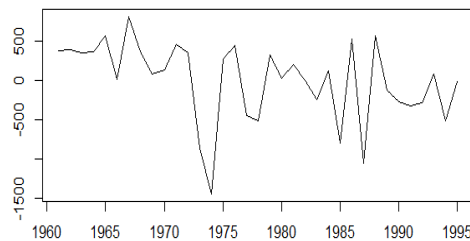


Figure 1.11: ddata of traffic accident .train residuals

Box-Ljung test

`Box.test(ddata traffic accidents.train arima residuals, lag = 25, type = "Ljung-Box")`
 data: raffic accidents.train arima residuals

X-squared = 14.078, df = 25, p-value = 0.9604

Jarque Bera test

data: raffic accidents.train arima residuals

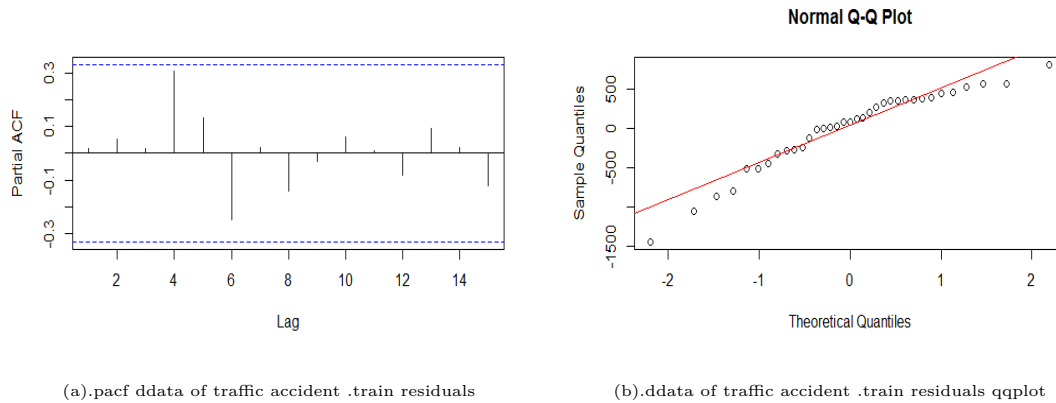
X-squared = 5.749, df = 2, p-value = 0.05644 5) **Make Predictions**

Figure 1.12: Residuals diagnostic

Once we have the final ARIMA model, we are now ready to make predictions on the future time points. We can also visualize the trends to cross-validate if the model works fine.

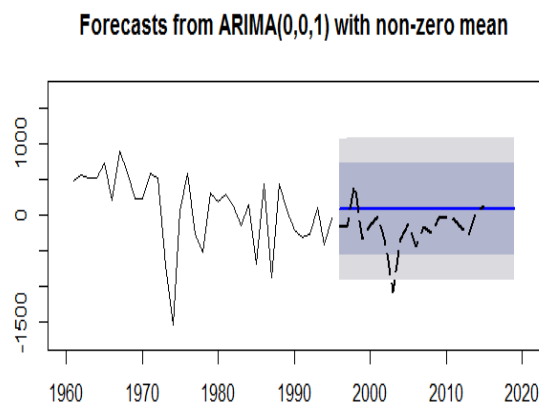


Figure 1.13: ddata traffic accident forecast

Application on SARIMA Model

In this example, we applique the Box-Jenkins methodology for giving SARIMA model to monthly hotel room averages time series in Kuwait for 1985-1995(see [26]):

1) Visualize the Time Series in training fase

See figure 1.9.

2) Stationarity the Series

Augmented Dickey-Fuller Test

data: hotel.data.train

Dickey-Fuller = -2.7543, Lag order = 15, p-value = 0.2637

alternative hypothesis: stationary

Augmented Dickey-Fuller Test data: diff(hotel datats.train, 12) Dickey-Fuller = -7.1603,

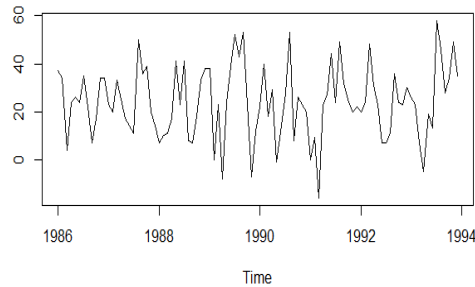


Figure 1.14: hotel ddata.train 1985 TO 1995

Lag order = 0, p-value = 0.01 alternative hypothesis: stationary

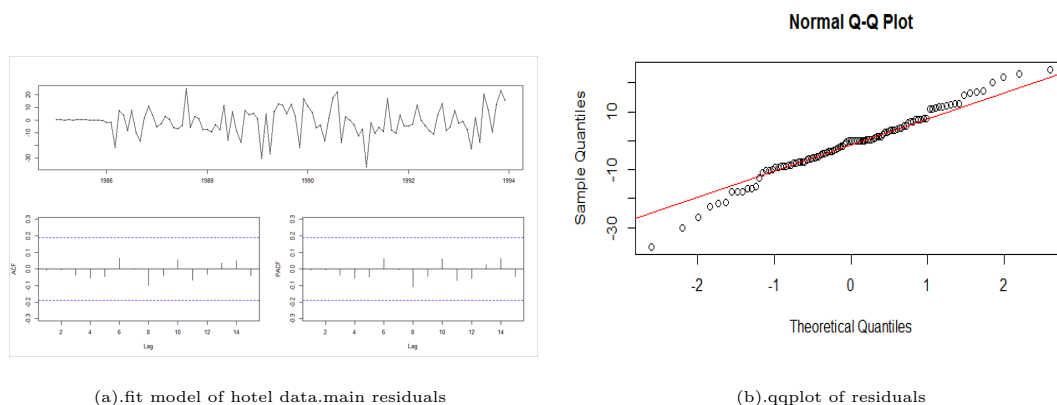
3) Find Optimal Parameters of Model

Series: data ts.train ARIMA(3,1,3)(1,1,1)[12] Coefficients:

ar1	ar2	ar3	ma1	ma2	ma3	sar1	sma1
0.4913	0.5668	-0.5573	-1.3164	-0.3255	0.6422	-0.1726	-0.2256
s.e. 0.1417	0.1444	0.0946	0.3108	0.2903	0.2799	0.2538	0.2443

σ^2 estimated as 148.7: log likelihood=-375.11 AIC=768.22 AICc=770.34 BIC=791.21

4) Residuals diagnostic



(a).fit model of hotel data.main residuals

(b).qqplot of residuals

Figure 1.15: Residuals diagnostic

Box-Ljung test

data: fit model residuals

X-squared = 4.3551, df = 15, p-value = 0.9963

Jarque Bera Test

data: residuals(fit model)

X-squared = 3.9052, df = 2, p-value = 0.1419 then the errors are uncorrelated.

5) Make Predictions

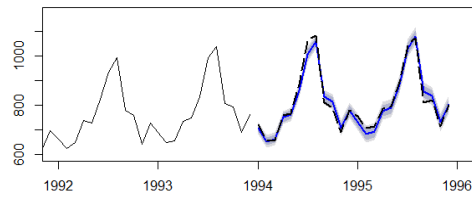


Figure 1.16: forecast fit model

The figure of the fit model on time series observations shows us, that the fit model is better for forecasting.

Chapter 2

Genetic Algorithms

Genetic algorithms (GAs) are numerical optimization algorithms inspired by both natural selection and natural genetics. They represent an intelligent exploitation of random search used to solve the optimization problem. This particular description of GAs is intentionally abstract because, in some sense, the term genetic algorithm has two meanings. So in this chapter, we will spell the principals of this method.

2.1 Definitions and terminology

2.1.1 Definitions

Définition 2.1.1. (Local maximum)

We said $x_0 \in \Omega$ is the local maximum of function f if $\forall x \in \Omega, \exists \epsilon > 0 : f(x_0) \geq f(x), \forall x \in R, |x - x_0| < \epsilon$.

Définition 2.1.2. (Local minimum)

We said $x_0 \in \Omega$ is the local minimum of function f if $\forall x \in \Omega, \exists \epsilon > 0 : f(x_0) \leq f(x), \forall x \in R, |x - x_0| < \epsilon$.

Définition 2.1.3. (Local optimum)

We said $x_0 \in \Omega$ is the local optimum of function f if it is either a local maximum or local minimum.

Définition 2.1.4. (Global maximum)

We said $x_0 \in \Omega$ is the global maximum of function f if $\forall x \in \Omega : f(x_0) \geq f(x), \forall x \in R$.

Définition 2.1.5. (Global minimum)

We said $x_0 \in \Omega$ is the global minimum of function f if $\forall x \in \Omega : f(x_0) \leq f(x), \forall x \in R$.

Définition 2.1.6. (Global optimum)

We said $x_0 \in \Omega$ is the global optimum of function f if it is either a global maximum or global minimum.

Example 2.1.7. Figure of function has global minimum and local minimum

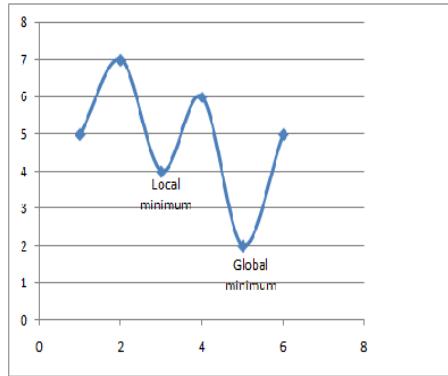


Figure 2.1: global minimum and local minimum

Définition 2.1.8. (Solution candidate)

A solution candidate is an element of problem space Ω .

Définition 2.1.9. (Solution space)

We call the union of all solutions of optimization problem it's solution space S , $S \subseteq \Omega$.

Remark 2.1.10. The solution space contain and can be equal to the globally optimal set χ^* , $\chi^* \subseteq S \subseteq \Omega$.

Définition 2.1.11. (Genotype)

We call an element in Ω , a genotype.

Définition 2.1.12. (Gene)

The distinguishable unit of information in genotype that encode the phenotypical properties are called a gene.

Définition 2.1.13. (Allele)

An allele is a value of specific gene.

Définition 2.1.14. (Phenotype)

The phenotype is the population in the actual real world solution space in which solutions are represented in a way they are represented in real world situations.

Définition 2.1.15. (Individual)

Each element of set Ω is called an individual presented by some chromosomes(genomes).

Définition 2.1.16. (Population)

A population Pop is a list of individuals used during an optimization process.

Définition 2.1.17. (Generation)

Generation is a set of population at moment donned of process.

2.1.2 Objectif function and fitness function

Définition 2.1.18. Let Ω is problem space and an objective function $f : X \rightarrow Y$ with $Y \subseteq R$ is a mathematical function which is subject to optimization, where X is the set of possible solutions. We can call f the target or score function.

Example 2.1.19. let $X = [-1, 1]$ and $Y = R$

$$f : X \rightarrow Y: f(x) = (x - 1) \exp(2x - 1) + 1$$

Définition 2.1.20. The fitness function is normally used to transform the objective function values into a measure of relative fitness, thus $F(x) = g(f(x))$, where f is the objective function, g transform the values of the objective function to a non-negative number and F is relative fitness.

Example 2.1.21. $F(x_i) = \frac{f(x_i)}{\sum_{i=1}^n f(x_i)}$ where n is the population size and x_i is the phenotype value of individual i .

2.2 Encoded

A population of individuals is maintained within Ω , each representing a possible solution to a given problem. Each individual is coded as a finite length vector or variable in terms of some alphabet.

Définition 2.2.1. Assume S to be a set of strings, let Ω is the search space of an optimization problem, then a function $C : \Omega \rightarrow S$ $x \mapsto C(x)$ is called coding function. Conversely, a function $\tilde{C} : S \rightarrow \Omega$ $s \mapsto \tilde{C}(s)$ is called decoding function.

Remark 2.2.2. Moreover, the following equality is often supposed to be satisfied : $C \circ \tilde{C} \equiv I_S$.

There are many kinds of individuals encoding, We introduce three popular encoding forms:

- Binary coded;
- Gray coded;
- Real coded.

2.2.1 Binary-coded

Binary encoding is the most common form of encoding. In this encoding, each chromosome is represented using a binary string. Each bit in the string can represent some characteristics of the solution.

Définition 2.2.3. Binary encoding represents the chromosomes by a string of bits $\{0,1\}$.

Remark 2.2.4. Holland and his students concentrated on such encodings, and GA practice has tended to follow this lead.

Precession of each element of k is determined by a string of length l_k and the desired resolution r_k . In general $r_k = \frac{UB-LB}{2^{l_k}-1}$ where UB and LB specify the upper and lower limits of the range of parameters.

Example 2.2.5. Encoding and Decoding principle

Encoding	\leftrightarrow	Decoding
genotype	\leftrightarrow	phenotype
coded domain	\leftrightarrow	decision domain
biological UGCAACCGU	$\xrightarrow{\text{expression}}$ and $\xleftarrow{\text{sequencing}}$	"blue eye"
10011110	$\xrightarrow{\text{decoding}}$ and $\xleftarrow{\text{encoding}}$	158

Remark 2.2.6. Binary encoding has problem with Hamming cliff distance. The distance between two genotypes x_g and y_g is defined by Hamming as $d(x_g, y_g) = \sum_{i=0}^{l-1} |x_{g,i} - y_{g,i}|$, and denote the number of different alleles in the two genotypes.

2.2.2 Gray-coded

The Gray encoded can be constructed in two step:

- 1) The phenotype is encoded using binary encoding,
- 2) Subsequently the binary encoded string can be converted into the corresponding Gray encoded string.

If the binary string $x \in \{0, 1\}^l = \{x_1, x_2, \dots, x_l\}$, then is converted to the corresponding Gray code $y \in \{0, 1\}^l = \{y_1, y_2, \dots, y_l\}$ by mapping $\gamma : \mathbf{B}^l \rightarrow \mathbf{B}^l$

$$y_i = \begin{cases} x_i & \text{if } i = 1 \\ x_{i-1} \oplus x_i & \text{other wise} \end{cases}$$

Where \oplus denotes addition modulo 2. The decoding of Gray encoded string is as follow : $x_i = \oplus_{j=1}^i x_{i-1}$, for $i \in \{2, \dots, l\}$.

A Gray encoded string has the same length l as a binary encoded string, and the encoding is redundancies-free.

