UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA

DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES Y AUTOMÁTICA

TESIS DOCTORAL

Operando en el mercado de valores:
análisis financieros híbridos y computación evolutiva

Trading the stock market:
hybrid financial analyses and evolutionary computation

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Iván Contreras Fernández-Dávila

Directores

José Ignacio Hidalgo Pérez
Laura Marta Núñez Letamendia

Madrid, 2014

OPERANDO EN EL MERCADO DE VALORES: ANÁLISIS FINANCIEROS HÍBRIDOS Y COMPUTACIÓN EVOLUTIVA

TRADING THE STOCK MARKET: HYBRID FINANCIAL ANALYSES AND EVOLUTIONARY COMPUTATION

IVÁN CONTRERAS FERNÁNDEZ - DÁVILA

TESIS DOCTORAL
UNIVERSIDAD COMPLUTENSE DE MADRID

Mayo de 2014

Directores:

Jose Ignacio Hidalgo Perez
Laura Marta Núñez Letamendia

“What we learn from history is that people don’t learn from history”
– Warren Buffett

Resumen en castellano

Esta tesis presenta la implementación de un innovador sistema de comercio automatizado que utiliza tres importantes análisis para determinar lugares y momentos de inversión. Para ello, este trabajo profundiza en sistemas automáticos de comercio y estudia series temporales de precios históricos pertenecientes a empresas que cotizan en el mercado bursátil. Estudiamos y clasifcamos las series temporales mediante el uso de una novedosa metodología basada en compresores de software. Este nuevo enfoque permite un estudio teórico de la formación de precios que demuestra resultados de divergencia entre precios reales de mercado y precios modelados mediante paseos aleatorios, apoyando así el desarrollo de modelos predictivos basados en el análisis de patrones históricos como los descritos en este documento. Además, esta metodología nos permite estudiar el comportamiento de series temporales de precios históricos en distintos sectores industriales mediante la búsqueda de patrones en empresas pertenecientes al mismo sector. Los resultados muestran agrupaciones que indican tendencias de mercado compartidas y ,por tanto, señalan que la inclusión de un análisis industrial puede reportar ventajas en la toma de decisiones de inversión.

Comprobada la factibilidad de un sistema de predicción basado en series temporales y demostrada la existencia de tendencias macroeconómicas en las diferentes industrias, proponemos el desarrollo del sistema completo a través de diferentes etapas. Iterativamente y mediante varias aproximaciones, testeamos y analizamos las piezas que componen el sistema final. Las primeras fases describen un sistema de comercio automatizado, basado en análisis técnico y fundamental de empresas, que presenta altos rendimientos y reduce el riesgo de pérdidas. El sistema utiliza un motor de optimización guiado por una versión modificada de un algoritmo genético el la que presentamos operadores innovadores que proporcionan mecanismos para evitar una convergencia prematura del algoritmo y mejorar los resultados de rendimiento finales. Utilizando este mismo sistema de comercio automático proponemos técnicas de optimización novedosas en relación a uno de los problemas más característicos de estos sistemas, el tiempo de ejecución. Presentamos la paralelización del sistema de comercio automatizado mediante dos técnicas de computación paralela, computación distribuida y procesamiento gráfico. Ambas arquitecturas presentan aceleraciones elevadas alcanzando los x50 y x256 respectivamente. Estápas posteriores presentan un cambio de metodologia de optimización, algoritmos genéticos por evolución gramatical, que nos permite comparar ambas estrategias e implementar características más avanzadas como reglas más complejas o la auto-generación de nuevos indicadores técnicos. Testearemos, con datos financieros recientes, varios sistemas de comercio basados en diferentes funciones de aptitud, incluyendo una innovadora versión multi-objetivo, que nos permitirán analizar las ventajas de cada función de aptitud. Finalmente, describimos y testeamos la metodología del sistema de comercio automatizado basado en una doble capa de gramáticas evolutivas y que combina un análisis técnico, fundamental y macroeconómico en un análisis top-down híbrido. Los resultados obtenidos muestran rendimientos medios del 30% con muy pocas operaciones de perdidas.

Abstract

This thesis concerns to the implementation of a complex and pioneering automated trading system which uses three critical analysis to determine time-decisions and portfolios for investments. To this end, this work delves into automated trading systems and studies time series of historical prices related to companies listed in stock markets. Time series are studied using a novel methodology based on clusterings by software compressors. This new approach allows a theoretical study of price formation which shows results of divergence between market prices and prices modelled by random walks, thus supporting the implementation of predictive models based on the analysis of historical patterns. Furthermore, this methodology also provides us the tool to study behaviours of time series of historical prices from different industrial sectors seeking patterns among companies in the same industry. Results show clusters of companies pointing out market trends among companies developing similar activities, and suggesting a macroeconomic analysis to take advantage of investment decisions.

Tested the feasibility of prediction systems based on analyses related to time series of historical prices and tested the existence of macroeconomic trends in the industries, we propose the implementation of a hybrid automated trading system through several stages which iteratively describe and test the components of the final system. In the early stages, we implement an automated trading system based on technical and fundamental analysis of companies, it presents high returns and reducing losses. The implementation uses a methodology guided by a modified version of a genetic algorithm which presents novel genetic operators avoiding the premature convergence and improving final results. Using the same automated trading system we propose novel optimization techniques related to one of the characteristic problems of these systems: the execution time. We present the parallelisation of the system using two parallel computing techniques, first using distributed computation and, second, implementing a version for graphics processors. Both architectures achieve high speed-ups, reaching 50x and 256x respectively, thus, they present the necessary speed-ups required by systems analysing huge amount of financial data. Subsequent stages present a transformation in the methodology, genetic algorithms for grammatical evolution, which allows us to compare the two evolutionary strategies and to implement more advanced features such as more complex rules or the self-generation of new technical indicators. In this context, we describe several automated trading system versions guided by different fitness functions, including an innovative multi-objective version that we test with recent financial data analysing the advantages of each fitness function. Finally, we describe and test the methodology of an automated trading system based on a double layer of grammatical evolution combining technical, fundamental and macroeconomic analysis on a hybrid top-down analysis. The results show average returns of 30% with low number of negative operations.

Contents

Index i

List of Figures iii

List of Tables vii

1 Introduction 1

1.1 Motivation 1

1.2 The Development of an Automated Trading System 4

1.3 Objectives and Contributions 5

1.4 Organization 7

2 The Stock Market: Prediction models 9

2.1 Behavioural Models 10

2.2 Introduction to the Economic Analysis 13

2.3 Technical Analysis 14

2.4 Fundamental Analysis 24

2.5 Automated Trading Systems 32

3 Evolutionary Algorithms 34

3.1 Genetic Algorithms 36

3.2 Grammatical Evolution 39

3.3 Fitness Functions 44

4 The Predictability of Stock Prices 48

4.1 The Random Walk Hypothesis 48

4.2 Stock Market and Randomness 50

4.3 Methodology 52

4.4 Experimental results: Real and modelled series clusterings 55

4.5 Undercover Patterns Among Industries 58

4.6 Experimental Results: Clusterings of Price Series by Sectors 61

5 An Automated Trading System Based on a Genetic Algorithm 66
5.1 Trading System Based on Genetic Algorithms: Implementation 67
5.2 Parallelisation on a Grid Architecture 78
5.3 Parallelization on General-Purpose Computing on Graphics Processing Units 86
5.4 Grid Computing Versus GPU Computing 98

6 Hybrid Automated Trading System based on Grammatical Evolution 102
6.1 Trading the Stock Market: Automated Trading Systems Based on Grammatical Evolution 103
6.2 Experimental Results: Automated Trading Systems Based on GE 111
6.3 Trading the Stock Market: Macroeconomic, Fundamental and Technical Analysis 117

7 Final Conclusions and Future Research 127
7.1 Future Research 129
7.2 Related Publications and merits 130
7.3 Research Projects and Grants 135

8 Resumen en Español 136
8.1 Introducción: Mercados y metodologías 136
8.2 Predictibilidad de Mercados: Patrones Ocultos en Series de Precios Históricos 139
8.3 Patrones ocultos entre industrias 145
8.4 Operando en el Mercado de Valores: Algoritmos Genéticos y Análisis Financiero Híbrido 148
8.5 Paralelización mediante un Grid de computación 154
8.6 Paralelización mediante tarjetas gráficas 156
8.7 Operando en el Mercado de Valores: Gramáticas Evolutivas y Funciones de Aptitud 161
8.8 Operando en el Mercado: Análisis Macroeconómico, Fundamental y Técnico 168
8.9 Conclusiones 171

Bibliography 177

List of Figures

2.1 Chart of the Moving Average Crossover (MAC). Top shows prices of a company together with the signals triggered by the indicator(red: sell and green: buy) in a particular period (8 months). Bottom shows the related SMAs (black: short SMA and grey: long SMA) 19

2.2 Chart of the Moving Average Convergence Divergence (MACD). Top shows prices of a company together with the signals triggered by the indicator(red: sell and green: buy) in a particular period (8 months). Bottom shows the crosses of the difference between two EMAs (black) and the signal EMA (gray) 20

2.3 Chart of the Relative Strength Index (RSI). Top shows prices of a company together with the signals tiggered by the indicator(red: sell and green: buy) in a particular period (17 months). Bottom shows the values of RSI (black) crossing with the thresholds, lines of overbought (red) and of oversell (green) 22

2.4 Chart of the Volume Price Confirmation Indicator (VPCI). Top shows prices of a company together with the signals and trend-lines over a period of 14 months. Signals are triggered by the indicator(red: sell and green: buy). Trend-lines are introduced each two months (red : negative trend and green : positive trend). Bottom shows the trend-lines and values of the volumes related to the same company and period. 23

2.5 Chart of the Supports and Resistances indicator (SR). Figure shows prices of a company together with the lines of supports (blue) and resistances (red) over a period of 22 months. Signals are triggered when the price itself break a resistance (buy) or a support (sell) lines. 25