Example 2.2.7. An example for using binary and Gray encoding :

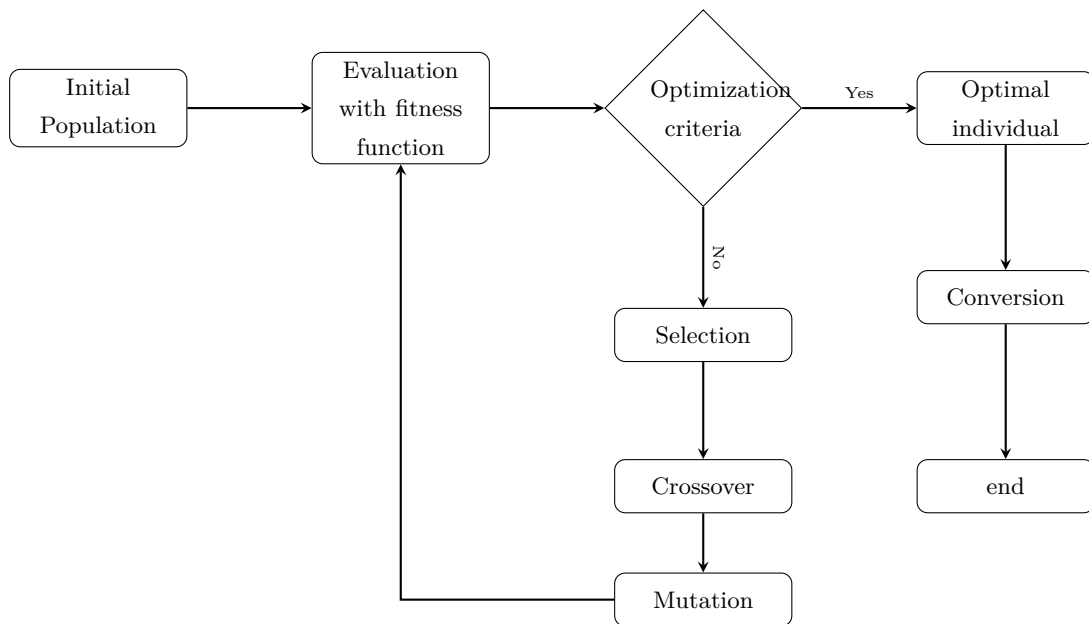
x_g	0	1	2	3	4	5	6	7
binary	000	001	010	011	100	101	110	111
Gray	000	001	011	010	110	111	101	100

2.2.3 Real-coded RCGA

In the RCGA a chromosome is coded as a finite length string of the real numbers corresponding to the design variable. We develop this section in the fourth chapter.

2.3 Genetic algorithms process

Graph of the GAs schematic process.



2.3.1 Initialization

Initially, many individuals solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contain several hundred or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (space Ω). Occasionally, the solution may be seeded in an area where optimal solutions are likely to be found.

Example 2.3.1. Generate random initial population contained between lower and upper bounds in solutions space, for example, $I = [0, 14]$ and the probabilities for having the gene 0 or 1 is $p = 0.5$

We generate six individuals(chromosomes) with four genes:

Random numbers	Binary coded	Random numbers	Binary coded
0.263	0	0.509	1
0.684	1	0.648	1
0.489	0	0.656	1
0.010	0	0.123	0
0.351	0	0.745	1
0.779	1	0.310	0
0.578	1	0.407	0
0.292	0	0.245	0
0.070	0	0.876	1
0.331	0	0.719	1
0.980	1	0.426	0
0.672	1	0.720	1

The first generation or initial population is:

Encoding individuals	Decoding individuals
$x_1 = 0100$	4
$x_2 = 0110$	6
$x_3 = 0011$	3
$x_4 = 1110$	14
$x_5 = 1000$	8
$x_6 = 1101$	13

2.3.2 Selection

Définition 2.3.2. Selection is the process of determining the number of time, or trial, a particular individual is chosen for reproduction and, thus the number of spring an individual will produce.

Définition 2.3.3. (Fitness distribution of population) The fitness $S : R \rightarrow Z^+$, assigns to each fitness value $f \in R$ the number of individuals in population pop carrying this fitness value. S is called the fitness distribution of population pop .

Définition 2.3.4. A selection method G is a function that transform a fitness distribution S into a new fitness distribution S'
 $S' = G(S)$.

Probability for Selection

The probability of choosing a certain individual is proportional to his fitness. It can be regarded as a random experiment with:

$$P(x_i) = \frac{f(x_i)}{\sum_{i=1}^m f(x_i)}$$

This formula makes sense if all fitness values are positive. If this not the case, a non-decreasing transformation $g : R \rightarrow R^+$ must be applied. Then the probability can be expressed as:

$$P(x_i) = \frac{g(f(x_i))}{\sum_{i=1}^m g(f(x_i))}$$

Remark 2.3.5. A linear transformation which offsets the objective function, such that:
 $F(x) = af(x) + b$

Where a is a positive scaling factor if the optimization is maximizing and negative if we are minimizing. And b is used to ensure that the resulting fitness values are non-negative.

Remark 2.3.6. It possible to describe a selection method as a function transform a fitness distribution into another fitness distribution.

Définition 2.3.7. (Expected fitness distribution) G^* denotes the expected fitness distribution after applying selection method G to the fitness distribution S i.e :
 $S^* = G^*(S) = E(G(S))$.

Théorème 2.3.8. The variance in obtaining the fitness distribution S' is :

$$\sigma_S^2 = S^*(1 - \frac{S^*}{N})$$

Proof. See [11] □

Remark 2.3.9. The index S in σ_S stand for "sampling" as it is the mean variance due to the sampling of the finite population.

Cumulative fitness distribution

1) For discreet fitness distribution:

Définition 2.3.10. We denote $\Gamma(f_i)$ the cumulative fitness distribution, defined by:

$$\Gamma(f_i) = \begin{cases} 0 & i < 1; \\ \sum_{j=1}^i S(f_j) & 1 \leq i \leq n; \\ N & i > n \end{cases} \quad (2.1)$$

Example 2.3.11. plot of cdf of discrete distribution,

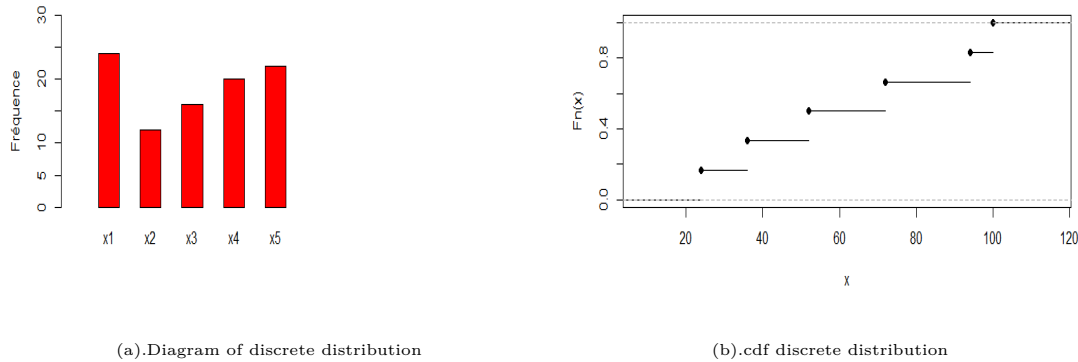


Figure 2.2: Diagram cdf of discrete distribution

2) For continuous fitness distribution:

Définition 2.3.12. Let us $g(f)$ is continuous fitness distribution. We denote $\Gamma(f)$ the cumulative fitness distribution defined by:

$$\Gamma(f) = \int_{f_0}^{f_n} S(f)df.$$

Example 2.3.13. Gaussian distribution $F(\mu, \sigma)$ with :

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

Définition 2.3.14. (Average fitness) We denote \bar{M} , the average fitness of population before selection, and \bar{M}^* is the expected average fitness after selection, for continuous distribution:

$$\begin{cases} \bar{M} = \frac{1}{N} \int_{f_0}^{f_n} S(f)fd f \\ \bar{M}^* = \frac{1}{N} \int_{f_0}^{f_n} S^*(f)fd f \end{cases} \quad (2.2)$$

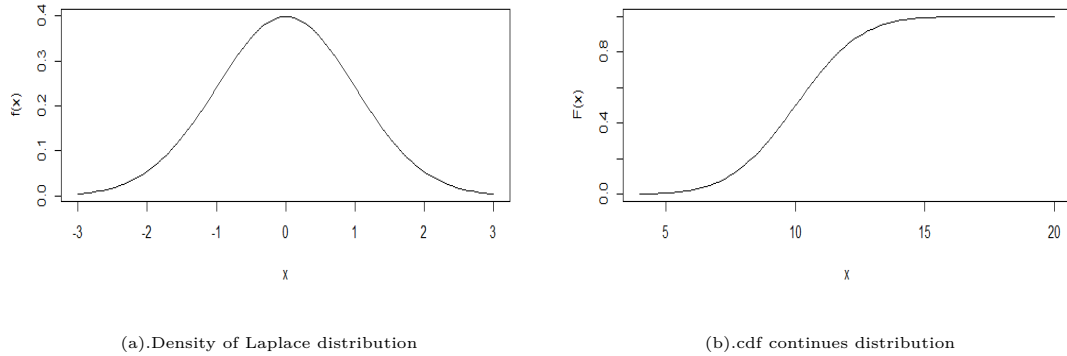


Figure 2.3: Diagram cdf of continues distribution

Définition 2.3.15. (Fitness variance) The fitness variance $\bar{\sigma}^2$ denotes the variance of the fitness distribution $S(f)$ before selection, and $(\bar{\sigma}^*)^2$ denotes the variance of the fitness distribution $S^*(f)$ after selection:

$$\begin{cases} \bar{\sigma}^2 = \frac{1}{N} \int_{f_0}^{f_n} S(f)(f - \bar{M})^2 df \\ (\bar{\sigma}^*)^2 = \frac{1}{N} \int_{f_0}^{f_n} S^*(f)(f - \bar{M}^*)^2 df \end{cases} \quad (2.3)$$

Définition 2.3.16. (Reproduction rate) The reproduction rate $\bar{R}(f)$ denote de ratio of the number of individuals with a certain fitness f before and after selection:

$$\bar{R}(f) = \begin{cases} \frac{S^*(f)}{S(f)} & S(f) > 0; \\ 0 & S(f) = 0; \end{cases} \quad (2.4)$$

Remark 2.3.17. A reasonable selection method should favor good individuals by assigning them a reproduction rate $\bar{R}(f) > 1$, and punish bad individuals by a ratio $\bar{R}(f) < 1$

During every selection phase, bad individuals will be lost and replaced by copies of better individuals.

Définition 2.3.18. (Loss of diversity) The loss of diversity P_d is the proportion of individuals of a population that is not selected during the selection phase.

Théorème 2.3.19. *If the reproduction rate $\bar{R}(f)$ increases monotonously in f , the class diversity of selection method is :*

$$P_d = \frac{1}{N}(\bar{\Gamma}(f_z) - \bar{\Gamma}^*(f_z)). \quad (2.5)$$

where f_z denotes the fitness value, such that $\bar{R}(f) = 1$

Proof. For all fitness values $f \in [f_0, f_z]$, the reproduction rate is less than one. Hence the number of individuals are not selected during selection is given by $\int_{f_0}^{f_z} (S - S^*)dx$.

It follows that :

$$\begin{aligned} P_d &= \frac{1}{N} \left(\int_{f_0}^{f_z} (S(x) - S^*(x))dx \right). \\ &= \frac{1}{N} \left(\int_{f_0}^{f_z} S(x)dx - \frac{1}{N} \int_{f_0}^{f_z} (S^*(x))dx \right). \\ &= \frac{1}{N} (\Gamma(f_z) - \Gamma^*(f_z)) \end{aligned}$$

□

Remark 2.3.20. - The loss of diversity should be as low as possible because a high loss of diversity increases the risk of premature convergence. - Beker [1989], has introduced a similar measure called "reproduction rate RR". RR gives the percentage of individuals that is selected to reproduce, hence $RR = 100(1 - P_d)$.

Certain selection methods rate the fitness of each solution and preferentially select the best solutions. The most commonly used methods of selecting chromosomes for parents to crossover:

1. Roulette wheel selection
2. Elitism selection
3. Rank selection
4. Tournament selection
5. Truncation selection
6. Boltzmann selection

Roulette Wheel selection RWS

RWS is the simplest selection approach. In this method, all the chromosomes (individuals) in the population, are placed on the (RWS) according to their fitness value. Each individual is assigned a segment of roulette. The size of each segment in the (RWS), is proportional to the value of the fitness of the individual. The virtual (RWS) is spined and the individual corresponding to the segment on the witch (RWS) stops are then selected. The process is repeated until the desired number of individuals is selected.

Example 2.3.21. We take example of five individuals selection:

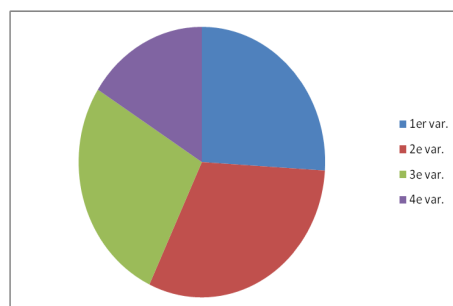


Figure 2.4: Roulette Wheel

Elitism selection ES

The idea of elitism has been already introduced. When creating new population by crossover and mutation, we have a big chance, that we will loose the best chromosome. Elitism is the name of the method, which first copies the best chromosome (or a few best chromosomes) to a new population. The rest is done in a classical way. Elitism can very rapidly increase the performance of GA, because it prevents losing the best-found solution.

Rank Selection

Rank selection ranks the population and every chromosome receives fitness from the ranking. Rank selection first ranks the population and then every chromosome receives fitness from this ranking. The worst will have fitness 1, second worst 2 etc. and the best will have fitness N (number of chromosomes in population).

The previous selection will have problems when the fitness differs very much. For example, if the best chromosome fitness is 90 percent of all the roulette wheel then the other chromosomes will have very few chances to be selected.

linear ranking selection LRS

The selection probability is assigned to the individual according to their rank:

$$P_i = \frac{1}{N}(\eta^- + (\eta^+ - \eta^-) \frac{i-1}{N-1}), i \in \{1, 2, \dots, N\}. \quad (2.6)$$

Here $\frac{\eta^-}{N}$ is the probability of the worst individual to be selected, and $\frac{\eta^+}{N}$ is the probability of the best individual to be selected.

Remark 2.3.22. As the population size is held constant, the condition $\eta^+ = 2 - \eta^-$ and $\eta^- \geq 0$ must be full filled. Not that all individuals get a different rank, i.e a different selection probability, even if the individuals have the same fitness value.

Exponential ranking selection ERS

Exponential ranking selection probability assigned to the individual is:

$$P_i = \frac{C^{N-i}}{\sum_{j=1}^N C^{N-j}}, i \in \{1, 2, \dots, N\}. \quad (2.7)$$

When $0 < C < 1$. The sum $\sum_{j=1}^N C^{N-j}$ normalizes the probability to ensure that $\sum_{j=1}^N P_i = 1$.

As $\sum_{j=1}^N C^{N-j} = \frac{C^N - 1}{C - 1}$, we can rewrite the above equation by:

$$P_i = \frac{(C-1)C^{N-i}}{C^N - 1}, i \in \{1, 2, \dots, N\}. \quad (2.8)$$

Example 2.3.23. You can see in the following picture, how the situation changes after changing fitness to order number.

Remark 2.3.24. After this, all the chromosomes have a chance to be selected. But this method can lead to slower convergence because the best chromosomes do not differ so much from other ones.

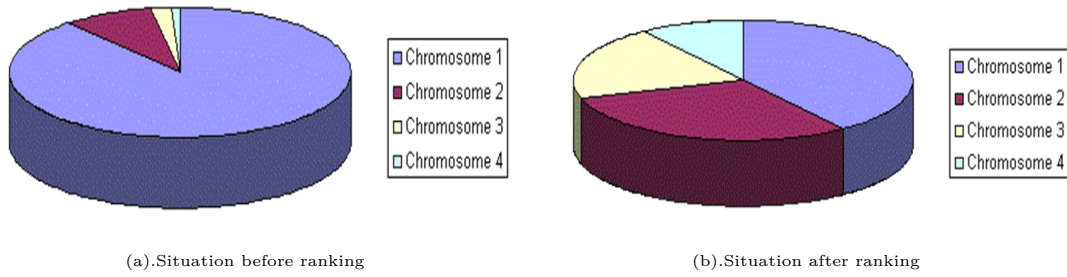


Figure 2.5: Rank Selection

Tournament selection TS

Tournament selection is a variant of the rank-band selection method. Its principle consists in randomly selecting a set of N individuals. His strategy provides selective pressure by holding a competition among N individuals from the population $Pop(i)$. The best individual is the highest fitness and copy the best one from this group into the intermediate population $Pop(i + 1)$ and repeat k times. For unconstrained optimization problems:

(i) Among two selected chromosomes, the chromosome with better fitness value is selected.

For constrained optimization problems:

(i) When two chromosomes (individuals) are feasible then the individual with better fitness value is selected.

(ii) When one chromosome (individual) is feasible and another is infeasible then the feasible one is selected.

(iii) When two chromosomes (individuals) are infeasible with unequal constraints violation, then the chromosome with less constraint violation is selected.

(iv) When two chromosomes (individuals) are infeasible with equal constraints violation, then anyone chromosome (individual) is selected.

Truncation selection TrS

This method is very simple technique; that orders the candidate solutions of each population according to their fitness. Then, only a certain portion of fittest individuals are selected and they all have the same selection probability.

Remark 2.3.25. This method is less used in practice, then other techniques except for very large population.

Boltzman selection BS

This method simulates the process of slow cooling of molten method to achieve the minimum function value in minimization problem. Let f_{max} be the fitness of the currently available best string. If the next string has fitness $f(x_i)$ such that $f(x_i) > f_{max}$, then the new string is selected. Other wise it is selected with Boltzmann probability :

$$P = \exp\left(\frac{-(f_{max} - f(x_i))}{T}\right) \quad (2.9)$$

Where $T = T_0(1 - \alpha)^k$ and $k = (1 + 100\frac{g}{G})$, g is the current generation number, and G the maximum value of g . The value of α can be chosen from the rang $[0,1]$ and to from

the [5,100]. The final state is reached when computation approaches zero value of T i.e, the global solution is achieved at this point.

Remark 2.3.26. The selection operator is intended to improve the average quality of the population by giving individuals of higher quality a higher probability to be copied into next generation.

2.3.3 Crossover

Crossover is a genetic operator that two chromosomes parents produce a new chromosomes children. Then the new chromosomes may be better than both of the parents. After this operation, the population enriched with better(off string) individuals and the evolution process may be continued.

The pairs of individuals selected undergoing crossover with probability P_c . A random number R_c is generated in the range $0 - 1$, and the individuals undergo crossover if and only if $R_c \leq P_c$. Otherwise the pair proceed without crossover. Typical values of P_c are 0.4 to 0.9. Without a crossover, the average fitness of population f_{ave} , will climb until it equals the fitness of the best member, f_{max} . After this point, it can only improve via mutation.

Remark 2.3.27. If $P_c = 0.5$ then half the new population will be formed by selection and crossover, and half by selection alone.

Many different crossover algorithms have been devised:

- One-point crossover
- Two-point crossover
- Cut and splice
- Uniform and half-uniform crossover
- Arithmetic crossover
- Heuristic crossover

Single-point crossover SPC

Single-point crossover proceeds by cutting the pair of selected strings at a random locus (picked by throwing a random number, R_L , between 1 and $L - 1$, where L is number of genes) and swapping the tails to create two child strings.

Example 2.3.28. For $R_L = 4$ $L = 12$ and then

Two-point crossover TPC

This operator randomly selects two crossover points within a chromosome then, interchanges the two parents chromosomes between these points to produce two new offspring.

Example 2.3.29. For $R_L = 4$, then

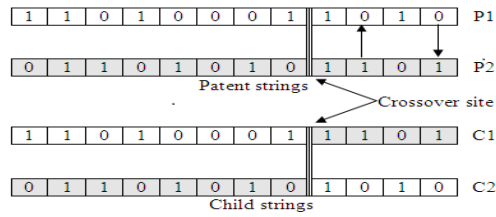


Figure 2.6: Crossover in one point

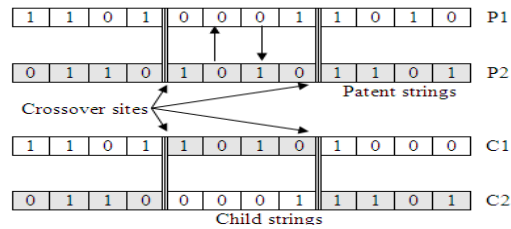


Figure 2.7: Crossover in two points

2.3.4 Cut and splice crossover

Like one-point crossover, except each parent has a different cut point. Can result in variable length children.

Example 2.3.30. Different cut point crossover:

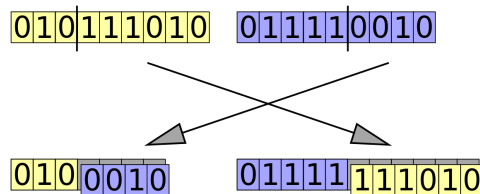


Figure 2.8: Cut and splice crossover

Uniform crossover and half Uniform crossover

Single and multi-points crossover define cross points as places between loci where a chromosome can be split. Uniform crossover generalizes this scheme to make every locus a potential crossover point.

- The individual genes are compared between two parents.
- The gene value are swapped with a fixed probability typically 0.5
- Half uniform crossover scheme is exactly of the non-matching genes are swapped.

Example 2.3.31. Consider the following two parents crossover made and resulting offspring :

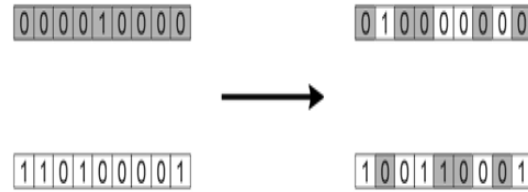


Figure 2.9: Uniform crossover

$$P_1 = 000010000$$

$$P_2 = 110100001$$

$$C_1 = 010000000$$

$$C_2 = 100110001$$

Here the first child1 is produced by taking bit from P_1 if the corresponding mask bit is 1 or the bit from P_2 if the corresponding mask bit is 0.

Offspring child is created using the inverse of mask, or equivalently, swapping P_1 and P_2 .

2.3.5 Heuristic crossover HC

This crossover operator uses the fitness values of the two parents chromosomes to determine the direction of the search. The offspring are created according to the equation:

$$offspring1 = bestparent + r(bestparent - worstparent)$$

$$offspring2 = bestparent$$

Where r is a random number between 0 and 1.

Remark 2.3.32. It possible that offspring1 will not be feasible. It happens if r is chosen such that one, or more of its genes fall outside of the allowable upper or lower bounds. For this reason, the heuristic crossover has a user defined parameter n for the number of time to try and r that in a feasible chromosome. If a feasible chromosome is not produced after n tries, the worst parent is returned as offspring1.

2.3.6 Mutation

In the natural world, several processes can cause mutation, the simplest being an error during replication. After crossover, the strings are subject to mutation.

Mutation plays the role of recovering the lost genetic materials as well as for randomly distribute genetic information. And introduce the new genetic structures in the population by randomly modifying the genes of chromosomes selected with a mutation probability P_m

Example 2.3.33. An example of a mutation in the binary string:

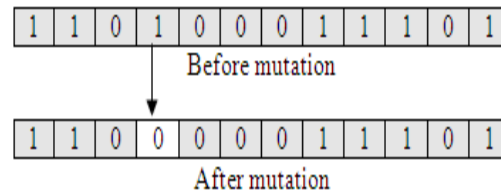


Figure 2.10: Mutation example

2.3.7 Types of Mutation

There are many different forms of mutation for different kinds of representation, in this chapter, we introduce various types of mutation operators in the binary string:

2.3.8 Insertion mutation

It is used in permutation encoding. First of all, pick two alleles values at random. Then move the second allele to follow the first shifting the rest along to accommodate.

Example 2.3.34. In insertion mutation we show 5 follow 2:

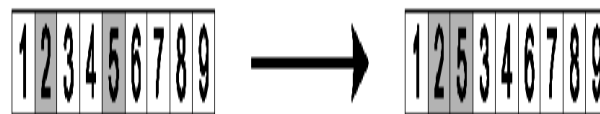


Figure 2.11: Insertion mutation

2.3.9 Inversion mutation IM

Inverse mutation is used for chromosomes with permutation encoding. In order to perform inversion, pick two alleles at random and then invert the substring between them. It preserves most adjacency information and only breaks two links but it lead to the discerption and only breaks order information.

Example 2.3.35. Inversion mutation plot for offspring:

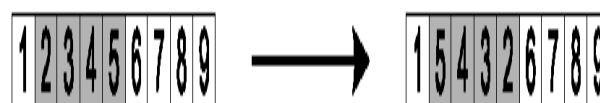


Figure 2.12: Inversion mutation

2.3.10 Scramble mutation ScM

Is also used with permutation encoded chromosome. In this mutation one has to pick a subset of genes at random and then randomly rearrange the alleles in those positions. Subset does not have to be contiguous.

Example 2.3.36. Scramble mutation plot for offspring:

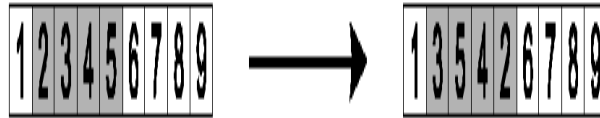


Figure 2.13: Scramble mutation

2.3.11 Swap mutation SwM

Is also used with permutation encoded chromosome. To perform swap, mutation select two alleles at random and swap their positions. It preserves most of the adjacency information but links broken disrupts order more.

Example 2.3.37. In swap mutation, we show that exchange between 2 and 6:

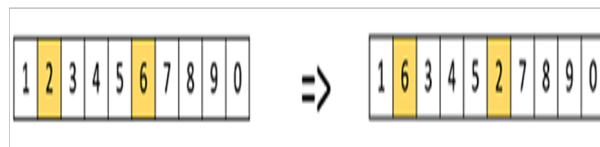


Figure 2.14: Swap mutation

2.3.12 Flip mutation FIM

Based on a generated mutation chromosome flipping of bit involve changing 0 to 1 and 1 to 0. A parent is considered and a mutation chromosome is randomly generated. It is commonly used in binary encoding.

Example 2.3.38. the principle of this mutation is popular in this example changing 1 to 0:

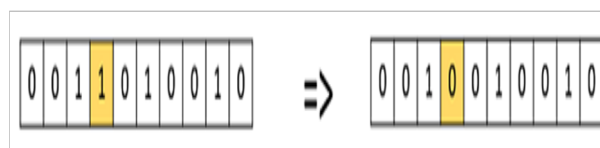
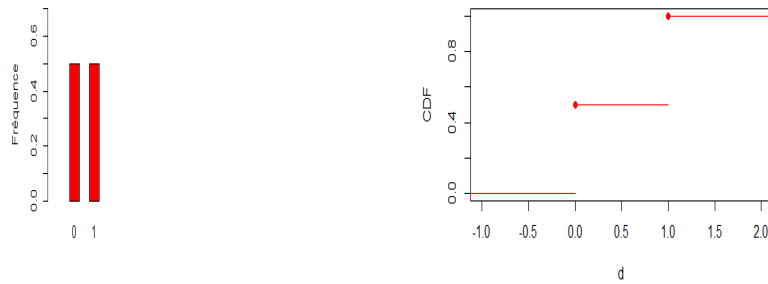


Figure 2.15: Bit flip mutation

2.4 Example of application

In this example we search the minimum of function f , $f(x) = 14x - x^2$ in solutions space for example $I = [0, 14]$

The first we generate a random initial population of 6 individuals with 4 bits contained between lower and upper bounds of I . The probabilities for have the gene 0 or 1 is $p = 0.5$ and probability density function (*pdf*) diagram And cumulative density function (*cdf*) diagram is:



(a).probability of bits

(b).cdf of bits

Random numbers	Binary coded	Random numbers	Binary coded
0.963	1	0.509	1
0.384	0	0.148	0
0.489	0	0.656	1
0.010	0	0.123	0
0.351	0	0.745	1
0.279	0	0.310	0
0.778	1	0.407	0
0.292	0	0.845	1
0.570	1	0.876	1
0.331	0	0.719	1
0.980	1	0.426	0
0.672	1	0.720	1

Then the first generation or initial population is:

Encoding individuals	Decoding individuals
$x_1 = 1000$	8
$x_2 = 0010$	2
$x_3 = 1011$	11
$x_4 = 1010$	10
$x_5 = 1001$	9
$x_6 = 1101$	13

The fitness function evaluation :

Encoding individuals	Decoding individuals	Fitness values	proportional	probability
$x_1 = 1000$	8	48	$\frac{48}{203}$	0.24
$x_2 = 0010$	2	24	$\frac{24}{203}$	0.12
$x_3 = 1011$	11	33	$\frac{33}{203}$	0.16
$x_4 = 1010$	10	40	$\frac{40}{203}$	0.20
$x_5 = 1001$	9	45	$\frac{45}{203}$	0.22
$x_6 = 1101$	13	13	$\frac{13}{203}$	0.06

Roulette wheel selection, it can be:

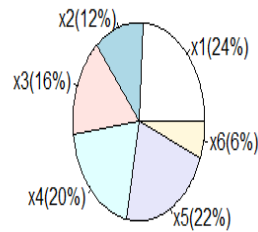


Figure 2.16: Roulettewoolforexampple

individuals	Lower bound	Upper bound
x_1	0	0.24
x_2	0.24	0.36
x_3	0.36	0.52
x_4	0.52	0.72
x_5	0.72	0.94
x_6	0.94	1

The probability for crossover is $p_c = 0.70$ and the diagram of d_{cf} is:

Selection of candidates parents for crossover:

$$0.153 \rightarrow x_1$$

$$0.26 \rightarrow x_2$$

$$0.647 \rightarrow x_4$$

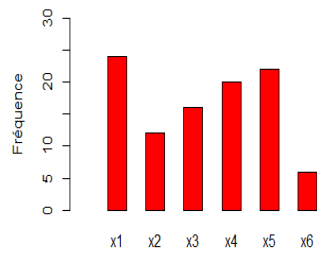


Figure 2.17: Diagram of parents

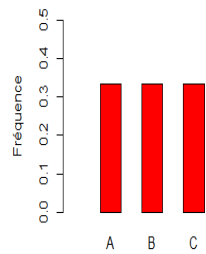


Figure 2.18: crossover point probability

$$0.037 \rightarrow x_1$$

$$0.341 \rightarrow x_2$$

$$0.512 \rightarrow x_3$$

we take three random number :

$$0.995 \rightarrow x_1, x_2 \text{ no crossover}$$

$$0.071 \rightarrow x_4, x_1 \text{ yes crossover}$$

$$0.566 \rightarrow x_2, x_3 \text{ yes crossover}$$

Crossover operation:

We take a random number for determinate the point of crossover; thus the number is 0.62, then the point of crossover is B:

$$x'_1 = 1000$$

$$x'_2 = 0010$$

$$x'_4 = 1010$$

$$x'_1 = 1000$$

$$x'_2 = 1010$$

$$x'_3 = 0011$$

New generation:

individuals	Binary string	decoded string
x_1	1000	8
x_2	0010	2
x_3	1010	10
x_4	1000	8
x_5	1010	10
x_6	0011	3

Mutation:

The probability of mutation is $p_m = 0.001$. We generate 6 numbers randomly to show how

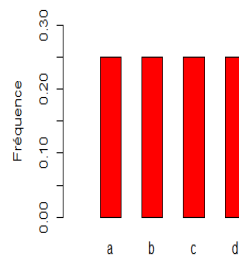


Figure 2.19: mutation probability

chromosome is under mutation:

0.225
0.368
0.256
0.237
0.416
0.088

Then the chromosome under mutation is x_6 by changing 0 to 1, but a 0 bit or gene to change in x_6 . For this, we generate a number randomly:

such this number is 0.538, then the bit is B, then:

individuals	Binary string	decoded string
x_6 before mutation	0011	3
x_6 after mutation	0111	7

If we have not the optimal solution, we start again the process with a new generation.