3.1 Typical genetic algorithm work flow 37

3.2 Basic grammar to build a simple trading strategy 40

3.3 EAs chromosomes : Main differences between GA and GE 41

4.1 Random series versus price series: Series 1 in blue shows daily prices of A2A S.P.A (Italy) over the period 2002 - 2011. The remaining series are modelled price time series by the Geometric Brownian Motion 51
4.2 Random series versus real series: Series 1 in blue shows monthly prices of Repsol (Spain) over the period 2002 - 2011. The remaining series are modelled price time series by the Geometric Brownian Motion 52
4.3 Random versus real price time series: Clustering of the monthly prices of 80 companies listed in Europe over the period 2001-2012, and 80 price time series modelled by the GBM 56
4.4 Random versus real time price series: Clustering of the daily prices of 80 companies listed in Europe over the period 2001-2012, and 80 price time series modeled by the GBM 57
4.5 Clusters among companies operating in different economic activities. 62
4.6 Clusters among companies operating in similar economic activities. 64

5.1 General work-flow of the GAwFO in a generation (cycle) N: evaluation, elitism, crossover, mutation and filling operator 69
5.2 Genetic encoding: Chromosome example for the GAwFO 70
5.3 Operation of crossover with dominance 72
5.4 Experimental Results for 85 companies listed in the S&P 500. Average of the accumulated return for the B&H =17.47 and for the ATS based on GAwFO (average of 20 runs)=14.31. 73
5.5 Convergency of the algorithm (profit of company ID1356) for GAwFO and sGA for 100 generations and several amounts of individuals 75
5.6 In-sample returns (company ID 1356) for the GA and the GAwFO for 100 and 500 generations and several amounts of individuals (X axis) 76
5.7 Execution time (Y axis) for the GA and the GAwFO for 100 and 500 generations and several amounts of individuals (X axis) 77
5.8 Profit (Y axis is the in-sample return) and execution time (the X axis is the delimited time) for the GA and the GAwFO for 500 generations and 500 individuals. 77
5.9 The client-server model server work-flow of Boinc 80
5.10 Execution times in the grid (y- axis) for 500 generations and different number of individuals (x- axis) 85

5.11 Speed-up in the grid 86
5.12 Execution time analysis of the genetic algorithm in the CPU. (One company, 5000 individuals, 500 generations). Total Time = 648 seconds. 87
5.13 General Jacket architecture 90
5.14 Genetic algorithm execution in the GPU. It has been run for a company with 5000 individuals, 500 generation. Total Time= 66 seconds 93
5.15 Execution times (y- axis) for 500 generations and different number of individuals (x- axis) 94
5.16 Speed-up of the 570 GTX over different CPUs (y-axis) with variable number of individuals (x-axis) and 500 generations 96
5.17 Execution of the GA in the GPU. It is run for 10 companies, with 50000 individuals, 500 generations and tournament selection algorithm. Total time =109 seconds, partial time for company =10.9 seconds 97
5.18 Execution times comparative (y- axis) for 500 generations and different number of individuals (x- axis) 99
5.19 Speed-up on the different platforms (y-axis)with variable number of individuals (x-axis) and 500 generations 100

6.1 Fragment of the grammar used in our ATS GE-based. The grammar has the aim to build programs providing investment signals on a specific period. . . . 106
6.2 Flow of a NSGAII. 110
6.3 Returns of 43 spanish companies over 2012 (recent data) and 4 companies of the S&P500 over 2004 (old data) 112
6.4 Returns in 2012 of the different fitness versions of the ATS based in GE for 36 European companies 114
6.5 Returns of the multi-objective versions of the ATS based in GE for 36 European companies. 115
6.6 Returns of the multi-objective and mono-objective versions of the ATSs based on GE for 36 European companies divided in 6 sectors. 117
6.7 The top-dow methodology 118
6.8 The double layer GE methodology 120
6.9 Fragment of the external grammar used in our ATS GE-based. The grammar has the aim to build programs providing a portfolio of the most promising companies. 122

6.10 Returns of 15 operations in 2013 by the ATS performing a macroeconomic, fundamental and technical analysis. The ATS performs 1 negative investment versus 13 of the Buy and Hold 124

8.1 En azul serie de precios históricos diarios de A2A (2001-2010). El resto de las series modeladas con GBM 142

8.2 En azul serie de precios históricos mensuales de Repsol (2001-2010). El resto de las series modeladas con GBM 142

8.3 Clustering de series mensuales de 80 compañias listadas en Europa entre el 2001 y 1012 y 80 series modeladas mediante la GBM 143

8.4 Clustering de series diarias de 80 compañias listadas en Europa entre el 2001 y 1012 y 80 series modeladas mediante la GBM 144

8.5 Agrupaciones entre empresas pertenecientes a diferentes industrias 146

8.6 Rendimientos medios del SAC y B&H para 85 empresas del S&P 500. B&H consigue un 17.47% con 24/61 operaciones negativas/positivas y el SAC un 14.31% con 8/77 operaciones negativas/positivas 150

8.7 Rendimientos (eje Y) para el sGA y el GAwFO para 100/500 generaciones y un numero en incremento de individuos (eje X) 152

8.8 Tiempos de ejecución (eje Y) para el sGA y el GAwFO para 100/500 generaciones y un numero en incremento de individuos (eje X) 152

8.9 Tiempos de ejecucion en el grid (eje Y) para 500 generaciones y un numero de individuos creciente (eje X) 156

8.10 Tiempos de ejecución (eje-Y) para 500 generaciones y diferentes número de individuos (eje-X) 159

8.11 Rendimientos de 43 empresas españolas durante el 2012 (recent data) and 4 compañías del S&P500 en 2004 (old data) 163

8.12 Rendimientos de 36 compañías europeas en 2012 de diferentes funciones de aptitud del SAC basado en GE 165

8.13 Flujo de el algoritmo multiobjetivo NSGAII. 166

8.14 Rendimientos para el SAC multi-objetivo en 36 compañías europeas. 167

8.15 Metodología basada en una capadoble de GE 169

8.16 Rendimientos de 15 ejecuciones en 2013 por el SAC mediante un análisis fundamental, técnico y macroeconómico. Rendimientos medios de aproximadamente un 30% 170

List of Tables

2.1 Set of variables used in the valuation of the companies through its fundamentals 26

2.2 Set of variables used in economic analysis related to countries 26

2.3 Set of variables used in industrial analysis related to different sectoral activities 27

2.4 Macroeconomic variables used in this work (industrial and economic). Acronyms from AMECO database. 30

4.1 Percents of success and number of companies per cluster of Figure 8.5 63

4.2 Percents of success and number of companies per cluster of Figure 4.6 65

5.1 Chromosome codification: parameters, ranges and decodification process . . 71

5.2 Main characteristics of the Server used in the experiments 82

5.3 Main characteristics of the grid used in the experiments 83

5.4 CPU Architectures used for comparison 93

5.5 Main characteristics of the GPUs used in the experiments 93

6.1 Detailed returns of 45 companies in 2013 by the ATS performing a macroeconomic, fundamental and technical analysis. The average return reachs a 30.14% versus a -13.35% of a static strategy. The ATS performs 10 negative operations versus 39 of the B&H strategy. 125

8.1 Porcentajes de éxito de clasificación del Clustering de la Figura 8.5 147

Chapter 1

Introduction

The introduction of this thesis has the aim to answer usual questions that researchers should make themselves before starting a project of this characteristics. This section will answer to key factors as the location or the local and global importance of the studied subject, furthermore we will also point to the used methodologies and the applied architectures. We will also argue the main motivations to continue the research in the studied matter and the contributions provided. Finally, this section presents the summary of the rest of the content of the thesis.

1.1 Motivation

At the end of September 2013 the World Federation of Exchanges, a trade association of 62 publicly regulated stock market exchanges around the world (www.world-exchanges.org/), approximates the total market capitalization of the major equity markets of the world in 60.7 trillion of dollars. This figure gives us an idea of the critical role played by stock markets and an insight about the reached importance of the development of trading systems able to operate successfully in the markets. Since the first approaches to trading systems some decades ago, the development and research in this area has been increased explosively.

Stock markets are complex systems depending on countless number of variables where a small change can dramatically change the movements of a market asset [37]. Furthermore, operations of traders related to the stock market depend on the sentiment of thousands of others traders, which in turn depends on their age, economical or social situation, nerves, hormones, knowledge, background, etc. This large number of factors has frequently led to suggest that stock markets are unpredictable systems or even random models. The search for predictive models (investment rules for the market based on technical, fundamental or macroeconomic analysis) together with the attempt to understand the behavioural models of prices formation in the market assets (e.g the random walk or the adaptive market hypothesis) have established one of the most important path throughout the history of research in topics related to finances.

This thesis will delves into the topics mentioned in the previous paragraph: behaviour models and prediction models. The effectiveness of predictive systems is directly related with the behaviour model which determines the movements performing in the markets. The close correlation of both subjects push us to analyse the feasibility of a predictive system able to obtain good performances before we develop it. Nowadays, behavioural models are still a matter of disputes and arguments among the formulation of different theories. With the aim to justify the development of a prediction system in this thesis we will argue against the random prices in the stock market and we also analyse the behaviours of different market sectors (industrial sectors) to motivate and justify the inclusion of different macroeconomic variables into the predictive system

The growing availability of data for financial markets and companies, together with the increasing complexity of socio-economic and financial environment (at present tragically illustrated by the financial crisis), makes it more difficult the decision making process for real time investments in stock markets. Complexity in this sense derives from the difficulty of the assets valuation process due that each type of asset requires addressing its valuation with a separate approach to be able to modelling the factors affecting its performance in order to anticipate its risk adjusted return. The huge number of potential interrelated factors and their changing time patterns, affecting financial assets, make the investment process remains a challenge.

We consider that a trading system should be profitable to the financial investments due to the complexity that the market provides. First at all, the financial investments are influenced by a great amount of factors of different sources. Financial investments are affected by factors as, government policies, natural factors, international trade, market sentiment, political factors etc. Thus, it is very complex to follow a successful flow of information and later conclude the consequences that the information implies. Secondly, the traders are affected by the called behavioural finances. Brokers or practitioners in general could be affected by human emotions, so their behaviour in the stock market became not objective. The high pressure induced by handling a large volume of money is the main reason, which is able to trigger loss aversion, overconfidence, overreaction, and other behavioural biases.

The information technology sector (computation power, advanced algorithms, high-level programming languages, etc.) has developed an unstoppable growth in the last decades. The importance achieved by the information technology has rebounded in the great majority of areas. Thus IT has become an important component of new capital investments and economists in each corner of the world search into the computers the best hope for a sustainable increases in economic growth rates.