Chapter 3

Modeling linear Time Series with Binary Genetic Algorithm BCGA

In this chapter, we will apply the BCGA on linear time series models $AR(p)$, $MA(q)$ and $ARIMA(p, d; q)$ in application study, by minimizing the mean MSE square errors and Akaike information criteria AIC .

3.1 BCGA on Autoregressive models $AR(P)$

In the first step, we begin by application the BCGA to give $AR(P)$ model for Derive oil production time series in Algeria(1980 to 2011):

1) Visualize the Time Series

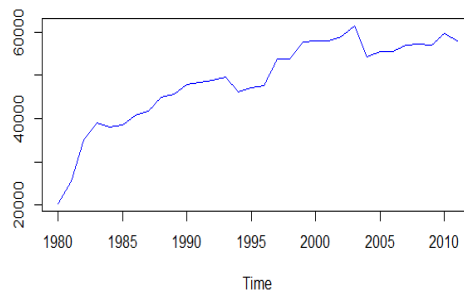


Figure 3.1: PRODUCTION DERIVE ENERGY FOR ALGERIA 1980 TO 2011

2) Stationarity the Series

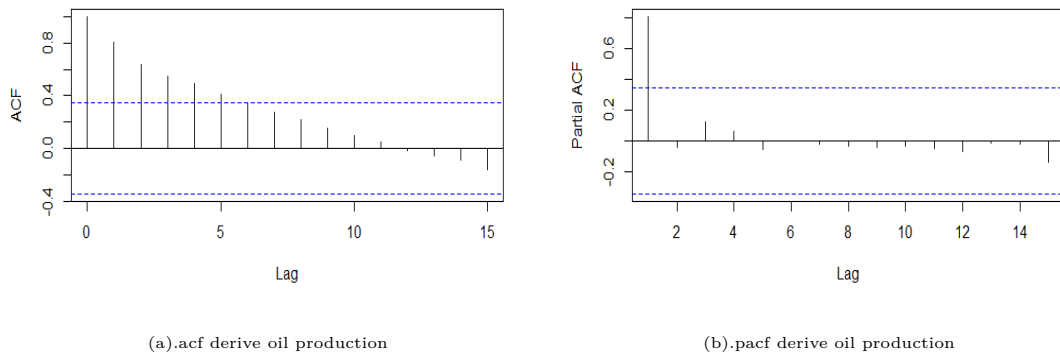


Figure 3.2: acf and pacf of derive oil production time series

Test de stationarity

Augmented Dickey-Fuller Test

data: product oil ts

Dickey-Fuller = -3.6693, Lag order = 1, p-value = 0.0431

alternative hypothesis: stationary

It's clear the Augmented Dickey-Fuller Test p-value = 0.0431 less then 0.05, indicate that time series is stationary.

3)Find Optimal Parameters of ARIMA Model with Box-Jenkins

The acf or pacf of this data indicated that one of $AR(1)$ or $AR(2)$, is the model of this time series. We check which one is optimal. `arima(x = product oil ts, order = c(1, 0, 0), include.mean = TRUE)`

Coefficients:

σ^2 estimated as 9501117: log likelihood = -304.18, aic = 614.35

Table 3.1: parameters estimation

ar1	intercept
0.9832	41176.46
s.e. 0.0218	15176.63

`arima(x = product oil ts, order = c(2, 0, 0), include.mean = TRUE)`

Coefficients:

σ^2 estimated as 8975099: log likelihood = -303.29, aic = 614.58.

Table 3.2: parameters estimation

ar1	ar2	intercept
1.2241	-0.2468	41553.61
s.e. 0.1784	0.1826	13761.45

Table 3.3: Box and Jenkins methods

MODEL	MSE	AIC	BAIC
MOD1: AR(1)	47604.08	614.35	618.7512
MOD2: AR(2)	47879.35	614.58	620.4407

This result on two tables show us the first model $AR(1)$ is better then $AR(2)$ because all statistics MSE , AIC and BIC of $AR(1)$ are less then of $AR(2)$.

Now we diagnostic the residuals of ARIMA(1, 0, 0)

4) Residuals diagnostic

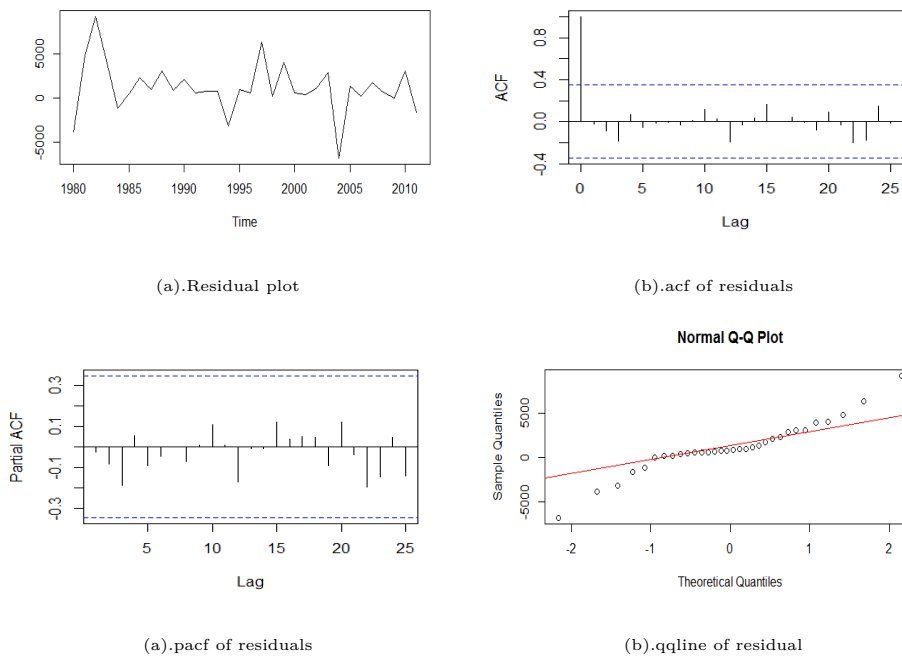


Figure 3.3: Residuals diagnostic

Box-Ljung test

data: arma product oil residuals

$X - squared = 6.4793, df = 15, p - value = 0.9705$

ThBox-Ljung test

data: arma product oil residuals

$X - squared = 6.4793, df = 15, p - value = 0.9705$

Then after this diagnostic, we can confirm that ARIMA(1, 0, 0) is fit model for data with Box-Jenkins

BCGA application

We applique BCGA for modeling the time series by choosing the fitness functions the AIC to optimize it. And follow all BCGA steps.

Selection

For example when we applique the binary genetic algorithms, on individuals in form $(\beta_i, c) \in [-1, 1]^2$, and use roulette wheel selection then:

Table 3.4: Fitness values of individuals

individuals	binary strings	$f(\beta_i)$	$F(\beta_i)$	$F(\beta_i) * 100$	rank
$\beta_1 = 0.9140625, c = 0.7890625$	(0.1110101,01100101)	$3.2264e^{+5}$	0.0028150	0.28150	4
$\beta_2 = 0.6796875, c = 0.4453125$	(0.1010111,00111001)	$9.2731e + 006$	0.0809074	8.09074	3
$\beta_3 = 0.171875, c = 0.6953125$	(0.0010110,01011001)	$7.0765e + 007$	0.6174216	61.74216	1
$\beta_4 = 0.4140625, c = 0.2890625$	(0.0110101,00100101)	$3.4253e + 007$	0.2988560	29.88560	2

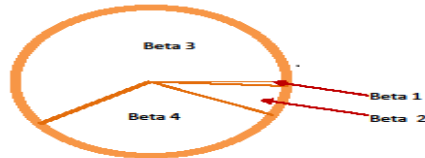


Figure 3.4: roulette wheel selection

Crossover

The crossover operator choused in this example is a single point Crossover with probability $P_C = 0.80$.

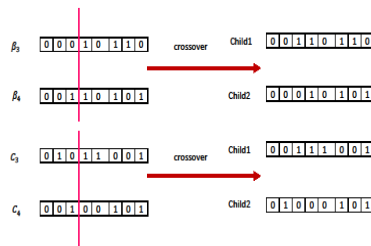


Figure 3.5: Single point Crossover

Mutation

The mutation operator used is the flip mutation, where the algorithm replaces each selected entry by a random number selected uniformly from the range for that entry. In binary-coded, any bit $a \in \{0, 1\}$, is replace within probability operator p_m by his complementary $\bar{a} = 1 - a$ see 3.6.

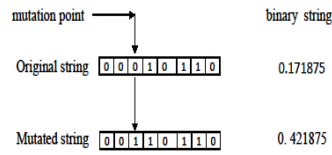


Figure 3.6: Binary Mutation

3.1.1 Iterative process

With a program in Matlab software for modeling our data 3.1, we have the follow result:

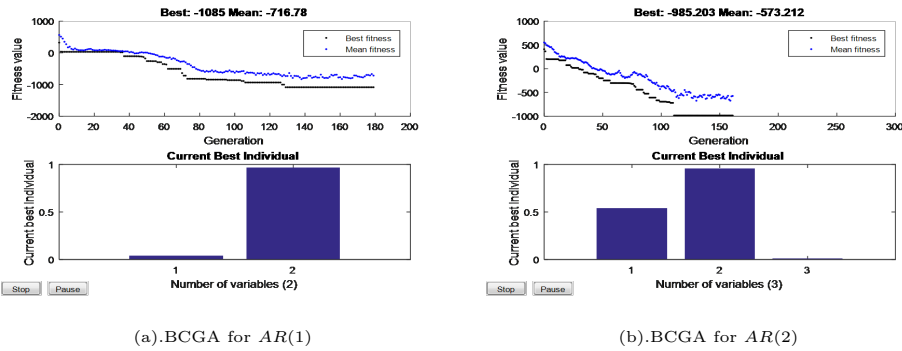


Figure 3.7: Comparison between for $AR(1)$ and $AR(2)$

Table 3.5: BCGAs method

MODEL	MSE	AIC	BIC	parameters
MOD3: $AR(1)$	1.71819	20.77943	20.77943	$(-0.27916, 0.96782)$
MOD4: $AR(2)$	$4.99692e - 10$	$-6.36511e + 02$	$-6.29307e + 02$	$(0.53855, 0.95561, 0.01280)$

Table 3.6: comparison between genetic algorithms and Box-jenkins models

Method	Model	MSE	AIC	BIC
Box-jenkins	MOD1:	47604.08	614.35	618.7512
genetic algorithms	MOD4:	$4.99692e - 10$	$-6.36511e + 02$	$-6.29307e + 02$



Figure 3.8: Comparison between BCGA model and Box-Jenkins model

It's clear that the model MOD4 introduced by genetic algorithms has a fewer statistics MSE , AIC and BIC , then of the model proposed by Box-Jenkins method. So we can conclude that the genetic algorithms method introduce the best model for this time series and improve the forecast.

3.2 BCGA on Autoregressive models MA(q)

In this example, we show the application of Box-Jenkins and BBGAs methodologies for giving $MA(p)$ model for euro hourly index data in Europe in 09/01/2017 see (www.investing.com):

First part Box-Jenkins application:

1) Visualize the Time Series in training faze

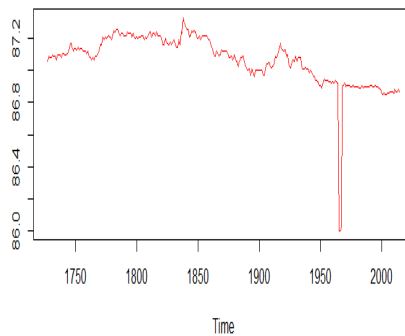


Figure 3.9: Hourly euro Index

2) Stationarity the Series

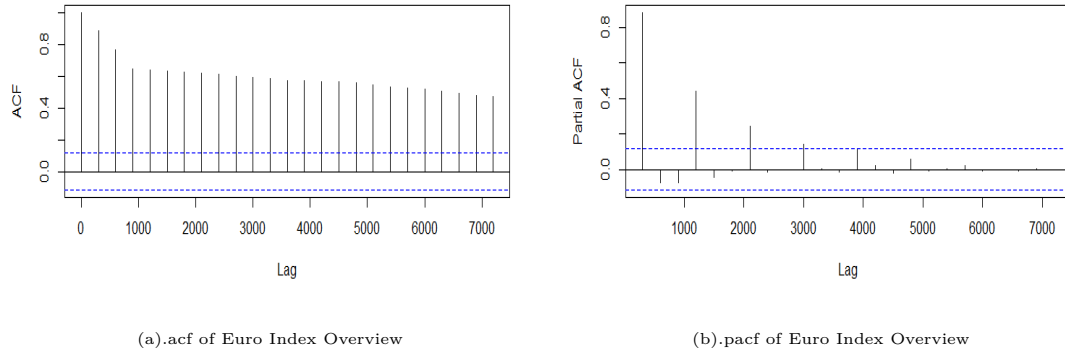


Figure 3.10: acf and pacf of time series

Augmented Dickey-Fuller Test

data: euro index ts

Dickey-Fuller = -2.6335, Lag order = 15, p-value = 0.3092

alternative hypothesis: stationary

That is clear the time series non stationary, then we need differentiate data for have stationary:

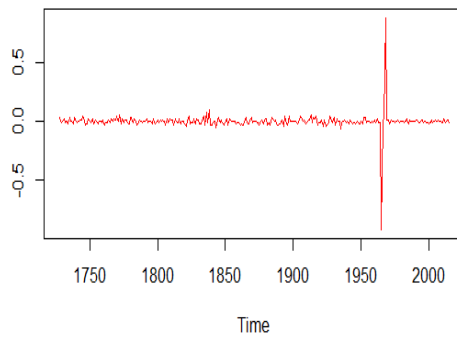


Figure 3.11: diff euro index time series plot

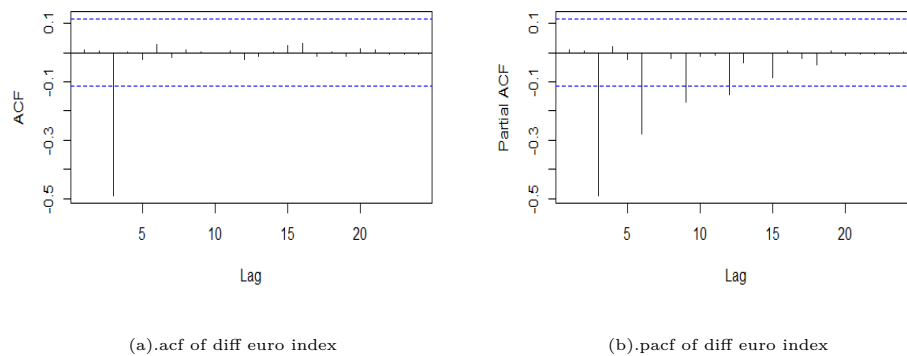


Figure 3.12: acf and pacf of differentiate time series

Augmented Dickey-Fuller Test

data: diff euro index ts

Dickey-Fuller = -5.9704, Lag order = 15, p-value = 0.01

alternative hypothesis: stationary

That is clear that differentiate data is stationery and the third spike is significant in ACF, but PACF is Tails off. Then we can say the model of difference data has an MA(3).