The development of IT, together with the telecommunications, has promoted the emergence of global processes that without it would have been inconceivable. The globalization is a wide process with an economic, technological, social and cultural character consisting of increased communication and interdependence among countries of the world uniting their markets, societies and cultures. The globalization is symbolized for instance by the tremendous development of the "hedge funds industry", a class of funds which invest in any kind of assets around the World (stocks, indexes, bonds, commodities, currencies, etc.) This new level of interconnectivity plays out in our financial markets where problems in one market have inescapable and often unpredictable effects on the rest of the markets worldwide.

We can summarize that in the past years interest in the trading systems has spread due to the next factors:

- (i) the explosive amount of information available for companies and markets.
- (ii) the progress of IT, mainly the inexpensive computing power and the advanced algorithms.
- (iii) the expansion of the globalization process.
- (iv) the attempt to avoid the psychological aspects that biases the investment process.

1.2 The Development of an Automated Trading System

Mechanical or automated trading systems are predictive systems based on rules that use market, business or macroeconomic information embedded in algorithms that look for the best combination of these rules to drive the stock trades in an attempt of obtaining the maximum possible return for a period. Finding optimal series of investment decision involves an inspection of the related complex search spaces. The complexity of the search space depends on the amount of analysed parameters. The number of combinations grows explosively forming a search space of increasing complexity, therefore, the more parameters are used to invest in the stock market, the more number of possible combinations should be analysed.

The amount of the analysed parameters is not the only point affecting to the complexity of the trading systems and over the time more features have been added to the the trading engines increasing the complexity. Thus, trading systems have evolved from very simple *If-then* algorithms to more sophisticated models that use methods like artificial intelligence, chaos theory, fractals, evolutionary algorithms, non-linear stochastic representations, econophysics models, etc., which are ultimately based on market, fundamental or macroeconomic data.

The described adverse environment points to the meta-heuristics as one of the best approaches to find good solutions in a short period of time. The application of meta-heuristics in trading systems has had a fast development in both the scientific and professional world. As a result of this development, many investment systems have benefited by the supporting of complex computation systems capable of analysing large amounts of data and information. The literature on the meta-heuristics is extensive ([3], [83], [54], [11],[107], [71], [72], [32], [12], [33], [1], [2], [99] among others), in this thesis we will focus on one of the best known and most successful branches within this type of methods: evolutionary meta-heuristics.

Evolutionary meta-heuristics, commonly referred as evolutionary algorithms are a set of search and optimisation methodologies inspired and based on principles and theories of the biology world. This procedures are characterized to emulate evolutionary behaviours of the nature basing on the survival of the best potential solution (individual) among a set of other solutions (population). In the academic literature a large spread of previous studies document the use of evolutionary algorithms to design and optimize automatic trading systems for the Stock Market. Furthermore, both the biggest investment firms and small companies has now begun to use the evolutionary algorithms to build automatic trading systems. In this work we focus the work in two well-know evolutionary algorithms, the genetic algorithms and the grammatical evolution. On one hand, the classic genetic algorithms are the most popular and proved methodologies among evolutionary algorithms, on the other hand grammar evolution is relative new methodology, which is considered more complex while more flexible solution. Both algorithms are optimisation methods which use operators inspired in the Darwinian principles of evolution, as the crossover, mutation or selection to evolve sets of solutions with the aim to find a good solution in a short time.

The heuristic methods simplify the search in the space of solutions allowing the quick search of a enough good solution, nevertheless one of the difficulties found by the researches and projects is the computational time required for training the trading system with daily data of stocks prices. This restriction is even more critical when we take into account that the majority of traders invest in an intra-day base. In this work we propose to combine the use of two different parallel computer architectures to speed up the functioning of a evolutionary algorithm used to design trading systems to invest in stocks. First, we have used a corporative grid, and later, a graphics device, both platforms have been used to obtain profits in computation time.

1.3 Objectives and Contributions

General objective

The final aim of this work is to build a novel automated trading system capable of analysing a high number of companies. The final solutions should provide as output a program optimised to earn high returns in two steps. First, the program solution selects the best portfolio to invest. Second, the program solution determine the specific times to perform investment operations.

Specific objectives

- Gather financial information on a group of companies, countries and sectors for building a robust database which works as a source of information for the automated trading system.

- Demonstrate the feasibility of a predictive system based on time series of historical data.
- Identify and analyse the usefulness of an automated trading system that assesses groups of companies linked to their industrial activities.
- Provide the system with a methodology to build and test novel solutions beyond the reach of a traditional system.
- Achieve high returns in periods of high risk and adverse conditions.
- Build a system able to respond to the stringent time requirements which are characteristics of investments in real time.

Throughout the different stages of development of the automated trading system, we present the next outlined contributions which support the thesis:

- We present results of divergence between the behaviour of the Random Walk Theory and the real exchanges.
- We find hidden patterns among companies performing similar economic activities.
- We test a novel genetic algorithm which works as optimisation methodology of the automated trading system and is capable to avoid the premature convergence.
- We propose several parallelisations an automated trading system for intra-daily investments which improve significantly the performance of the system in terms of computational time.
- We presents a automated trading system with one of the first multi-objective implementations based on Grammatical Evolution
- We present an automated trading system able to build and evaluate its own technical indicators.

- We use a novel hybrid analysis that for first time in the academic literature unifies a macroeconomic analysis of sectors and countries, a fundamental analysis of companies based on accounting information and a technical analysis of companies based in volume and prices.

1.4 Organization

Once we have present the introduction of the thesis, outlining the motivations, the development and the contributions, we show the organization and the content of the remaining chapters of the dissertation. Thus, the rest of the document is organised as follows.

In the early chapters of the thesis, Chapter 2 and Chapter 3, we are focused in the theoretical background of automated trading systems. First, Chapter 2 presents some critical points about the financial systems and its history, focusing in the Stock Markets and the correlated financial analysis: technical and fundamental. Furthermore, we take this section to define all financial variables used along the thesis. Second, Chapter 3 provides the basis of the main methodologies used in this work and which support the engine of our automated trading system: genetic algorithms and grammatical evolution. This chapter also presents the evaluation functions (fitness functions) that the system will use throughout the thesis.

Before beginning the development of the predictive system, Chapter 4 performs a series of preliminary analyses. First, we focus on the study of the behaviour of the stock market time series which have been target of controversial researches. Second we cope with the analysis of these time series by industrial sectors. Thus, the chapter provides the motivation to develop a prediction model based on historical time series of prices and supports the idea of including a macroeconomic analysis to determine the most profitable trends. The experiments are based on clustering methodologies which nowadays are used in spread of areas and are already considered a mature analysis tool. In both strands we cluster sets of historical time series of prices using a metric derived from the Kolmogorov complexity. The first strand discuss one of the postulated financial theories which defines a random behaviour of the market meaning that the prices flow randomly boycotting any prediction model. We propose a novel approach to test the existence of undercover common and uncommon patterns in stock prices by using only information of historical time series of prices. In the second strand, we use the same methodology to analyse the division by sectors of the Stock Market searching connexions into clusters of companies developing alike industrial activities, and we compare this connexions with clusters of sectors with similar activities.

In Chapter 5 we formulate the investment problem describing a novel automated trading system based on genetic algorithms. Chapter 5 introduces with detail the implementation of the Genetic Algorithm with the Filling Operator. This version of a genetic algorithm fits with our trading system allowing to avoid the premature convergence and improving the final results. Furthermore, Chapter 5 describes two parallel architectures, grid computing and graphical computing, and supplies experimental results showing different ways of combining the use of GAs and parallelisation systems. Parallelisation using the proposed techniques provide a substantial acceleration in the power and capacity of the GA search for this type of financial applications. On one hand, we present a parallelisation on BOINC an on the other hand we develop a lower level parallelisation with a framework based on CUDA. Finally we conclude the chapter facing both parallel architectures.

Chapter 6 presents the final stage of the thesis developing an automated trading system able to perform a top-down analysis mixing macroeconomic and microeconomic analyses. First the chapter introduces an automated trading system related with the microeconomic analysis. We implement and test novel characteristics as: the self-generation of indicators, the use of multi-strategies or the implementation of multi-objective optimisation based on grammatical evolution. Second, we introduce and describe a new layer of analysis related with macroeconomics. Thus we implement and test the whole system with an innovative analysis able to select the best companies and times to invest.

Although we will proportionate the necessary explanations in each chapter about its experiments, methodologies, figures and results; finally, we summarize and conclude in Chapter 7, where we show a compilation of the most important conclusions of the dissertation. Furthermore, we argue some points related to the possible line of future research.

Chapter 2

The Stock Market: Prediction models

In the seventeenth century the city of Amsterdam hosted what we consider the first floors of history, the first building dedicated solely to the sale of shares. It was the financial centre of the world in the early seventeenth century, where the first bank was born and all kinds of financial products such as sovereign debt or futures contracts, even where the first stock market crash in the history took place. Since the origins of the stock market until nowadays the stock market has increased its weight in the world until to reach the glamorous Wall Street considering vertex of the global finances.

Money looks to rule the world becoming a very important piece in our present civilization. Money flows in several ways, however a great part of the total amount of money of the world belongs to the stock markets. At the end of September 2013, the Paris-based World Federation of Exchanges (http://www.world-exchanges.org/), an association of 58 publicly regulated stock market exchanges around the world, approximate the total market capitalization of the major equity markets of the world in 60.7 trillion of dollars. We could summarize the definition of the Stock Market as a relationship-based system of two types of agents in which on one hand, companies needing of large financing come to the stock market to issue shares or bonds and on the other hand, savers (both institutions and particulars) want to get returns on their surplus deciding to purchase products issued by companies in the stock market. The stock market become an instrument of financing for companies and of investing for savers. Thus the stock market facilitate the mobility of wealth becoming in a leading indicator of the economy.

The stock markets are complex systems depending on countless number of variables where a small change can dramatically change the movements of a market asset. Furthermore, operation of traders related to the stock market depend on the sentiment of thousands of others traders, which in turn depends on their age, economical or social situation, nerves, hormones, experiences, knowledge, etc. Often this tremendous amount of factors has driven to think that the Stock Market is an unpredictable system or even a random model. The search of predictive models (investment rules for the market based on technical, fundamental or macroeconomic analysis) together with the search of behavioural models of the prices formation in the market assets (i.e the random walk or the adaptive market hypothesis) have established one of the most important path throughout history of research in the topics related to finances.