3) Find Optimal Parameters of Model

With `auto.arima` command in R, the better model proposed with less *AIC* and *BIC* is ARIMA(0, 0, 3) model, indeed:

```
arima(x = diff euro index ts, order = c(0, 0, 3), include.mean = TRUE)
```

Coefficients:

σ^2 estimated as 0.003966: *loglikelihood* = 386.67, *aic* = -765.35, *BIC* = -745.0304,

Table 3.7: parameters estimation

ma1	ma2	ma3	intercept
-0.0172	-0.0209	-0.6933	-8e-04
s.e. 0.0432	0.0454	0.0429	1e-03

$MSE = -0.0009486277$

4) Residuals diagnostic

Box-Ljung test

data: diff euro index residuals

X-squared = 1.1434, df = 15, p-value = 1

Jarque Bera Test

data: euro index residuals

X-squared = 260750, df = 15, p-value < 2.2e-16

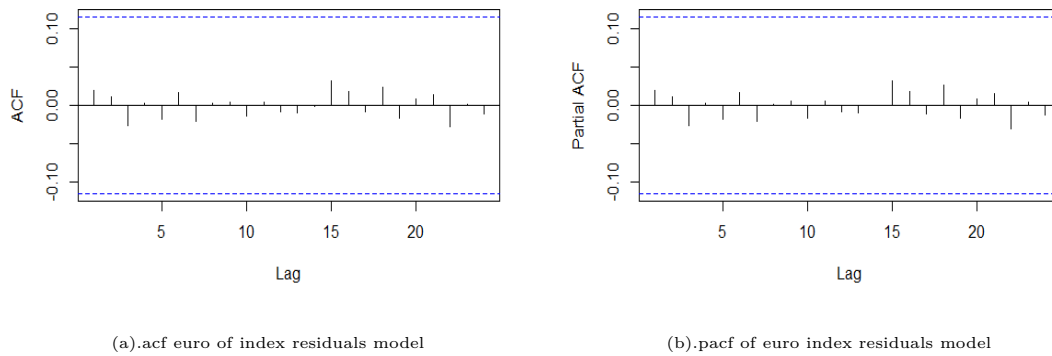


Figure 3.13: Residuals diagnostic

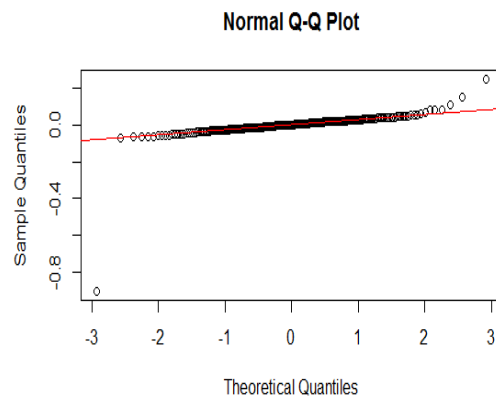


Figure 3.14: qqnorm of residuals model

Second part: BBGAs application:

Calling now GAs for giving the model for our data. This method is described in the following steps:

1. Calculate the mean of observations X_t , i.e \bar{X}_t
2. Calculate the errors ϵ_t by taking difference $X_t - \bar{X}_t$
3. Centering errors ϵ_t i.e calculate :

$$\epsilon'_t = \epsilon_t - \bar{\epsilon}_t, \tag{3.1}$$

where $\bar{\epsilon}_t$ is the mean of ϵ_t

Find Optimal Parameters of Model

Now we estimate moving average models parameters for lag 3 by BCGA, proposed by the first result of box-Jenkins technic. And we take a comparison between them.

When we follow all steps of BCGA in the same of the first application in section (1), by

estimate $MA(3)$ parameters with optimized the function AIC and calculate all statistics. So, we have the following result:

Table 3.8: Comparison between BCGAs model and Box-Jenkins model

MODEL	MSE	AIC	BIC	parameters
Box-Jenkins model MA(3)	-765.35	-745.0304	-0.00095	(-0.0172 , -0.0209, -0.6933, -8e-04)
BCGA model MA(3)	3.48301e-15	-9.48190e+03	-9.46794e+03	(-0,00066,-0.39821 ,0.78180,-0.39924)

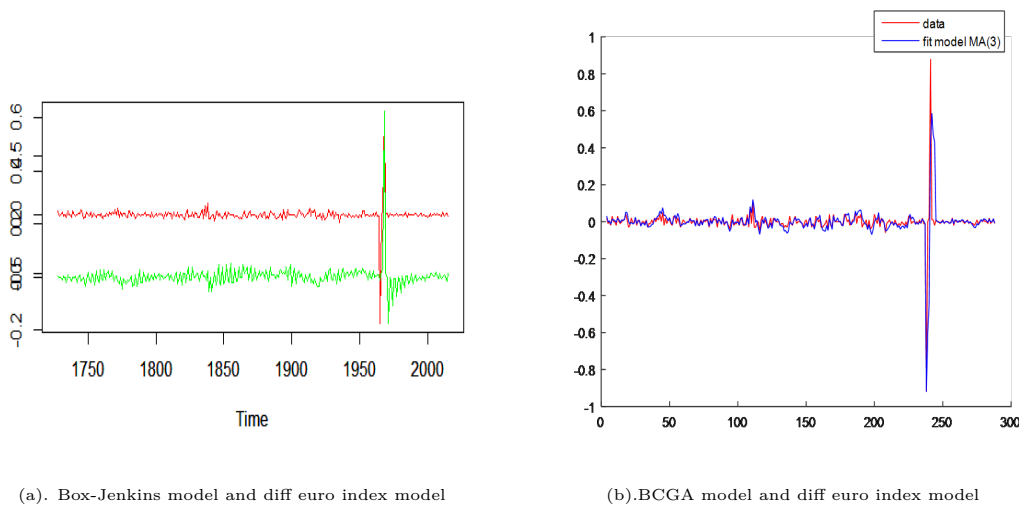


Figure 3.15: Comparison between Box-Jenkins and BCGA models

Comparing between all indicators on table 3.8 for two models, and show on the figure 3.15, we conclude that the model introduces by BCGA method is better then the model introduce by Box-Jenkins method.

3.3 BCGA on Seasonal Autoregressive Integrated Moving Average models $SARIMA(p, d, q)$, application study

This study was introduced in the article published on «international journal of statistics and economics», Vol.18 Issue. N°1(Année 2017).pages 1-15.

Chapter 4

Real Genetic Algorithms RCGA and comparison study with BCGA models

4.1 Introduction

For continuous optimization problems, real numbers representation is a natural way represent, solutions no difference between genotypes and phenotypes. A chromosome is a vector of floating point numbers and each gene represents a variable of the problem $x \in R^n$. The use of real coding RCGA initially appears on the specific application, such as in (Laisias and Al.,1989) for chemistries problems, and in (Davis,1989) for the use meta operator in order to find an adequate parameters for standard GA. The main difference in implementation of RCGA and BCGA is their recombination operators(crossover and mutation operators).

4.2 Crossover

They have many crossover operators proposed for this type encoding:

4.2.1 Whole arithmetic crossover method AMXO

In Arithmetic crossover, two parents produce two offspring:

If $c^k = (c_1^k, c_2^k, \dots, c_n^k)$ for $k = 1, 2$. are the parents, then the children are:

$$\begin{aligned} ch_i^1 &= \lambda c_k^1 + (1 - \lambda)c_k^2, \\ ch_i^2 &= \lambda c_k^2 + (1 - \lambda)c_k^1. \end{aligned}$$

Where λ is a proper fraction which can be chosen randomly.

Remark 4.2.1. • Alternatively, the proper fraction λ can be chosen in the following way:

$$\lambda = \frac{f_{max}}{f_{max} + f_{min}}$$

Where f_{max} denotes the max of fitness value of chromosomes and f_{min} denotes the min of fitness value of chromosomes.

- If λ is constant, then the crossover is called uniform arithmetical crossover. Otherwise, it is known as non-uniform arithmetic crossover.

4.2.2 Local arithmetical crossover LAC

the same as whole arithmetical crossover, except that the value of λ is randomly selected for each gene location.

4.2.3 linear crossover LC

If the parents are: $ch^k = (c_1^k, c_2^k, \dots, c_n^k)$ for $k = 1, 2, 3$. then the children are:

$$\begin{aligned} ch_i^1 &= \frac{1}{2}c_i^1 + \frac{1}{2}c_i^2, \\ ch_i^2 &= \frac{3}{2}c_i^2 + -\frac{1}{2}c_i^1, \\ ch_i^3 &= -\frac{1}{2}c_i^1 + \frac{3}{2}c_i^2. \end{aligned}$$

An offspring selection mechanism is applied, which chooses the two parents in the population.

4.2.4 Extended line crossover ELC

In extended line crossover, one offspring is generated from two parents. The offspring is given by:

$$ch_i = \lambda c_i^1 + \lambda(c_i^2 - c_i^1).$$

where λ is randomly chosen, (uniformly λ in $[-0.25, 1.25]$)

4.2.5 linear BGA crossover (Schlierkamp-Veosen 1994)

Under same consideration as above :

$$ch_i = \lambda c_i^1 + r_i \gamma \Lambda$$

Where $\Lambda = \frac{(c_i^2 - c_i^1)}{\|(c_1 - c_2)\|}$ the sign is chosen with a probability of 0.9. Usually r_i is $0.5(UB_i - LB_i)$ and

$$\gamma = \sum_{k=0}^{15} \alpha_k 2^{-k},$$

where $\alpha_k \in \{0, 1\}$ is randomly generated with $P(\alpha = 1) = \frac{1}{16}$. And LB is lower bound and UB is upper bound.

4.2.6 Flat crossover FC

An offspring $ch = (c_1, c_2, \dots, c_n)$ is generated where c_i is randomly (uniformly) chosen value from the set $[c_i^1, c_i^2]$.

4.2.7 Blend Crossover $BLX - \alpha$

The $BLX - \alpha$ crossover operator, where α is positive, mates two parents to produce one offspring. An offspring is the following:

$ch = (c_1, c_2, \dots, c_n)$, where c_i is a randomly (uniformly) chosen number of interval $[a_i - d_i\alpha, b_i + d_i\alpha]$, where $a_i = \min(c_i^1, c_i^2)$, $b_i = \max(c_i^1, c_i^2)$ and $d_i = |c_i^1 - c_i^2|$.

Remark 4.2.2. The $BLX - 0.0$ crossover is equal Flat crossover.

For the value of α , Eshelman and Schafer have used $\alpha = 0.5$.

4.2.8 $PBX - \alpha$ crossover

In $PBX - \alpha$, two parents produce one offspring. The offspring is given by equation $ch = (c_1, c_2, \dots, c_n)$. The difference compared to $BLX - \alpha$ is that c_i is randomly chosen number from the $[l_i, u_i]$ interval. The libris l_i and u_i are given by:

$$l_i = \max(LB_i, c_i^1 - I_i\alpha), u_i = \min(UB_i + I_i\alpha) \text{ and } I_i = |c_i^1 - c_i^2|.$$

4.2.9 Heuristic crossover operator (HX)

The HX has been applied to solve nonlinear constraint optimization problem and as well as un constrained optimization problems having various levels of difficulty. The offspring is generated in the following manner :

$$ch_i = \alpha(c_i^2 - c_i^1) + c_i^2$$

where α is a uniformly distributed randomly number in interval $[0, 1]$.

It should be noticed that the parent c^2 has fitness value not worse than that of the parent c^1 . If the offspring lees outside the feasible region, a new random number is generated to produce another offspring using:

$$ch_i^2 = (c_i^2 + \beta(c_i^1) - c_i^2)$$

where β is number, witch is generated using the laplace distribution function. The process is repeated up to k times.

4.2.10 Parent-centric crossover PCCO

The PCCO create the offspring in the neighborhood of the female parent using a probability distribution. The male parent is considered to define the rang of probability distribution.

4.2.11 laplace crossover operator LX

A new crossover operator which uses Laplace distribution was introduced. This parent centric crossover operator has been named as Laplace crossover LX. The density function of Laplace distribution is:

$$f(x) = \frac{1}{2b} \exp\left(-\frac{|x-a|}{b}\right), \quad -\infty < x < +\infty;$$

the corresponding distribution function is as follow:

$$F(x) = \begin{cases} \frac{1}{2} \exp\left(\frac{|x-a|}{b}\right) & x \leq a; \\ 1 - \frac{1}{2} \exp\left(-\frac{|x-a|}{b}\right) & x > a; \end{cases}$$

where $a \in R$, and $b > 0$.

the α is a random number in $[0,1]$ is generated.

And a random β is generated which follows the Laplace distribution by simply inverting the distribution function of Laplace distribution as follows:

$$\beta = \begin{cases} a - b \ln(\alpha) & \alpha \leq \frac{1}{2} \\ a + b \ln(\alpha) & \alpha > \frac{1}{2} \end{cases}$$

The offspring are giving by the equations:

$$\begin{aligned} ch_i^1 &= c_i^1 + \beta |c_i^1 - c_i^2| \\ ch_i^2 &= c_i^2 + \beta |c_i^1 - c_i^2|. \end{aligned}$$

Both offspring symmetrically with respect to position of the parents.

4.2.12 Multiple crossover MX

A multi-crossover MX formula with more than two chromosomes was proposed. It was assumed that three chromosome c^1 , c^2 and c^3 are selected from the population randomly. let α is a random number selected from $[0,1]$, if $\alpha \geq P_c$, then no crossover operation is performed. If $\alpha < P_c$, then multiple crossover formulas are the following:

$$\begin{aligned} ch_i^1 &= (c_i^1 + r(2c_i^1) - c_i^2 - c_i^3) \\ ch_i^2 &= (c_i^2 + r(2c_i^1) - c_i^2 - c_i^3). \\ ch_i^3 &= (c_i^3 + r(2c_i^1) - c_i^2 - c_i^3). \end{aligned}$$

where $r \in [0, 1]$ is a random value determining the crossover grade.

4.3 Mutation operators in real coded genetic algorithms RCGA

Let us suppose $c = (c_1, c_2, \dots, c_n)$ a chromosome and $c_i \in [LB_i, UB_i]$, a gene to be muted. Next the gene c_i resulting from the application of different mutation operators is shown. Mutation is regulated with the mutation probability P_m . Let us introduce a various of mutation operators used in RCGA.

4.3.1 Power mutation operator PM

The mutation operators is based on power distribution :

$$f(x) = px^p, 0 \leq x \leq 1$$

The corresponding density function is as follows:

$$F(x) = x^p, 0 \leq x \leq 1$$

where p is the index of distribution. This mutation is used to create a solution c' in the vicinity of a parent solution c . A uniform random number t between 0 and 1 created, and s created which follow the mentioned distribution:

$$c' = \begin{cases} c + s(c - LB) & \text{if } t < \alpha \\ c + s(UB - c) & \text{if } t \geq \alpha. \end{cases}$$

Where $t = \frac{c-LB_i}{UB_i-LB_i}$, l and u are lower and upper bound and α is a uniformly distributed random between [0.1]. The probability of producing a mutation solution c' on left(right) side of c is proportional to distance of c from $[LB, UB]$ and the muted solution is always feasible.

4.3.2 Boundary mutation BM

The boundary mutation is a variation of the uniform mutation. The difference is that a selected element of chromosome is replaced by the lower or upper boundary of the feasible area.

4.3.3 Random mutation RM

The c_i is a random (uniform) number from domain $[LB_i, UB_i]$.

4.3.4 Makinen-Periaux and Toivanen mutation

It has been applied to solve some multidisciplinary shape optimization problem in aerodynamics and electromagnetic as well as a large set of constrained optimization problem (Deep and Thakar 2007).

Let us $\alpha \in [0.1]$; Then the mutation point c' is :

$$c' = (1 - \hat{t})l_i + \hat{t}u_i,$$

where

$$\hat{t} = \begin{cases} t - t\left(\frac{t-r}{t}\right)^b & \text{if } r < t, \\ t & \text{if } r = t \\ t + (1-t)\left(\frac{r-t}{1-t}\right)^b & \text{if } r > t. \end{cases}$$

$$t = \frac{c-LB_i}{UB_i-c}.$$

Remark 4.3.1. This operator does not decrease as the generation in creases.

4.3.5 Mühlen Bein's mutation

The mutated element is given by:

$$c'_i = (c_i \mp r_i \gamma).$$

Where r_i defines the mutation rang and it is normally set to $0.1(UB_i - LB_i)$. The + or - sign is chosen with a probability of 0.5 and :

$$\gamma = \sum_{k=0}^{15} \alpha_k 2^{-k}, \alpha_k \in [0, 1]. \quad (4.1)$$

Is randomly generated with $P(\alpha_i = \frac{1}{16})$, value in the interval $[C_i - r_i, C_i + r_i]$, are generated using this operator, with probability of generating a neighborhood of C_i being very high. The minimum possible proximity is produced with precision of $r_i 2^{-15}$.

4.3.6 Discrete mutation (DX)

For DX, we choose γ , where:

$$\gamma = \sum_{k=0}^{\pi} \alpha_k B_m^k, \quad (4.2)$$

with,

$$\pi = \frac{\log(r_{min})}{\log(B_m)}, B_m > 1, \quad (4.3)$$

is a parameter called the base of the mutation and r_{min} is the lower limit of the relative mutation rang.

4.3.7 Continuous mutation CX

The same of DX but:

$$\gamma = \sum_{k=0}^{\pi} \alpha_k \phi(B_m^k), \quad (4.4)$$

with $\phi(z_k)$, being a triangular probability distribution, with $\frac{B_m^k - B_m^{k-1}}{2} \leq z_k \leq \frac{B_m^{k+1} - B_m^k}{2}$.

4.3.8 Creep mutation

In creep mutation a random gene is selected and its value changed with a random value between lower and upper bound.

4.3.9 Uniform mutation UM

The mutation operator changes the value of chosen gene with uniform random value selected between the user specified upper and lower bound, for that gene. It used in case of integer representation.

4.3.10 Non-uniform mutation operator NUM

The element to be mutation at k th generation is denoted by $c, c_i \in [LB_i, UB_i]$. As before LB_i and UB_i are lower and upper bounds of c_i respectively. The mutated segment is given by:

$$c'_i = \begin{cases} c_i + \Delta(k, UB_i - c_i) & \text{if } t = 0 \\ c_i - \Delta(k, c_i - LB_i) & \text{if } t = 1 \end{cases}$$

where t is a random digit that takes either the value 0 or 1. The value of the function Δ is calculated by:

$$\Delta(k, y) = y(1 - \alpha^{(1-\frac{k}{T})^b}) \tag{4.5}$$

where α is a uniformly distributed random number in interval $[0,1]$, T is the maximum number of generations, b is a parameter determining the degree of non-uniformity. This operation returns a value in the rang $[0, y]$, such that the probability of returning a number close to zero increase as k increases. In the initial generations non-uniform mutation tends to search the space locally.

Remark 4.3.2. This method will be more used in comparison study in next section in RCGA, for this we will be showing the convergence of this operators based on the stochastic process theory in one dimensional case, i.e., $n = 1$. See [91]

Définition 4.3.3. Given a vector $c = (c_1, \dots, c_i, \dots, c_m)$, where m is the dimension of vectors. We call c' is its neighbor, if and only if one of its component is changed and other components remain unchanged. The neighborhood N of a vector c consists of all its neighbors. That is, $N = \{c', c', \text{ is a neighbor of } c\}$.

We assume that $f(x)$ has a unique minimal value at x^* . In Fig.4.1, let x_0 be one initial solution, x'_0 another initial solution lying between x^* and x_0 , \underline{x}_0 a number satisfying $f(\underline{x}_0) = f(x_0)$ and $\underline{x}_0 \neq x_0$, and ϵ an arbitrary small positive number. Without loss of generality, we assume that variable x lies on the right side of x^* and \underline{x} lies on the left side. Based on this method, we have

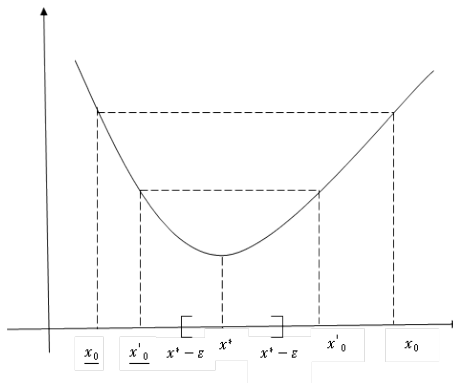


Figure 4.1: Analysis on the unimodal function

$$x_1 = \begin{cases} x_0 & \text{if } \zeta = 0, \text{ or if } \zeta = 1, \text{ and } x_0 - \Delta(1, x_0 - LB) \leq \underline{x}_0 \\ x_0 - \Delta(1, x_0 - LB) & \text{if } \zeta = 1 \text{ and } x_0 - \Delta(1, x_0 - LB) > \underline{x}_0 \end{cases} \tag{4.6}$$

$$x'_1 = \begin{cases} x'_0 & \text{if } \zeta = 0, \text{ or if } \zeta = 1, \text{ and } x'_0 - \Delta(1, x'_0 - LB) \leq \underline{x}'_0 \\ x'_0 - \Delta(1, x'_0 - LB) & \text{if } \zeta = 1 \text{ and } x'_0 - \Delta(1, x'_0 - LB) > \underline{x}'_0 \end{cases} \tag{4.7}$$

Lemme 4.3.4. Let $p_1 = P\{x_1 \notin (x^* - \epsilon, x^* + \epsilon)\}$, $p'_1 = P\{x'_1 \notin (x^* - \epsilon, x^* + \epsilon)\}$, if $x^* < x'_0 < x_0$, then $p'_1 < p_1$. Similarly if $x_0 < x'_0 < x^*$, $p'_1 < p_1$ also hold.

Proof. We have

$$\begin{aligned} p_1 &= 1 - P\{x_1 \in (x^* - \epsilon, x^* + \epsilon)\} \\ &= 1 - P\{\zeta = 1, x^* - \epsilon < x_0 - \Delta(1, x_0 - LB) < x^* + \epsilon\} \\ &= 1 - \frac{1}{2}P\{x^* - \epsilon < x_0 - \Delta(1, x_0 - LB) < x^* + \epsilon\} \end{aligned}$$

Similarly $p'_1 = 1 - \frac{1}{2}P\{x^* - \epsilon < x'_0 - \Delta(1, x'_0 - LB) < x^* + \epsilon\}$.

Let $q = P\{x^* - \epsilon < x_0 - \Delta(1, x_0 - LB) < x^* + \epsilon\}$

and $q' = P\{x^* - \epsilon < x'_0 - \Delta(1, x'_0 - LB) < x^* + \epsilon\}$.

Thus it is enough to show that

$$q < q' \tag{4.8}$$

By Eq.(4.5), we have

$$\begin{aligned} q &= P\{x^* - \epsilon < x_0 - \Delta(1, x_0 - LB) < x^* + \epsilon\} \\ &= P\{x^* - \epsilon < x_0 - (x_0 - LB)(1 - \alpha^{(1-\frac{1}{T})^b}) < x^* + \epsilon\} \\ &= P\left\{\left(\frac{x^* - LB - \epsilon}{x_0 - LB}\right)^{\frac{1}{m}} < \alpha < \left(\frac{x^* - LB + \epsilon}{x_0 - LB}\right)^{\frac{1}{m}}\right\} \end{aligned}$$

$$q = \left(\frac{x^* - LB + \epsilon}{x_0 - LB}\right)^{\frac{1}{m}} - \left(\frac{x^* - LB - \epsilon}{x_0 - LB}\right)^{\frac{1}{m}} \tag{4.9}$$

$$\tag{4.10}$$

Where $m = (1 - \frac{1}{T})^b$.

Similarly, we have

$$q' = P\{x^* - \epsilon < x'_0 - \Delta(1, x'_0 - LB) < x^* + \epsilon\}. \tag{4.11}$$

$$\tag{4.12}$$

$$= \left(\frac{x^* - LB + \epsilon}{x'_0 - LB}\right)^{\frac{1}{m}} - \left(\frac{x^* - LB - \epsilon}{x'_0 - LB}\right)^{\frac{1}{m}} \tag{4.13}$$

$$\tag{4.14}$$

From 4.13,4.9, and let q subtracts q' , we may derive Eq. (4.8) and thus the correctness of the Lemma. Since x_0 is a given initial solution (individual), we can assume that $x_0 > x^*$.

Let

$$p_1^+ := P\{x_0 \text{ is the initial solution and } x_1 \notin (x^* - \epsilon, x^* + \epsilon)\}$$

$$p_1^- := P\{x_0 \text{ is the initial solution and } x_1 \notin (x^* - \epsilon, x^* + \epsilon)\}$$

Then, $p_1 := p_1^+$, (or p_1^-)

For $n \geq 2$, we define

$$p_n^+ := P\{x_{n-1} > x^*, x_n \notin (x^* - \epsilon, x^* + \epsilon)\}$$

$$p_n^- := P\{x_{n-1} < x^*, x_n \notin (x^* - \epsilon, x^* + \epsilon)\}$$

Then, $p_n := p_n^+ + p_n^-$ □

Théorème 4.3.5. *For any $\epsilon > 0$, we have, $\lim_{n \rightarrow \infty} p_n = 0$*

Proof. By the description of NUM, it is easy to know that the stochastic process $\{x_i; i = 0, 1, 2, \dots\}$ is a Markov process. By the property of conditional expectation, Markov property and Lemma 4.3.4, we can obtain that

$$\begin{aligned} p_2 &= P\{x_2 \notin (x^* - \epsilon, x^* + \epsilon)\} \\ &= E(I_{x_2 \notin (x^* - \epsilon, x^* + \epsilon)}) \\ &= E(E(I_{x_2 \notin (x^* - \epsilon, x^* + \epsilon)} | x_1)) \\ &= E(E^{x_1}(I_{x_2 \notin (x^* - \epsilon, x^* + \epsilon)})) \\ &= E^{x_1}(I_{x_2 \notin (x^* - \epsilon, x^* + \epsilon)}) P\{x_1 \notin (x^* - \epsilon, x^* + \epsilon)\} \\ &\leq \max\{p_1^+, p_1^-\} p_1 \end{aligned}$$

$$\begin{aligned} p_3 &= P\{x_3 \notin (x^* - \epsilon, x^* + \epsilon)\} \\ &= E(I_{x_3 \notin (x^* - \epsilon, x^* + \epsilon)}) \\ &= E(E(I_{x_3 \notin (x^* - \epsilon, x^* + \epsilon)} | x_2)) \\ &= E(E^{x_2}(I_{x_3 \notin (x^* - \epsilon, x^* + \epsilon)})) \\ &= E(E^{x_2}(I_{x_1 \notin (x^* - \epsilon, x^* + \epsilon)})) \\ &= E^{x_2}(I_{x_1 \notin (x^* - \epsilon, x^* + \epsilon)}) P\{x_2 \notin (x^* - \epsilon, x^* + \epsilon)\} \\ &= \max\{E^{x_1}(I_{x_1 \notin (x^* - \epsilon, x^* + \epsilon)}), E^{x_1}(I_{x_1 \notin (x^* - \epsilon, x^* + \epsilon)})\} P_2 \\ &\leq (\max\{p_1^+, p_1^-\})^2 p_1 \end{aligned}$$

By induction we have

$$\begin{aligned} p_n &= P\{x_n \notin (x^* - \epsilon, x^* + \epsilon)\} \\ &\leq (\max\{p_1^+, p_1^-\})^{n-1} p_1 \end{aligned}$$

Obviously, $0 < p_1^+, p_1^- < 1$, so $\lim_{n \rightarrow \infty} p_n = 0$. □

By the greedy selection of NUM, it is easy to know that

$$p_n = P\{x_i \notin (x^* - \epsilon, x^* + \epsilon); i = 0, 1, 2, \dots\}.$$

So Theorem implies that for any $\epsilon > 0$, we have

$$\lim_{n \rightarrow \infty} (1 - P\{x_i \notin (x^* - \epsilon, x^* + \epsilon); i = 0, 1, 2, \dots\}) = 1$$

i.e

$$\lim_{n \rightarrow \infty} (P\{\exists i = 0, 1, 2, \dots, n, \text{ s.t. } x_i \in (x^* - \epsilon, x^* + \epsilon)\}) = 1 \quad (4.15)$$

Eq.4.15 indicates that for any $\epsilon > 0$; $\{x_i\}_{i=1}^{\infty}$ is to enter the domain $(x^* - \epsilon, x^* + \epsilon)$ almost surely, and so $\{x_i\}_{i=1}^{\infty}$, converges to x^* almost surely.

4.4 Comparison study between BCGA and RCGA models

In this section, we will compare between two types of GAs (BCGA ,RCGA) and Box-Jenkins method, applied on monthly hotel room averages data in Kuwait figure 1.9 in chapter one. The plan of this study is:

1. Estimate the parameters of Box-Jenkins model proposed with BCGA and RCGA using same crossover operators and different mutation operators,
2. Estimate the parameters of Box-Jenkins model with BCGA and RCGA using different crossover and mutation operators,
3. comparison between all results.

The crossover and mutation operators used in this comparison are,:

First step:

- Single point crossover and two point crossover operators for RCGA and BCGA,
- Non-uniform mutation operator for all RCGA and flip mutation for all BCGA,

Second step:

- Extended linear crossover, Laplace crossover arithmetic crossover for RCGA and two point crossover operator for all BCGA,
- Non-uniform mutation operator for RCGA and flip mutation for all BCGA,

Third step:

- Arithmetic crossover for RCGA and two point crossover operator for BCGA,
- Makinen-Periaux and Toivanen mutation operator for RCGA and flip mutation for all BCGA,

Remark 4.4.1. 1. We use the comparison in all step with three selection methods: roulette wheel selection, tournament selection, and random selection.