This thesis will cover part of the two topics mentioned in the previous paragraph: behaviour models and prediction models. The effectiveness of the predictive systems depends directly on the behaviour model which determines the movements performing in the markets. The close correlation of both subjects push us to analyse the feasibility of a predictive system able to obtain good performances before we develop it. Nowadays, behavioural models are still a matter of disputes and arguments among the formulation of different theories. With the aim to justify the development of a prediction system in this thesis we will argue against the random prices in the stock market.

2.1 Behavioural Models

After decades of research and numerous papers and books published about predictability in stock returns, both theory and empirical evidence about this topic remain ambiguous, and not conclusive. Since small variations in expected returns can produce large and economically significant changes in asset values, the interest in stock returns predictability will remain for always. Along the literature of financial markets arose several trends or different hypothesis about the predictability of the markets. In this document, we only take a brief look to the pillars of the financial market theories, *the Random Walk* and *the Efficient Market Hypothesis.* Furthermore, we introduce the main financial theory supporting this thesis: *the Adaptive Market Hypothesis.*

As we previously mentioned the Stock Market is a non lineal system based in countless variables. That is why we can find many practitioners and researchers in the literature which support the randomness of the Stock Market. The theory supporting this idea was called the Random Walk of the prices. The Random Walk theory is a financial concept summarising that stock prices do not follow any particular path. The first serious hypothesis about the random movements of stock prices was proposed in [56] and the term was popularised in [75]. The theory states that the fluctuations of stock prices are random, hence there is no correlation between the last price and the future price. This definition implies an unpredictable future movement of an action. According to the Random Walk theory, stock prices are random making it impossible to consistently outperform market averages by using any investment strategy.

In conjunction with the Random Walk, the academic literature on this topic has been also developed around the efficient markets hypothesis (EMH) framework proposed in the 60s, which has two main propositions, the weak form and the semi-strong form. The weak form of the EMH claims that prices fully reflect the information implicit in the sequence of past prices, in other words, the examination of the sequence of historical stock prices is worthless when it comes to forecasting future prices. The random walk hypothesis that states that stock market prices evolve randomly and thus cannot be predicted is closely related with the weak form of the EMH. The semi-strong form of the EMH asserts that prices reflect all relevant information that is publicly available related to political, macroeconomic, industry, or firm-specific variables. Weak form information considers only the information contained in past stock prices or volume traded whereas semi-strong form refers to market-wide information available[1] .

The financial investments are influenced by a great amount of factors of different sources. Financial investments are affected by factors as, government policies, natural factors, international trade, market sentiment, political factors etc. Thus, it is very complex to follow a successful flow of information and later conclude the consequences that the information implies. Secondly, the traders are affected by the called behavioural finances. Brokers or practitioners in general could be affected by human emotions, so their behaviour in the stock market became not objective. The high pressure induced by handling a large volume

[1] There is a third form in which the EMH is stated, the strong-form, in which share prices reflect all information, public and private. However due to the existence of legal barriers to private information becoming public there is agreement among academics that strong-form efficiency is not hold since they recognize that the use of private information can generate excess returns.

of money is the main reason, which is able to trigger remorse, loss aversion, overconfidence, overreaction, mental accounting, herd behaviour, and other behavioural biases. The so-called behavioural finances become a strong trend of thought and over the year 2008 just after the crack, with a global economy and financial system with a large amount of problems, the EMH is considered quite in doubt. To a large extent, the financial system itself is the cause of its own problems. In this context it is not easy to think that these markets play the role of passive markets, tending towards equilibrium and reflecting in real time conditions and prices. Markets are active, follow its own objectives, and have a huge power in the governments, global economy and in the history itself.

Although behavioural finance had been gained support in recent years, it is not without its critiques. The most notable critic of behavioural finance is Eugene Fama, the founder of market efficiency theory and recently awarded with the Novel Prize. Professor Fama suggests that even though there are some anomalies that cannot be explained by modern financial theory, market efficiency should not be totally abandoned in favour of behavioural finance. In fact, he notes that many of the anomalies found in conventional theories could be considered short-term chance events that are eventually corrected over time. In his 1998 paper, entitled “Market Efficiency, Long-Term Returns and Behavioural Finance”, Fama argues that many of the findings in behavioural finance appear to contradict each other, and that all in all, behavioural finance itself appears to be a collection of anomalies that can be explained by market efficiency.

It is in the disputed environment of these two trends of thought where born the Adaptive Market Hypothesis (AMH) in an attempt to reconcile economic theories based on the EMH with behavioural economics.

The novel theory about the behaviour of the markets was proposed by Andrew Lo in [69] claiming the AMH "as a new version of the EMH, derived from evolutionary principles". Lo is a well-known researcher and lecture in finances at the MIT (Massachusetts Institute Technologies). He has published several works with quite repercussion in this topic [i.e [67], [70], [69]]. The strongest point of the AMH is that it can be combined with the rationalist theories. AMH is according to the balance of the behavioural economy and the market efficiency, however it supports that the classical theory cannot reflect the behaviour of the market in every case. Within the framework of the AMH, investment strategies evolve over time as investors learn which strategies work better in different circumstances and gradually implement them. Although it is true that in doing this profitable strategies will progressively disappear, while new price patterns will emerge and new strategies will be developed to exploit them. Professor Lo uses evidences and conclusions of another researchers in the field of Behavioural Economics, which are related with D.Kahneman (Nobel Prize in 2002), G.Caginalp (Editor of the Journal of Behavioural Finance), D.Ariely and C.Camerer among others. They have tested empirically that the financial decisions, finally human decisions, have a heuristic component. This heuristic component makes markets deviate from the financial theory.

Thus, the AMH is proposed as an alternative theory to the EMH. The features criticised as probed deviations in the EMH at the insight of the AMH would be the follow human reactions studied by the behavioural finances (overconfidence, overreaction, loss aversion, remorse, etc.). The AMH is based on Darwin masterpiece "On the Origin of Species" [28], into the paradigm of the ecology and the evolution of the species, as Lo noted: "Economics is a uniquely human endeavour and, as such, should be understood in the broader context of competition, mutation, and natural selection , in other words, evolution".Using this metaphor, financial markets become an instance of an ecosystem where different tactics and techniques are used by practitioners. Competition is very high, so that only those best adapted survive. The final target is the perpetuation of the species by transferring their genetic material to achieve the highest amount of return in a long term.

2.2 Introduction to the Economic Analysis

The behaviour of the stock market always has been very complicated to forecast, even more nowadays. One of the most exciting topics among the financial issues is the investments in the stock markets. History can provide just a set of market events to analyse; events attached to a complex human environment composed by non-linear elements and time-lagged effects. Currently, the huge speculation makes the task of understanding the market even harder. Furthermore many factors produce significantly impact on the financial markets, as the interest rates, the exchange rates or growth rate. And finally, there is not a solid theory about the consequences, effects or implications that these factors originate in the financial assets.

On one hand the practitioners of finances are focused on achieving the best way to predict the behaviour of stock prices in an attempt to beat the market. On the other hand, academics are more attracted by the intellectual challenge provided by the environment of the stocks markets. This scenario has contributed to that in the last two decades we have witnessed the emergence of automated systems focused on the trading of the Stock Market. The use of the computers to automate features of the investment process has became an important role in the actual stock market and in many cases has already been able to break the process of manual investments. In this chapter we introduce the analyses used by our system to invest in the market.

If we observe investment analysts at exchange markets, the majority of them tend to fall into one of two schools of thought, namely those of fundamental or technical analysis. Technical analysis is focused on the price movement of a security and uses it to predict future price movements. Fundamental analysis, on the other hand, looks at large spread of economic factors, known as fundamentals. Both analysis are commonly used by the investors, even both at the same time as they are somewhat complementary methods. For example many fundamental investors combine its primary knowledge with the technical methodology to decide entry and exit points, as many technical investors use fundamental techniques to limit the universe of possible stock of good companies. This chapter describes how the analysis works trying to predict the behaviour of the stock market environment,

2.3 Technical Analysis

The roots of modern technical analysis are based on the Dow Theory, formulated by Charles H. Dow in a series of Wall Street Journal editorials. The Dow philosophy about markets have been widely used from the beginning and even today the basic components of Dow theory are still useful.

Richard Schabacker is considered the Father of modern technical analysis, picking up where Charles Dow left off in [100], [101], [102]. Schabacker was the first to classify charts patterns (very common today), he developed the theory of price gaps, formalized the use of trend-lines and proved the importance of support and resistance levels. His nephew, Robert

Edwards, follows his work with one of the considered seminal works of the technical analysis: *Technical Analysis of Stock Trends*.

Technical analysis is based in that all information of the market is already reflected in the stock price, a security analysis technique to predict the trend of the prices through the analysis of historical market data, primarily price and volume. Technical analysis does not care about the real *value* of a stock is, instead their price predictions are only extrapolations from historical price patterns.

Technical analysis success is actually in the edge of the controversy. Researchers as Messe and Rogoff [77] find out that the fundamentals variables are useless to invest in short/long term ("Determination Puzzle of Foreign Exchange"). Otherwise, Technical Analysis has demonstrated satisfactory behaviours to forecast trends. The majority of empirical researchers claim high profits in their results, at the same time theoretical evaluations often evaluate these strategies with a low predictability power. To illustrate this idea we cite the authors of [87] that state "Among a total of 95 modern studies, 56 studies find positive results regarding technical trading strategies, 20 studies obtain negative results, and 19 studies indicate mixed results. Despite the positive evidence on the profitability of technical trading strategies, most empirical studies are subject to various problems in their testing procedures, e.g. data snooping, ex-post selection of trading rules or search technologies, and difficulties in estimation of risk and transaction costs.

The technical analysis can be split in two sections showing bellow :

- Graphical analysis : analyses revealed only the information in the charts of the securities, without using additional tools. Technicians use charts searching for archetypal price chart patterns, such as the well-known head and shoulders or double top/bottom reversal patterns.

- Technical analysis in the strict sense: studies the calculation of indicators according to the different characteristic variables of the securities behaviour. Technicians study technical indicators, moving averages, relative strength index, channels, etc.