2. The figure used in this study is for difference data of monthly hotel clients data in Kuwait.

So we have don't forget that the result of Box-Jenkins method in chapter one, the ARIMA(3,1,3)(1,1,1)[12] model is the better model for this data:

- 1) In the first, when we applique the RCGA and BCGA algorithms using same crossover operators and different mutation operators, we have the following result on figures and tables:

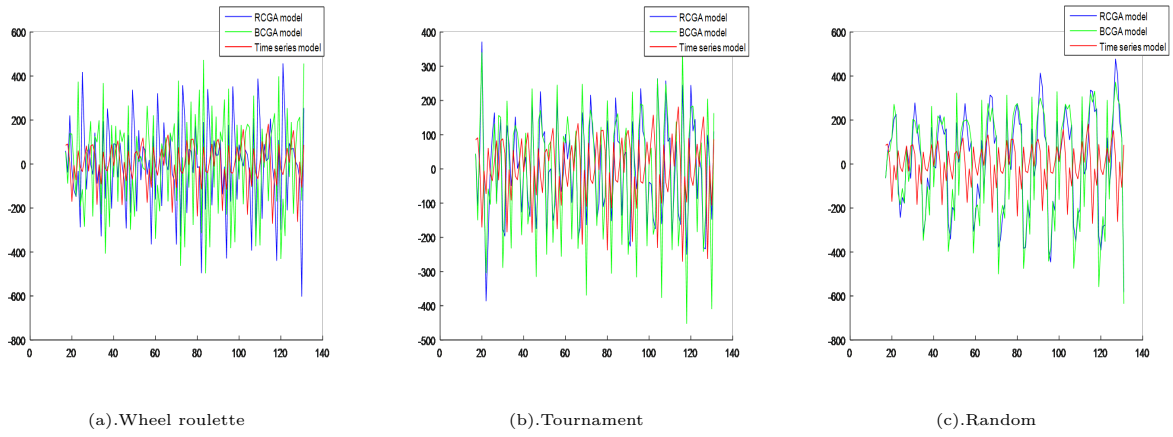


Figure 4.2: One point crossover

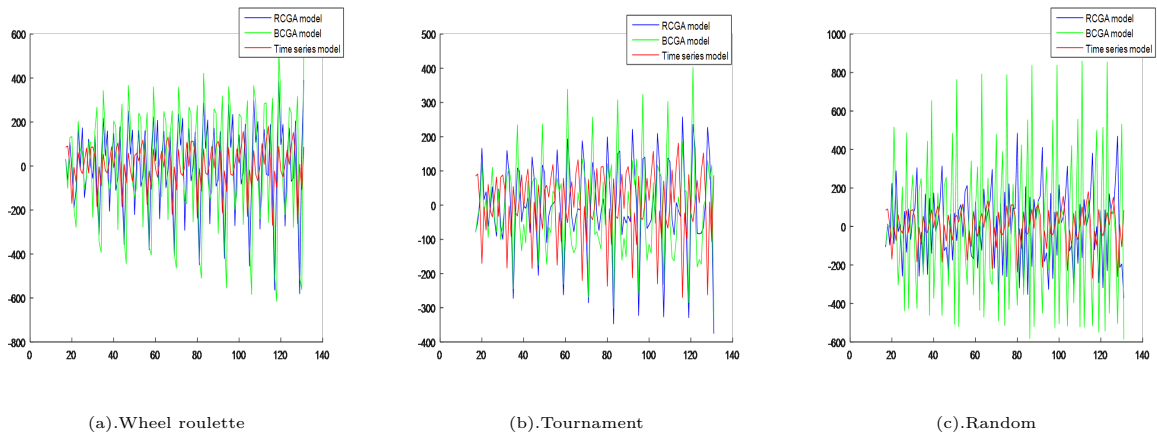


Figure 4.3: Two point crossover

Table 4.1: BCGA application

BINARY CODED GENETIC ALGORITHMS(BCGA)					
Crossover and Mutation	Selection types	AIC	BIC	MSE	ARE_f
Single point with flip mutation	Wheel roulette	-1.3043e+03	-1.2652e+03	1.6045e-06	28.3232
	Tournament	-1.1929e+03	-1.1539e+03	4.8847e-06	5.8328
	Random	-1.0859e+03	-1.0468e+03	1.4250e-05	55.3441
Two point with flip mutation	Wheel roulette	-1.2624e+03	-1.2233e+03	2.4392e-06	36.4903
	Tournament	-1.2571e+03	-1.2181e+03	2.5707e-06	31.2935
	Random	-1.0441e+03	-1.0050e+03	2.1648e-05	51.6448

Table 4.2: RCGA application

		REAL CODED GENETIC ALGORITHMS(RCGA)			
Crossover and Mutation	Selection types	<i>AIC</i>	<i>BIC</i>	<i>MSE</i>	<i>ARE_f</i>
Single point with non-uniform mutation	Wheel roulette	-1.1745e+03	-1.1699e+03	2.8275e-05	12.8853
	Tournament	-1.7401e+03	-1.7356e+03	2.0664e-07	1.7230
	Random	-1.5518e+03	-1.5473e+03	1.0625e-06	51.3316
Two point with non-uniform mutation	Wheel roulette	-1.4027e+03	-1.3981e+03	3.8865e-06	23.3916
	Tournament	-1.3815e+03	-1.3770e+03	4.6722e-06	35.4764
	Random	-1.3182e+03	-1.3137e+03	8.1015e-06	35.2068

2) In the second, when we apply the RCGA and BCGA algorithms using different crossover and mutation operators, we have the following result on figures and tables:

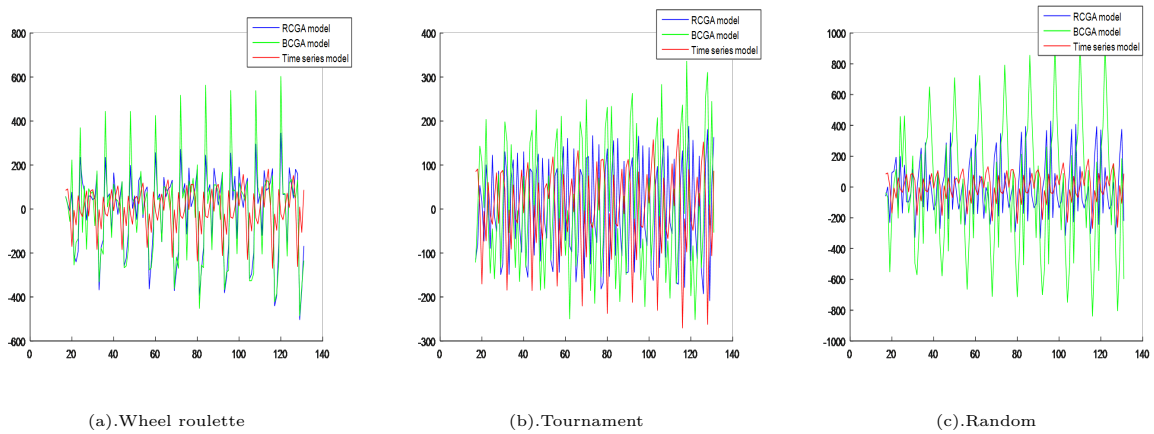


Figure 4.4: Extended linear crossover

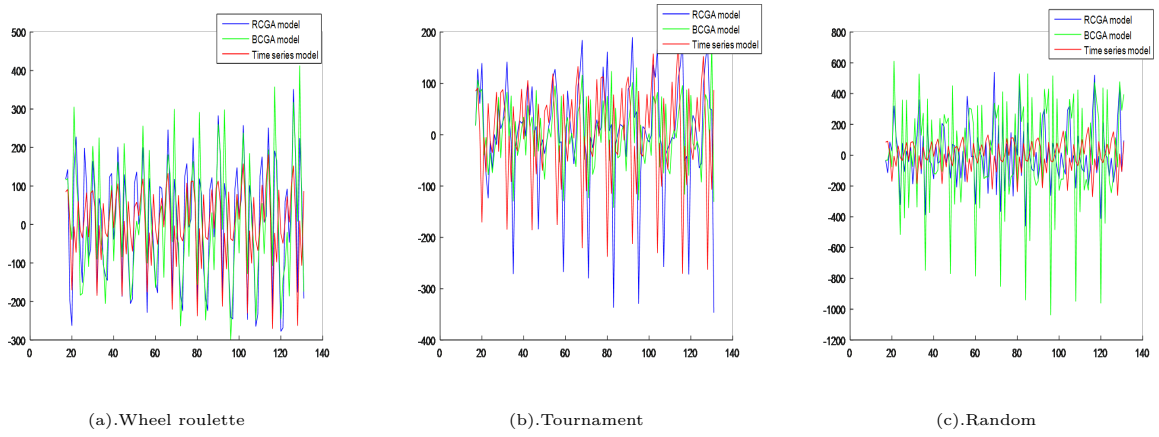


Figure 4.5: Laplace crossover

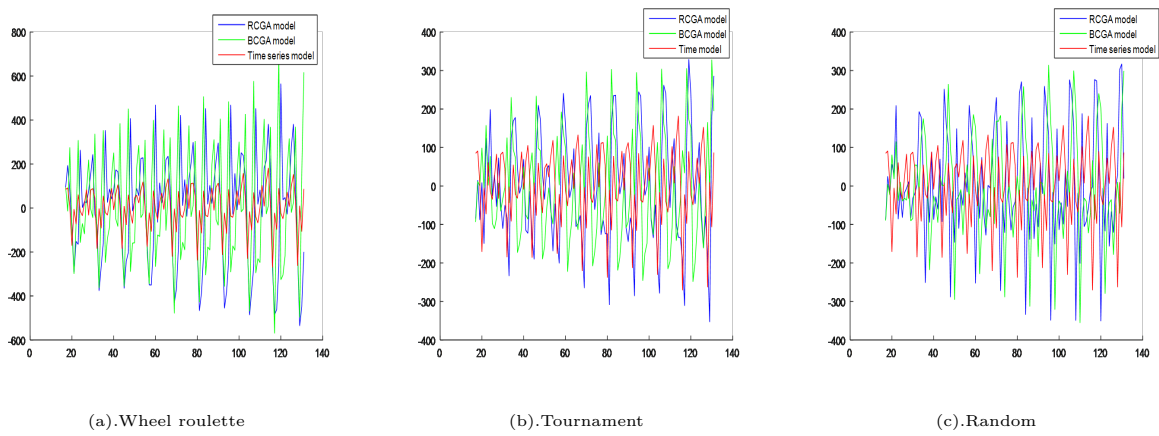


Figure 4.6: Arithmetic crossover

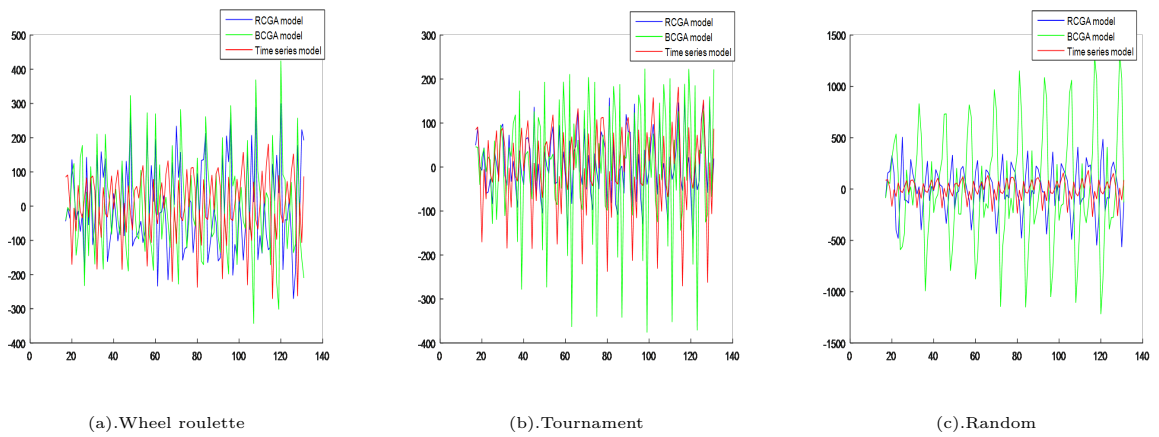


Figure 4.7: Makinen-Periaux and Toivanen mutation

Table 4.3: RCGA with various crossover and mutation operators

		REAL CODED GENETIC ALGORITHMS(RCGA)			
Crossover and Mutation	Selection types	<i>AIC</i>	<i>BIC</i>	<i>MSE</i>	<i>ARE_f</i>
Extended linear crossover with non-uniform mutation	Wheel roulette	-1.3906e+03	-1.3860e+03	4.3176e-06	19.5356
	Tournament	-1.5341e+03	-1.5296e+03	1.2395e-06	5.8490
	Random	-1.4206e+03	-1.4160e+03	3.3263e-06	23.5867
Laplace crossover with non-uniform mutation	Wheel roulette	-1.7904e+03	-1.7859e+03	1.3345e-07	21.3803
	Tournament	-1.4714e+03	-1.4668e+03	2.1387e-06	33.2422
	Random	-2.9304e+03	-2.9258e+03	6.6117e-12	0.6769
Arithmetic crossover with non-uniform mutation	Wheel roulette	-1.8661e+03	-1.8615e+03	6.9113e-08	21.9403
	Tournament	- 2.1173e+03	-2.1127e+03	7.7775e-09	15.2560
	Random	-2.4454e+03	-2.4409e+03	4.4834e-10	5.1750
Arithmetic crossover with Makinen-Periaux and Toivanen mutation	Wheel roulette	-2.4010e+03	-2.3965e+03	6.5959e-10	8.0643
	Tournament	-1.7930e+03	-1.7884e+03	1.3049e-07	5.1825
	Random	-1.3651e+03	-1.3606e+03	5.3884e-06	15.8515

It easy to show that when we use some various of crossover and mutation operator, the results are performed for RCGA then of BCGA, because all statistics indicators *AIC*, *BIC*, *MSE* and *ARE_f* are minimums in general, and errors of models are reduced. Then we can conclude that using RCGA have many advantages, Better precision(we see that after) and easy to handle a large dimensional problem.

conclusion and perspective

Bibliography

- [1] Burcu N., Burak G., "Forecast Accuracy Measures: Results from Simple Moving Average Methods on Independent Normal Time Series", *International Journal of Statistics and Economics*. Volume 14, Issue Number: 2, 53-67. 2014.
- [2] Louie R., Yong U., Glasure P.R., "Forecasting Brazilian Industrial Production Index with level and trend changes after crisis and SARIMA models", *International Journal of Statistics and Economics*, Volume 17, Issue Number: 1, 22-29. 2016.
- [3] Selim S., Demir C., "Forecasting with ARIMA model on External trade of ALgeria", *International Journal of Statistics and Economics*, Volume 17, Issue Number: 2, 32-43. 2016.
- [4] Abdelhak CH., "Modelling Multivariate Durations", *International Journal of Statistics and Economics*, Volume 17, Issue Number: 1, 82-93. 2016.
- [5] A. El-Zein, M. M. El Bahy and M. Talaat., "A Simulation Model for Electrical Tree in Solid Insulation Using CSM Coupled with GAs", *IEEE CEIDP Conference*, October pp. 645-649. 2008
- [6] Tomaso B. M. and Irene P., "Building ARMA models with genetic algorithms", Springer, Verlag berlin, pp 335-342. 2001.
- [7] Chat F. C. and Prothero D. L., "Box-Jenkins seasonal forecasting: Problems in a case-study (with discussion)", *J.R statist. Soc.A*, 136,295-336.
- [8] Hannan E. J., 1980. "The estimation of the order of an ARMA process", *annal of institue of statistical*, vol. 8, n0 .5, pp 1071-1081.
- [9] Faraway,J.and ChatField C. 1998. "Time series forecasting with Neural Networks: A comparative study using the Airline data", *Appl. Statist.*, 47, 231-250.
- [10] Kirchgssner G., and Wolters J., 2007. "Introduction to Modern Time Series Analysis", Springer-Verlag Berlin Heidelberg.
- [11] Goldberg, D.E. 1989. "Genetic algorithms in search optimization and machine learning", Addison- wesley. Reading, MA.
- [12] H. Akaike, " Statistical in predictor identification", *annal of institue of statistical mathématiques*, vol. 22, n01, pp 203-217.