The trading system developed in this work is focused on the second one of the outlined categories, in our opinion more accurate and appropriate to a computer program. Technical analysts widely use market indicators of many sorts, some of which are mathematical transformations of price, often including up and down volume, advance/decline data and other inputs. These indicators are used to help assess whether an asset is trending, and if it is, the probability of its direction and of continuation. Technicians also look for relationships between price/volume index and market indicators. These indicators play a crucial role in the whole technical analysis and in our trading system and are commonly called *technical indicators*.

2.3.1 Applied Technical Indicators

Technical indicators are variables whose values are derived mainly from the time series price of a security. Usually most of the technical indicator use the closing prices to calculate its values, however they also use any combination between high, low, open or close price over a specific period of historical data. Furthermore, there are technical indicators based also on the volume, the open interest, or any combination among them. The main target of a technical indicator is to procure a series of signals that helps to predict the direction of the future prices or the price trend. Trends are conditioned by the analysed period, thus there are multiple trends in a particular asset and the trend predictions change deepening on the periodicity of the prediction: long, short or medium term.

A technical indicator should not be used as lonely source of information because then it can draw false signals. That is at least with simple indicators, because in the wide world of these variables there are complex indicators formed by sets of another simple indicators. To avoid the false signals the information provided by an indicator should be combined with others indicators or other investment tools. Furthermore, some technical indicators can exhibit good performance with a specific firm or kind of market and obtain bad results for other companies or sectors.

Technical indicators have multitude of characteristics to be classified, however we can split the technical indicators in two main categories: the leading indicators and the lagging indicators. A leading indicator, as it name suggest, is an indicator to lead price movements, thus it should change its trend before the real trend of a security changes. Usually, they are useful forecasting in the short-term trends. This set of indicators comprise some of the more popular ones as the Relative Strength Index, the Commodity Chanel Index or the

Momentum. On the other hand a lagging indicator is an indicator which should change after the real trend of a specific price security, as it name implies, is an indicator that follows the price action and that is more accurate for signaling long terms trends. This type of indicators involves some popular indicators as the moving average of convergences and divergences or the exponential moving averages which usually are used during trending periods or long-term investments. There are a countless number of technical variables, next we describe the detail of the technical indicators that we will use along this thesis.

Moving Average Crossover (MAC)

Before we can show how the MAC works, we need to explain its basics, that is, a moving average. A moving average is a convolution of the function of close prices with a pulse function which represents the period of the moving average. The simplest way of calculating the moving average in a period of time N, is:

- First: summing up the weighted closing price of the current value and the N-1 value prior to it.
- Second: dividing this result by N (values in the period).

Through this thesis we use four different moving averages: the simple, the weighted, the exponential and the Hull moving averages. First, if the weight used for each value is equal, we are calculating the Simple Moving Average (SMA). One issue ascribe to the SMA is the time frame, where all the values are equally weighted. Therefore, we also implement two typical moving averages, the Weighted Moving Average (WMA) and the Exponential Moving Average (EMA). The WMA applies weights to each value decreasing in arithmetical progression and the EMA applies the weigh decreasing exponentially (never reaching zero). The moving average is a trend following indicator which, inevitably, always reacts with a certain degree of lag depending on the chosen period. The EMA can be seen as a way of reducing the lag effect inherent to the SMA. Finally, the Hull Moving Average (HMA) is a combination of different WMA proposed by Alan Hull in [52]. The author claims to build a moving average more responsive to current price activity whilst maintaining curve smoothness. Therefore, the main characteristic of this moving average is its ability to deal with the typical delay of the lagging indicators.

Let:

- n=period for the moving averages
- t= position in the time series
- $\alpha = 2/(n+1)$

The equations related to the four moving averages are:

$$SMA(n, prices)_t = \sum_{t}^{i=t-n} (price_i)/n \tag{2.1}$$

$$WMA(n, prices)_t = 2/n * (n+1) * \sum_{t}^{i=t-n} (price_i * (n+i-t)) \tag{2.2}$$

$$EMA(n, prices)_t = EMA(n, prices)_{(t-1)} + \alpha * (price_t - EMA(n, prices)_{(t-1)}) \tag{2.3}$$

$$HMA(n, prices)_t = WMA(\sqrt{n}, (2 * WMA(n, prices) - WMA(n/2, prices)) \tag{2.4}$$

As a trading strategy, multiple MA with different periods can be used. In this thesis we work with an indicator formed by 2 MAs. This indicator is commonly called Moving Average Crossover (MAC). The buy and sell signals are triggered by the crosses between averages. When the short-term MA crosses above the longer-term MA, a buy signal is triggered. When the short-term average crosses below the long-term MA, we get a sell signal. Except the complete system in Chapter 6, all the MAC are composed by two SMA.

Figure 2.1 is an example of how the MAC works in a specific period time. On the top of the chart it shows the prices of a company together with the signals triggered by the indicator. The red signal means that the indicator trigger a sell signal, instead the green signal means a buy signal. The graph in the bottom of the chart shows two different moving averages, the short in black and the long in grey.

Moving Average Convergence / Divergence

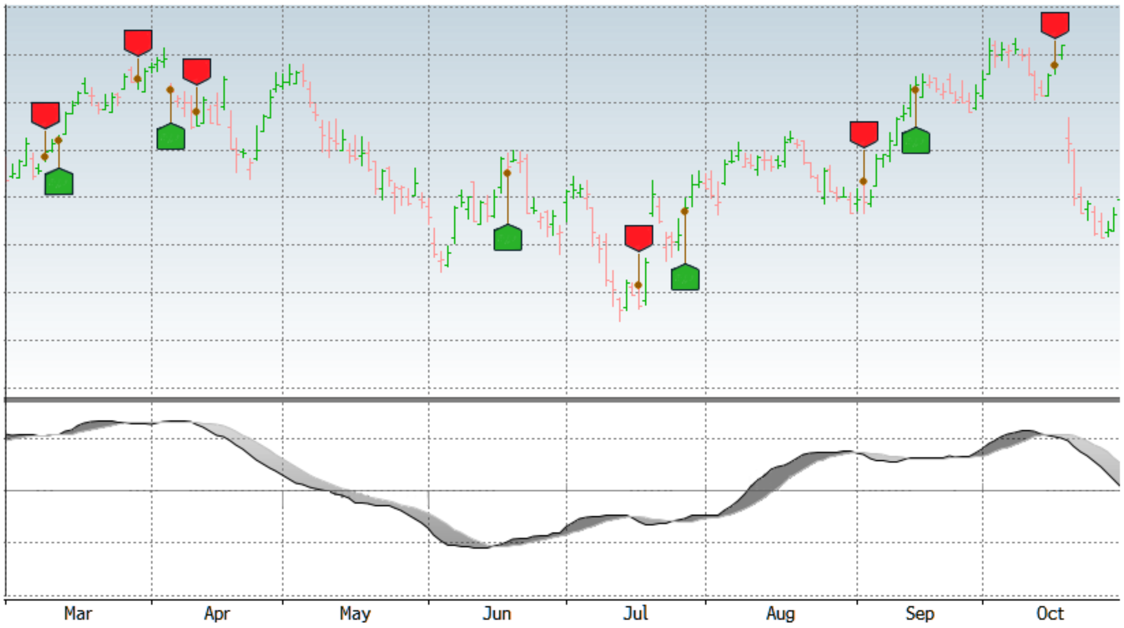


Figure 2.1: Chart of the Moving Average Crossover (MAC). Top shows prices of a company together with the signals triggered by the indicator(red: sell and green: buy) in a particular period (8 months). Bottom shows the related SMAs (black: short SMA and grey: long SMA)

An evolution of the MAC strategy is the moving average convergence/divergence (MACD) indicator created by Gerald Appel during the late 1970's. For a more detailed description of the MACD the reader can go to the Gerald Appel book [4], nevertheless we show the basic theory in two points:

- First: the MACD line which is the difference between two EMAs. In the original indicator of Appel the period of the EMAs were 12 and 26 respectively.
- Second: the signal line which is an EMA of the MACD line. Again, the original period was 9.

MACD works similarity to the MAC, when the signal line crosses above the MACD line, we have a buy signal. When the signal line crosses below the MACD line, we get a sell signal. These two lines are usually represented along with the difference between them, in the form of histogram which can help the visual traders. As such the MAC, except in

Chapter 6, we use the classic version and all the MACD are composed by two EMA.

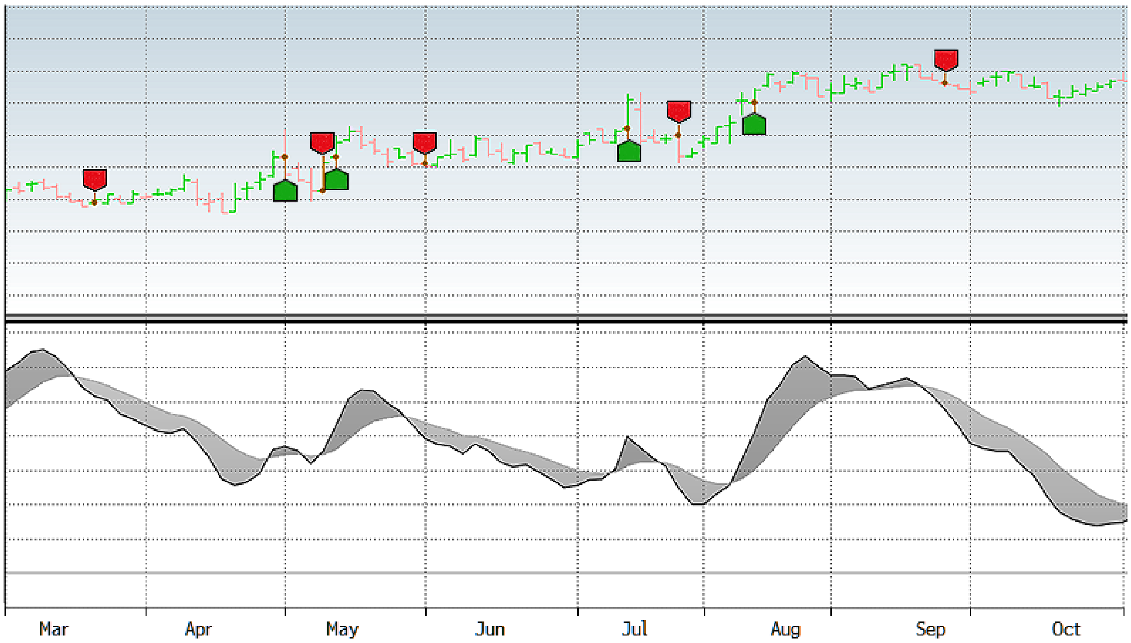


Figure 2.2: Chart of the Moving Average Convergence Divergence (MACD). Top shows prices of a company together with the signals triggered by the indicator(red: sell and green: buy) in a particular period (8 months). Bottom shows the crosses of the difference between two EMAs (black) and the signal EMA (gray)

Figure 2.2 represent an example of how the MACD works in a specific period of time. It shows the prices of a company together with the signals triggered by the indicator in the top of the chart. The red signal means that the indicator trigger a sell signal, instead the green signal means a buy signal. The graph in the bottom of the figure shows two lines, the black one is the signal, the grey line is the MACD line.

Relative Strength Index (RSI)

The Relative Strength Index (RSI) was presented by J. W. Wilder in his book “New Concepts in Technical Trading Systems” [113] It is based on the relative strength factor (RS) of a certain period (14 in the book of Wilder) which compares individual upward or downward movements of successive closing prices.

$$RSI = 100 \times \frac{RS}{RS+1} \tag{2.5}$$

And where RS is formulated as:

$$RS = \frac{Average\,of\,the\,X\,price\,increases\,of\,upward\,days}{Average\,of\,the\,X\,price\,decreases\,of\,downward\,days} \tag{2.6}$$

This way, although the RS can oscillate between zero and infinite , the RSI oscillates between 0 and 100. So, the RSI bounds the erratic movement of the RS and give us clearly areas for the price action momentum. The main strength of the RSI, in words of Wilder, is the ability to predict near reversals in the current trend, that is, a leading indicator. There are different ways to use this indicator to create the buy and sell signals, we use two approaches to the RSI:

- The first method is the Relative Strength Index by Overbought/Oversold (RSIO), it is based on the range of the RSI (0-100). This implies that the RSI can also be used to identify the overbought/oversold levels in a counter. Most technical analysts consider the RSI value above 70 as “overbought zone” and below 30 as “oversold zone”, however thanks to the computer these levels can be adjusted according to the evaluated security. For instance, volatile stocks may hit the static overbought and oversold levels more frequently than stable stocks.

- The second method is the Relative Strength Index by Divergences (RSIOD), it is based on the signals produced by the indicator when its movement diverges from the price action. A positive divergence means a buy signal and appears when the RSI builds a positive trend despite the lower trending by the price. Similarly, a negative divergence occurs when the RSI starts a negative trend while the real price follows a higher trend.

Figure 2.3 represent an example of how the RSI works in a specific period time. It shows the prices of a company together with the signals triggered by the indicator in the top of the chart. The red signal means that the indicator trigger a sell signal, instead the green signal means a buy signal. The graph in the bottom of the figure shows the two thresholds

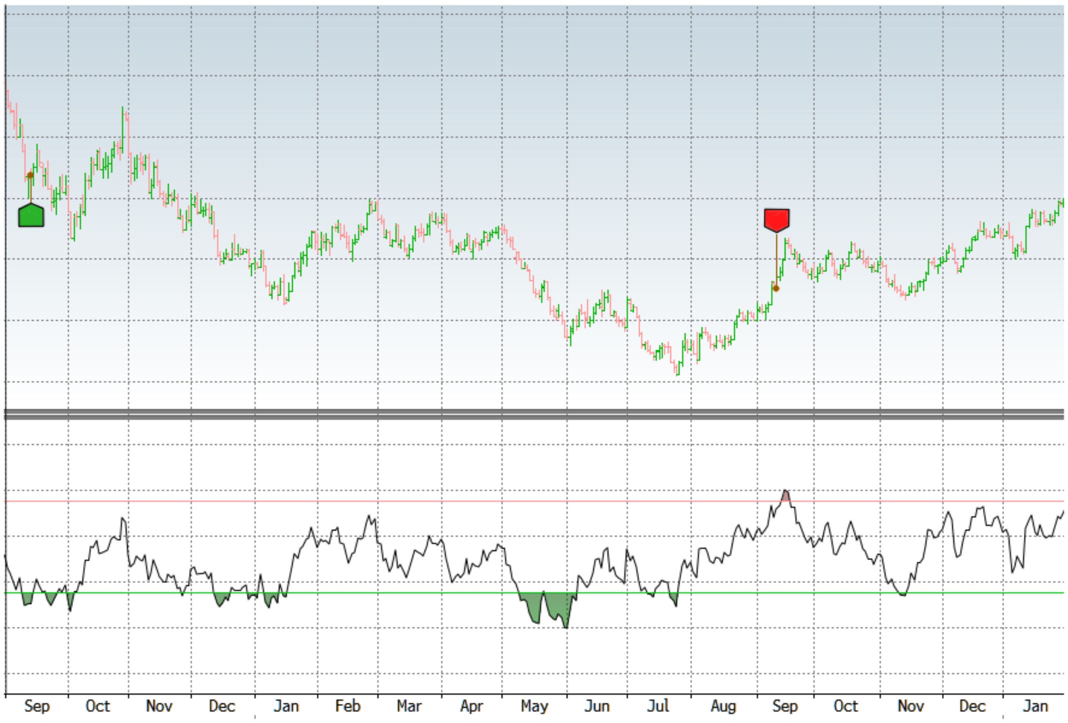


Figure 2.3: Chart of the Relative Strength Index (RSI). Top shows prices of a company together with the signals tiggered by the indicator(red: sell and green: buy) in a particular period (17 months). Bottom shows the values of RSI (black) crossing with the thresholds, lines of overbought (red) and of oversell (green)

(overbought and oversold) and the calculated RSI.

Volume Price Confirmation Indicator (VPCI)

A technical indicator that looks at the relationship between volume and price in an attempt to assess whether a trend will continue.

The Volume Price Confirmation Indicator (VPCI) was described by Dormeier in [35]. According to the author, the VPCI measures the intrinsic relationship between the prevailing price trend and volume. "Volume proceeds price" has been a mantra of technical analysis for a very long time. In theory, increases in volume generally precede significant price movements. Dormeier concludes that when examining price and volume together, they give indications of supply and demand that neither can supply independently.

The price/volume relationship confirms or contradicts the price trend. When volume increases, it confirms price direction; when volume decreases, in contradicts price direction. Thus, the operation rules can be summarised next:

- On one hand, when the trend of prices is positive and the trend of volumes is negative, VPCI contradicts the price trend and it triggers a sell signal.
- On the other hand, if the trends of prices and volumes are negatives, again VPCI contradicts the price trend and it triggers a buy signal.

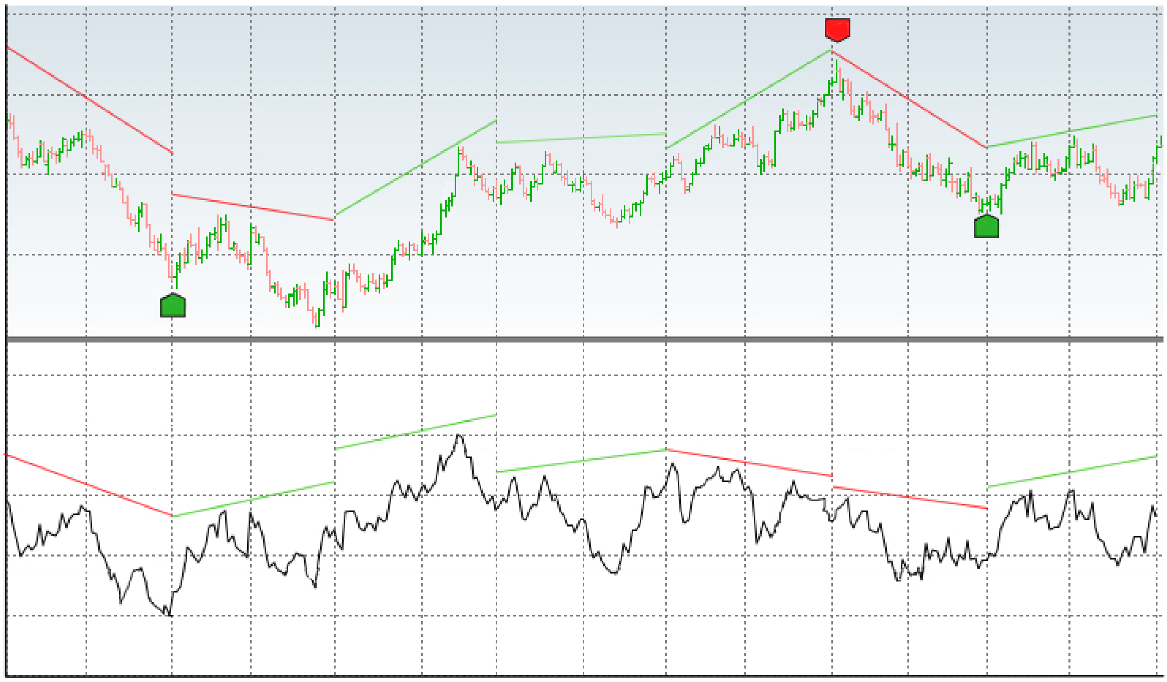


Figure 2.4: Chart of the Volume Price Confirmation Indicator (VPCI). Top shows prices of a company together with the signals and trend-lines over a period of 14 months. Signals are triggered by the indicator(red: sell and green: buy). Trend-lines are introduced each two months (red : negative trend and green : positive trend). Bottom shows the trend-lines and values of the volumes related to the same company and period.

Figure 2.4 represent an example of how the VPCI works in a specific period time and for one specific company. It shows the prices, trend-lines and the signals triggered by the indicator. The red signal means that the indicator trigger a sell signal, instead the green signal means a buy signal. Trend-lines are introduced each two months, the red lines represent negative trends and green positive trends. The graph in the bottom of the figure shows the volumes of the related company together with the trend lines presented with same colour code than the price lines.

The Supports and Resistances

The Supports and Resistances (SR) are basic concepts of the technical analysis also forming a classical indicator. This indicator presents certain price thresholds which usually are not crossed by the price, such that the price tends to rebound before reach it. Once the price pass these levels (including some noise), it is likely to continue dropping/raising until it finds another support/resistance level.