- [13] Holland J. M. 1992. "Adaptation in natural and artificial system", university of Michigan press. Ann Arbor, MI(Secand edition).
- [14] Gabr M.M., and Chikr Z. M. 2011. "Iterated Neural Networks models for time series analysis" , Metron, international journal of statistics, Vol.LXIX, n. 2,PP. 129-149.
- [15] Farahat M. and M. Talaat., A New Approach for Short-Term Load Forecasting Using Curve Fitting Prediction Optimized by Genetic Algorithms IEEE, 14th International Middle East Power Systems Conference MEPCON 10, pp. 106-110. 2010.
- [16] Peng P. and Chen Q. "Improved genetic algorithms and application to ARMA modeling", Sice annal conference in Fukai, Augst, 4-6. 2003.
- [17] Roberto B., Francesco B., and Domenico C., "Estimating threshold subset autoregressive moving-average models by genetic algorithms", Meron-international journal of statistics, vol. LyII; no1, pp. 39-61. 2004.
- [18] D.A coley, "An Intoduction to Genetic Algorithms for Scientists and Engineers", World Scientific Publishing, Singapore, 1999.
- [19] A.Chipperfield, P.Fleming, H.Pohlheim, C.Fonseca, "Genetic Algorithm Toolbox,For Use With Matlab, V 1.2 Users Guide, Dept. of Automatic Control and System Engineering", University of Sheffield, U.K., 1994.
- [20] M.Melanie, "An Introduction to Genetic Algorithms", Fifth printing, England, 1999.
- [21] T.Vallée, M.Yeldzoglu, "Economics Genetic Algorithmes Presentation and applications", Décembre, Université de Nantes et Université Montesquieu Bordeaux, v. 4.2, 2001.
- [22] HB.Nielsen, "Non-Stationary Time Seriesand Unit Root Tests", Econometrics 2 , Fall 2005.
- [23] U.Bodenhofer, "Genetic Algorithms Theory and Applications", FLLL, Second Edition, WS 2001/2002.
- [24] P.Bajpai, M.Kumar, "Genetic Algorithm, an Approach to Solve Global Optimization Problems", Indian Journal of Computer Science and Engineering Vol 1 No 3 199-206, 2010.
- [25] J.J. Daudin, C. Duby, S. Robin, P. Tr´ecourt, "Time Series Analysis", (INA-PG, Math´ematiques),May 1996.
- [26] G. Mohmoud., "Time Series Analysis", Kuwait university.2003.
- [27] K. Deep and M. Thakar, " A new crossover operator for real coded genetic algorithms", Applied Math. compute, vol. 188. no 1, pp. 859-911. 2007.
- [28] N. Turkan., " Optimization of multimodal continuouse function using a new crossover for real coded genetic algorithms". Expert syst-appl., 3614. pp.8172-8177. 2009.

- [29] K. Deb, A. Kumal., " Real coded genetic algorithms with simulated binary crossover: study on multimodal and multiobjective problems". *Complex systems*, vol. 9. pp. 431-454. 1995.
- [30] A. Wright., " Genetic algorithms for real parameter optimization". Edited by G.J.E. Rawlins.1991.
- [31] C.K. Oei, D.E Goldberg and S.J Chang., " Tournament selection, niching and preservation of diversity". University of Illinois, illgal report number 91011. 1991.
- [32] P. Arnold., " Genetic algorithms search for predictive pattents in multidimenssional time series ". *Complex systems*. vol.19. 2011.
- [33] J. Zhu, D Fucaili, and T. Fu., " Improved real coded genetic algorithms and it's simulation". *Journal of soft ware* vol.9 no 162. 2014.
- [34] M. M Grigioni, U. Morbidicci and A. Tura., " Genetic algorithms for parameter estimation in mathematical modeling of glucose metabolisme". *computer in biologie and medecine*, vol.35.no 10.pp. 862-874.2005.
- [35] A.V Panteleev, D.V. Metlestskaya., " An application of ganetic algorithms with binary and real coded for aproximate synthesis of suboptimal control in deterministic system"., *Automation and remote controle*, vol. 72.no.11. pp.2328-2338. 2011.
- [36] M. H. Satman., " Machine coded genetic algorithms for real parameter optimization problem". *Gazi university, journal of science GUJSI*, vol 26.pp.85-95.2013.
- [37] D.E. Golderg and K. Deb., " A comparison of selection schemes used in genetic algorithms",. In *foundations of genetic algorithms* vol.1. pp. 69-93. 1991.
- [38] D. Stevenson., " A proposed standard for binary floating pont arithmitic ",. Draft 8. of IEEE task. pp. 51-62. 1981.
- [39] J. Gaffney, D.A Green and C.E.M Pearce., " Binary versus real coding for genetic algorithms : a falce dichotomy",. *Anziam journal*. vol 51. pp. 347-359. 2010.
- [40] F. Baltaglia., " evolutionary computation methode and theirre application in statis-tics",. *Statistica*, amno LXIX, vol. 1. pp. 2-3.2009.
- [41] K. Balcombi., " Model selection using information criteria and genetic algorithms",. *Computational economics*, vol 25. pp. 207-228. 2005.
- [42] R. Baragona, and F. Battacilia., " Evolutionary computing in statistical data analy-sis",. *Fondation of computational intellgence*, vol. 3. pp. 347-386. 2009.
- [43] R. Baragona, F. Battacilia and C. Galzini., " Genetic algorithms for identification of addotive and inovation , outlier in time series",. *Computational statistics and data analysis*, vol. 37. pp. 277-295. 2001.

- [44] M. Chalerjee, M. Laudato and L.A Lynch., " Genetic algorithms and their statistical application an introduction"., Computational statistics and data analysis, vol. 22. pp. 633-651. 1996.
- [45] D. Preve., " Essays on time series analysis, with application to financial economics"., ISSN 1652-9030. Acta university. 2008.
- [46] P.D Feigin and S.I Resnick., " Estimation for autoregressive process with positive innovation, communications in statistics stochastic model"., 1992.
- [47] D. Harvy, S. Leybourn and P. Newbold., " Testing the equality of prediction mean squared errors"., International journal of forecasting, vol. 13. pp. 281-291. 1997.
- [48] P. Corter, M. Radian and J. Nevo., " Genetic and evolutionary algorithms for time series forecasting"., Portugeuse foundation of science and technologie. 1998.
- [49] G. Box and G. Jenkins., " Time series analysis forecasting and controle"., Holden dag, San Francisco, USA. 1976.
- [50] C. Chai, C. Chuek, M. DP and T. Huat., " Time series modelling and foecasting using genetic algorithms"., In proceedings of the first international conference on knowledge based intellegent electronic systems, vol. 1. pp. 260-268. 1995.
- [51] C. Huang and H. Yang., " A time series approach to short term load forecasting throught evolutionary pogramming structures"., In proceeding of EMPD. pp. 583-588. 1995.
- [52] G. Schwarz., " Estimating the dimenssion of model"., The annal of statistics, vol. 6. pp.461-480. 1978.
- [53] K. Smail Ibrahim and B. Eman., " An algoithms forecasting time series using wavelet"., IJCSI, vol. 11, no. 1. 2014.
- [54] R. Mechgoug., " La prediction de series temporelles utilisant les pardigme de soft computing"., Thèse de doctorat, 2013.
- [55] G. Melard., " Methodes de prevision à court terme"., Editions de l'université de Bruxelles. 1990.
- [56] A.S Weigened and N.A Gershen., " Time series prediction forecasting the future and under steding the past "., Eddission wesley. 1993.
- [57] A.C. Harvy., " Forecasting time series models and Kalman filter"., Cambridge univer-sity press. 1989.
- [58] C. Stern, W. Wright, S. Makridakis and Z. Hyndman., " Forecasting methods and applications"., John Wiley and Bons. 1998.
- [59] D.R. Brillinger., " Time series data analysis and theory"., Holt Rin hart and Wiston, New York. 1975.
- [60] J.E. Baker., " Adaptative selection methods for genetic algorithms"., Proceeding of first international conference on genetic algorithms. pp. 101-111. 1985.

- [61] W.S. Celeveland., " The inverse autocorrelations of a time series and their applications"., *Technometrics*, vol. 14. pp. 277-298. 1972.
- [62] O. Deweck., " A basic introduction to genetic algorithms"., *ESD.77/16.888*. Springer. 2010.
- [63] A. Sofiane, L. Baron and M. Blazinski., " Real/ binary like coded versus binary coded genetic algorithms to automatically generate fuzzy knowledge bases: a comparative study"., *Engineering applications of artificial intelligence*, vol. 77. pp. 313-325. 2004.
- [64] A. Shaifali, G. Richa and G. Puneat., " A review paper on different encoding schemes used in genetic algorithms"., *ISSN: 2277*; vol. 4. no.1. 2014.
- [65] V. Thomas Y. Murat., " Presentations des algorithmes genetiques de leurs applications en economie"., *Université de Nante*, vol.1. no.2. 2001.
- [66] Z. Michalewicz, C. Janikow and J. Knaweczy., " A modified genetic algorithms for optimal control problems"., *Computers and mathematics with application*, vol. 23. PP. 83-94. 1992.
- [67] F.K. Oduro Gyanah, E. Harris and K.F. Darkwah., " Sarima time series model application to microwave transmission of Yeji-Salaga(Ghana) line of sigt link"., *IJ of application science and technology*, vol. 2. no.09. 2012.
- [68] R.F. Engl., " The use of ARCH and GARCH models in applied economics"., *Journal of economic perspective*, vol. 15. no. 4. pp. 157-168. 2001.
- [69] I. Loganathan, I. Nanthakumer and Y. Ibrahim., " Forecasting international of tourism demand in Malaysia using Box-Jenkins SARIMA application"., *South Asian journal of tourism and heritage*, vol. 3.pp.50-60. 2010.
- [70] A. Frifa, E. Yamo ah, I.I. Bashiru and A. Saeed Karim., " SARIMA modelling and forecasting of monthly rainfall in the Brong Ahafo of Chana"., *World environment*, vol.6. pp. 1-9. 2016.
- [71] Suhartono., " Time Series Forecasting by using Seasonal Autoregressive Integrated Moving Average: Subset, Multiplicative or Additive Model"., *Journal of Mathematics and Statistics* 7 (1): 20-27, 2011.
- [72] M. Gerolimetto., " ARIMA and SARIMA models"., *Dip. di Statistica University a Ca'Foscari Venezia*. 2010.
- [73] A. Ratnadip and R.K. Agrawal., " An introductory study on time series modeling and forecasting"., *LAP, Lambert Academic Publishing, Germany*, 2013.
- [74] J. Lee., " Univariate time series modeling and forecasting (Box-jenkins method)"., *Econ.413*. University of Alabama. 2009.
- [75] J.M. Kihoro, R.O. Otieno and C. Wafula., " Seasonal time series modeling and forecasting :a comparison study of ARIMA and ANN models"., *AJST*. vol. 5. no.2. pp. 41-49. 2004.

- [76] R. Parrelli., " Introduction to ARCH and GARCH models"., Econ University of Illinois. 2001.
- [77] G. Margherita., " ARIMA and SARIMA models", at:www.dst.unive.it margherita/TSLecturenotes6.pdf (accessed 14-August, 2015).
- [78] K. R. Peltokangas and S. Sorsa., " Real coded genetic algorithms and nonlinear parameter identification"., University of Oula .CEL. no.34. 2008.
- [79] F. Herre ra, M. Lozano and J. L. Verdelpy., " Tackling real coded genetic algorithms: operators and tools for behavioural analysis"., Artificil intelegent review, vol.12. pp.265-319. 1998.
- [80] M. Jean and D. Nicolas., " Genetic algorithms"., 2005.
- [81] C.L. Lin, H. Huag and W. Chih., " Structure specified real coded genetic algorithms with application"., TMRF-eBOOK, vol.1. pp. 160-187. 2010.
- [82] K. Imtiaie, S. Yang and C. He., " A directed mutation operators for real coded genetic algorithms"., Eds.6024. pp. 491-500. 2010.
- [83] Z.Q. Chen and R.L. Wang., " Two efficient real coded genetic algorithms for real parameter optimization"., ICIC international., vol.7. pp. 4871-4883. 2011.
- [84] S. Hu., "Akaike information criterion"., North carolina state university. 2007.
- [85] J.J. Daudin, C. Duby, S. Robin and P. Trecourt., " Analyse des series chronologiques "., INA-PG mathematiques. 2. 1996.
- [86] P.J. Brockwell and R.A. Davis., " Time series theory and methods"., Springer serie in statistics. 1986.
- [87] K. Kiukko., " Model averaging with akaike weights"., Turku PET center modelling. 2007.
- [88] N. Robert., " The mathematical structure of ARIMA models"., Fuqua school of business Duke university. 2014.
- [89] L.S Richard., " Time series"., university of north carolina, version 1. 1999.
- [90] G.E.P. Box and D.A. Pierce., " Distriution of residual autocorrelation in autoregressive integrated moving average time series models"., Journal of the Americanstatistical association , vol. 65. pp. 1509-1526. 1970.
- [91] A. K. Bhuniaa, P. Palb, S. Chattopadhyayc, B. K. Medyad., " On Genetic Operators for Unconstrained Optimization Problems"., Advanced Modeling and Optimization, Volume 12, Number 2, 2010.
- [92] X. Zhao and X.-S. Gao., "Evolutionary Programming Based on Non-Uniform Mutation", MM Research Preprints Academia Sinica No. 23. pp. 352-374. December 2004.

General conclusion and perspective

The main objective of time series analysis is to know the relation between observation for choosing an optimal model to forecast the future. In this thesis, we have presented and exploit one of robustness methods for optimization, is the genetic algorithms with his types BCGA and RCGA, for modeling the time series and performing the forecasting and comparing them too with Box - Jenkins method. Application of this method to linear model ARIMA and SARIMA models in the third and fourth chapter improve the model's construction and the forecast with giving us a good models comparing with the classical method. In the fourth chapter, when we let the variable in their nature called real encoded requires some various crossover and mutation operators more than BCGA operators. A new evolutionary programming algorithm crossover operators based on the non-uniform mutation(NUM) and modification in errors are proposed. A highly comparative framework for time-series analysis was developed and its broad scientific utility demonstrated. GAs with his types is generally faster and more robust then Box - Jenkins method.