A support is a level price below the current price where the buying power exceeds the sales, so bearish momentum will be slowed and therefore the price will rebound. A resistance is the opposite concept of a support. It is a level above the current price in which the sales force will exceed the purchases ending the upward momentum, and therefore forcing the the price back down. The support and resistances are commonly identified as previous downs and highs achieved by the historical prices

Figure 2.5 represent an example of how the SR works in a specific period time for a particular company. The figure shows the price values together with the lines representing the supports and resistances. The signals are triggered when the price break one of the support lines in blue or resistance lines in red. We obtain a sell signal if the price breaks a support and a buy signal if it does with a resistance.

2.4 Fundamental Analysis

Fundamental analysis is based on the study of all available information in the market about a particular company and its sectoral and macroeconomic context. Thus, this analysis seeks to understand and evaluate the true value of the security or action. This value is used as an estimate of its financial value, which in turn is supposed to be an indicator of the future performance expected. Tables 2.2, 2.3 and 2.1 show a classification of the fundamental

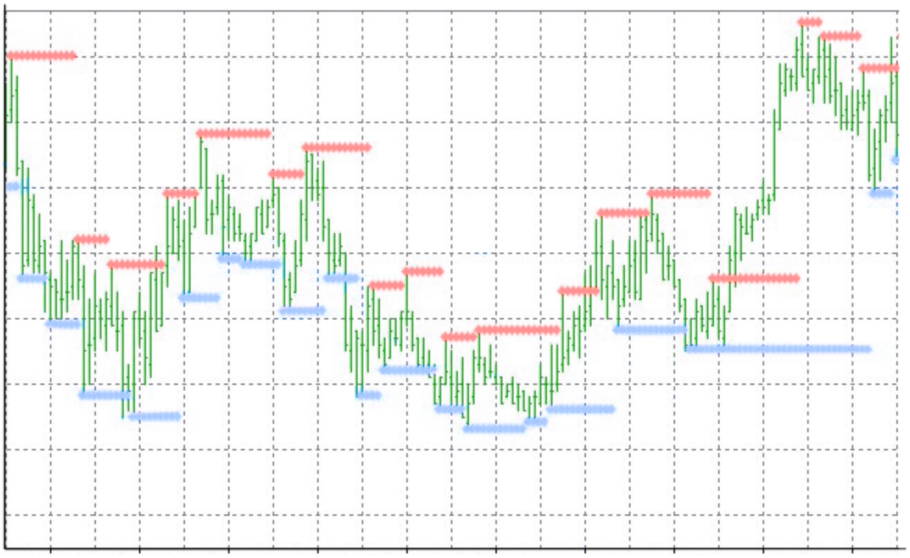


Figure 2.5: Chart of the Supports and Resistances indicator (SR). Figure shows prices of a company together with the lines of supports (blue) and resistances (red) over a period of 22 months. Signals are triggered when the price itself break a resistance (buy) or a support (sell) lines.

analysis grouping by their different uses. As the tables pointed we can split the fundamental analysis in three strands:

- A specific company analysis based on the evaluation of the financial statements of the firms 2.1
- An industrial analysis evaluating the different economic activities of the companies 2.3
- An analysis that seeks to evaluate more general aspect of the economy related to economic regions 2.2.

Although all these strands are widely used by professional investors, usually they do not apply in trading systems. There are few trading systems using fundamentals to analyse companies, and as far as our knowledge is concerned, only few of them use an analysis that takes into account sectoral and political factors in investment decisions. In order to argue the inclusion of these unusual analysis in a computer trading systems, we will perform empirical tests in the Chapter 4 to measure the utility that an industrial analysis can reach in a trading system. Next we present the applied fundamental indicators to our trading system.

Table 2.1: Set of variables used in the valuation of the companies through its fundamentals

PHASES OF THE FUNDAMENTAL ANALYSIS
COMPANY ANALYSIS
Products
Position of the company in the sector (competitive forces)
Prices and quality
Import and exports
Markets strategies
Diversification policy
Technology
Financing Policy
Analysis of the balance sheet, ratios, income statement

Table 2.2: Set of variables used in economic analysis related to countries

PHASES OF THE FUNDAMENTAL ANALYSIS
MACROECONOMIC ANALYSIS
ECONOMIC ANALYSIS
The Gross domestic product (GDP) or growth of a country
The consumer price index (CPI) or inflation
Interest rates and money supply
Exchange rate
Imports and Exports
Public and Private Investment
Savings Rate
The public deficit, public debt to GDP
Balance of payments
Fiscal Policy
Producer price index
Labour market (wage costs, labor flexibility)
Unemployment rate

2.4.1 Fundamentals to Analyse Companies

The fundamental analysis tries to evaluate the companies in the most accurate way. Therefore, in this thesis we evaluate the most important aspects of a company : valuation,

Table 2.3: Set of variables used in industrial analysis related to different sectoral activities

PHASES OF THE FUNDAMENTAL ANALYSIS
MACROECONOMIC ANALYSIS
INDUSTRIAL ANALYSIS
Importance of the sector at national level
Comparison or importance in the international scope
Exposure to foreign competition
Degree of concentration and cooperation
Degree of maturity
Barriers to entry in the sector
Exit barriers
Porter Analysis
Institutional Analysis (legislation)
Lifecycle sector
Sensitivity to economic cycles
Trends in the short and medium term
Margins with those operating in the sector
Technology

solvency, ability to repay and profitability. We use the next eleven ratios :

Valuation: One of the most used set of ratios is the one that measure the valuation of a company, these ratios estimate the attractiveness of a potential or existing investment. We use four ratios evaluating the more attractive companies to invest with the lower values.

$$Ratio_0 = \frac{MARK\_CAP}{NET\_INC} \qquad Ratio_1 = \frac{MARK\_CAP}{EBIT}$$

$$Ratio_2 = \frac{MARK\_CAP}{EBITDA} \qquad Ratio_3 = \frac{MARK\_CAP}{COM\_EQY}$$

Solvency: This set of indicators looks to measure the ability of a business to meet its related debt and other liabilities in a short or long term. We use this ratio comparing it with its competitors in the same industry, the lower the ratio, the greater the probability that the company will default on its debt obligations.

$$Ratio_4 = \frac{COM\_EQY}{LT\_DEBT} \qquad Ratio_5 = \frac{COM\_EQY}{TOT\_AS}$$

Ability To Repay: A widely used term to refers the financial capacity of an individual to make good on a debt. With this fundamental ratio we measure the relation between the borrower's total current income and existing debt. Thus we can avoid the enterprises more likely to be unable to cope with its debts. The higher the ratio the better capacity to repaid debts

$$Ratio_6 = \frac{EBITDA}{LT\_DEBT}$$

Profitability and growth: This type of financial measure is used to assess the ability of a company to generate earnings as compared to the expenses and other costs incurred during a specific period of time. We use these ratios looking for the highest values in relation to the ratios of competitors or looking for the same relation among previous periods of the company company.

$$Ratio_7 = \frac{EBIT}{TOT\_AS} \qquad Ratio_8 = \frac{EBITDA}{TOT\_AS}$$

$$Ratio_9 = \frac{NET_INC}{TOT\_AS} \qquad Ratio_{10} = \frac{SALES_{t+1} - SALES_t}{SALES_t}$$

Millions of fundamentals ratios can be used to trade companies, thus the first step to solve the investment problem is to select the indicators to be included in the decision making trading system. Next, we present the fundamental variables that we use to calculate our fundamentals ratios :

- **Common equity** (COM_EQY): total market value of the common stockholders, thus discounting the preferred stockholders. It is equal to shareholders' equity minus preferred equity.
- **EBIT**: As its own name indicate (Earnings Before Interest and Taxes) this variable is basically the net income avoiding the interest and taxes.
- **EBITDA**: The same definition than the EBIT but also excluding depreciation and amortization (Earnings Before Interest, Taxes, Depreciation and Amortization). EBITDA is useful because eliminates the effects of financing and accounting decisions.
- **Market capitalization**(MARK_CAP): The total value of the issued shares of a listed company.

- **Long term debt** (LT_DEBT): Loans of a company including any financial liabilities that are to come due in a greater than one year

- **Total Assets**(TOT_AS):The sum of current and long-term assets owned by the company

- **Net Income**(NET_INC): Total profits of a company. Net income is calculated by taking revenues and adjusting for the cost of doing business, depreciation, interest, taxes and other expenses.

- **Sales volume**(SALES): Total volume of sales of a company.

The variables that we have selected allow us to build a set of 11 ratios. Each ratio is related with a specific side of the financial of a company, thus covering 4 different aspects in the investments decisions: solvency, profitability, valuation and ability to repay.

2.4.2 Fundamentals to Analyse Macroeconomics

The securities markets are substantially influenced by general political or macroeconomic aspects. The macroeconomic analysis focuses on studying the global behaviour of the economic system that is reflected in a large number of variables such as investment or consumption of a country, sales or unemployment of sectors, etc. The purpose of macroeconomics is to obtain a simplified vision of the economy but at the same time allows to know the level of economic activity of a country or industrial sector.

Macro-economists study aggregate indicators such as GDP, unemployment rates, and price indexes to understand how the whole economy functions. Macro-economists develop models that explain the relationship between such factors as national income, output, consumption, unemployment, inflation, savings, investment, international trade and international finance. Next, we detail all the macroeconomic information used in this thesis. Table 2.4 shows the variables with original database acronyms. Data belongs to AMECO, which is the annual macroeconomic database of the Directorate General of European Commission for Economic and Financial Affairs. AMECO contains data for 28 European countries and other countries, we only use the data related with the Euro-area.

Table 2.4: Macroeconomic variables used in this work (industrial and economic). Acronyms from AMECO database.

VAR	**DESCRIPTION AND FORMULATION**
	INDUSTRIAL VARIABLES
PVG1	Price deflator gross value added; agriculture, forestry and fishery products PVG1= (UVG1 : OVG1) x 100 OVG1 : Gross value added at constant prices; agriculture, forestry and fishery products
PVG2	Price deflator gross value added; industry excluding building and construction PVG2= (UVG2 : OVG2) x 100 OVG2 : Gross value added at constant prices; industry excluding building and construction
PVG4	Price deflator gross value added; building and construction PVG4= (UVG4 : OVG4) x 100 OVG4 : Gross value added at constant prices; building and construction
VPRI	Industrial production, construction excluded
UVG1	Gross value added at current prices in agriculture, forestry and fishery products
UVG2	Gross value added at current prices in industry; ex. building and construction
UVG4	Gross value added at current prices in building and construction
UVG5	Gross value added at current prices in services
UVGM	Gross value added at current prices in manufacturing industry
PVG5	Price deflator gross value added; services PVG5= (UVG5 : OVG5) x 100 OVG5 : Gross value added at constant prices; services
PVGM	Price deflator gross value added in the manufacturing industry. PVGM= (UVGM : OVGM) x 100 OVGM : Gross value added at constant prices; manufacturing industry.
QLCM	Real unit labour costs in the manufacturing industry QLCM t = [(UWCM t : NWTM t) : (UVGM t : NETM t)] UWCM : Compensation of employees; manufacturing industry NWTM : Employees, persons; manufacturing industry NETM : Employment, persons; manufacturing industry t = year under review
UIGCO	Gross fixed capital formation at current prices in construction UIGCO : UIGDW + UIGNR
UIGDW	Gross fixed capital formation at current prices in dwellings
UIGNR	Gross fixed capital formation at current prices in non residential construction and civil engineering
UIGEQ	Gross fixed capital formation at current prices in equipment UIGEQ = UIGMA + UIGTR

Table 2.4 continues in next page

VAR	DESCRIPTION AND FORMULATION
UIGMA	Gross fixed capital formation at current prices in metal prod. and machinery
UIGTR	Gross fixed capital formation at current prices in transport equipment
ECONOMIC VARIABLES	
NUTN	Total unemployment.
NETN	Employment persons in the total economy.
NSTD	Number of self-employed.
UCPH	Private final consumption expenditure at current prices.
ZCPIH	Harmonised consumer price index.
UIGG	Gross fixed capital formation at current prices in general government.
UIGP	Gross fixed capital formation at current prices in private sector UIGP = UIGT - UIGG UIGT : Gross fixed capital formation at current prices in total economy
USNN	Net national saving USNN= USGN - UKCT USGN : Gross national saving UKCT : Consumption of fixed capital at current prices in the total economy
HVGTP	Gross national disposable income per head of population HVGTP= (UVGT : NPTD) x 1000 UVGT : Gross national disposable income NPTD : Total population
UVGD	Gross domestic product at current market prices
HVGDP	Gross domestic product at current market prices per head of population HVGDP= (UVGD : NPTD) x 1000 UVGD : Gross domestic product at current market prices NPTD : Total population
PLCD	Nominal unit labour costs; total economy PLCD t = [(UWCD t : NWTD t) : (OVGD t : NETD t)] UWCD : Compensation of employees-; total economy NWTD : Employees, persons; all domestic industries OVGD : Gross domestic product at constant market prices NETD : Employment, persons; all domestic industries t = year under review
ZVGDF	Total factor productivity
UXGS	Exports of goods and services at current prices
UMGS	Imports of goods and services at current prices
ISN	Nominal short-term interest rates
ILN	Nominal long-term interest rates
UDGG	General government consolidated gross debt
UBGS	Net exports of goods and services at current prices UBGS = UXGS - UMGS

Table 2.4 continues in next page

VAR	DESCRIPTION AND FORMULATION
	UXGS : Exports of goods and services at current prices UMGS Imports of goods and services at current prices
UBCA	Balance on current transactions with the rest of the world UBCA = UBGS + UBRA + UBTA UBRA : Net primary income from the rest of the world UBTA : Net current transfers from the rest of the world
UBLA	Net lending (+) or net borrowing (-); total economy UBLA = USNN + UBKA + UITT + UKCT USNN : Net national saving UBKA : Net capital transactions with the rest of the world UITT : Gross capital formation at current prices, total economy UKCT : Consumption of fixed capital at current prices; total economy

2.5 Automated Trading Systems

In the last two decades we have witnessed the explosive emergence of automated systems focused on the trading of the Stock Market. An Automated Trading System (ATS) is a computerised system that automatically submits trading orders to an exchange. This issue has trigger that even floor has been replaced by the adoption of electronic trading [2], that we could consider one of the major changes that has occurred in the structure of the market in last years [103], [76].

The early projects developing ATS were difficult. The scepticism towards the technology in the finance's world is evident even today. Novel technologies are often not welcomed and some traders use the this systems in a disguised way, preferring to use such terms as *statistical modelling techniques* because they do not want to explicitly the technology name name to their client presentations. Nevertheless the use of the computers to automate features of the investments process has a important role in the actual stock market. For instance, the 33% of all EU and USA stock trades in the year 2006 were generated by automated programs or algorithms, according to Boston-based financial services industry research and consulting firm Aite Group (http://www.aitegroup.com/). In the last years algorithmic execution

[2] Although some exchanges maintain the floor trading systems (New York Stock Exchange), there are a lot of exchanges have replaced floor trading systems by electronic trading systems, including the Swiss Stock Exchange, the London Stock Exchange, the Vienna Stock Exchange and the Amsterdam Stock Exchange.

have a significant percent of all financial movements as a percentage of total U.S. equities trading volume has increased from approximately 28% in 2004 to just over 50% in 2010 [76].

An automated treading system defines the investment problem as to maximize the return (or risk adjusted return) for a specific time period when investing long or short-sell in a financial asset (in our case a stock). Since the performance of investment decisions in stock markets is influenced by a wideness of factors of different type: political, macroeconomic, regulatory, local, international, etc. that are uncertain, there is not a single and perfect rule with specific parameters or threshold values that can be used to maximize future returns, but a broad set of potential rules which combine indicators and ratios representing different factors and driven by a range of values in their parameters. Therefore, the investment problem consists first of finding the best combination of variables and second of fine-tuning the parameters for these variables to obtain the maximum return when applied to the investment decision. Thus, the input set of rules for a trading system consists in indicators and ratios to be used as investment criteria and the related parameters for these variables, while the output is the return obtained by the trading system this way defined.

An ATSs could be a simple system using only a specific technical indicator as such the moving average, or a complex systems based on methodologies as such Fibonacci retracement [41], linear regression [65], neuronal networks [44], fuzzy logic [36], genetic programming [34], etc. In this thesis we develop automatic trading systems focused on genetic algorithms and grammatical evolution which will be described in Chapter 3 together the related work of ATSs based on this type of algorithms.

Chapter Conclusions This chapter has presented some of the most important background theories supporting the behaviour of the Stock Markets along the history. We have focused in theories which play a critical role in this thesis as the Random walk that we discuss in Chapter 4, or the AMH that supports the ATSs developed in this dissertation. Finally, the Chapter has detailed the three main analysis used to optimise our ATS (macroeconomic, fundamental and technical analysis) defining the related financial variables used along the work.

Chapter 3

Evolutionary Algorithms

By the 1980s symbolic artificial intelligence wanted to simulate physiological systems, all the processes of human cognition, especially perception, robotics, learning and pattern recognition, thus, triggering a higher number of approaches trying to found solutions in specific AI problems. The sub-symbolic approaches, such as neural networks, fuzzy systems and evolutionary computation are now studied collectively by the emerging discipline of computational intelligence (IEEE Computational Intelligence Society).

Nowadays the importance of this type of artificial systems in trading and other topics can not be questioned [111] [114]. Evolutionary Algorithms (EAs) have become in one of the most important branches in the computational intelligence. EAs are defined as search and optimization procedures that have its origins and inspiration in the biological world . EAs are characterised by emulating the evolutionary behavior of nature and are based on the survival of the best individuals or individual. Individuals are potential solutions of the problem and they are implemented as data structures. EAs work with sets of these individuals forming the called populations, which evolve from generation to generation thanks to modifications applied by the genetic operators. These operators manage the parameters of each individual allowing multiple implementations adapted to the problem. There are countless number of evolutionary approaches, as the evolutionary strategies, evolutionary programming, genetic programming, memetic algorithms, differential evolution, etc. In this thesis we propose the use of the EAs to optimise an an Automatic Trading System (ATS). Among them, we will focus on the well-know genetic algorithms and the flexible grammatical evolution.

One of the most important advantages of EAs is that they are inherently parallel. Most other algorithms work in series and they can only explore the space of solutions in one direction at a time. However, since GAs have multiple offspring, they can explore the space of solutions in multiple directions at once. Therefore, EAs are suitable for problems with a complex landscape (discontinuous, noisy, changes over time, multiple local optima, etc.). The problem that we face in this thesis has a vast space of solutions, and it is impossible to explore exhaustively. Many search algorithms may be trapped in local optima, however these algorithms have proven to be effective in escaping from local optima and finding the global optimum even in very rugged and complex fitness landscapes.

Another advantage of EAs is the use of fitness evaluations. When a EA assess the suitability of a particular individual allows to the algorithm polls at the same time each one of the spaces to which that chain belongs to. After many evaluations, the algorithm provides an increasingly accurate value of the average fitness of each one of these solution spaces, where each one contains many members. Therefore, when a EA evaluate a small number of individuals in a explicitly way, at the same time the EA is evaluating a much larger group of individuals in a implicitly way. Likewise, the EA can addressed into the sub-space of solutions with the fittest individuals, to find the best individual of that group. In the context of Evolutionary Algorithms, this is known as the schema theorem, and is the main advantage of EAs over other methods of problem solving.

Due to this parallelism, that allows them implicitly evaluate many schemes at a time, EAs are particularly good at solving problems whose space of potential solutions is too large to perform an exhaustive optimisation in a reasonable time. Most problems fall into this category are known as "non-linear". In a linear problem, the fitness of each component is independent, so any improvement in any part will result in an improvement in the overall system. The problem faced in this thesis belongs to the non linear problem, where changing one component can have ripple effects throughout the system. The nonlinearity produces a combinatorial explosion in the number of possibilities. The implicit parallelism of EAs allows them to overcome even this huge number of possibilities, and successfully find optimal or very good results in a short period of time directly after sample only small regions of the vast fitness landscape

3.1 Genetic Algorithms

This section briefly study GAs as optimisation algorithms to tune a set of parameters due to the following reasons:

- The approximations showed in this thesis are based on EAs.
- Chapter 5 test a trading system based on GAs.
- Understanding GAs helps to understand GEs.

A genetic algorithm (GA) is a heuristic search method that mimics the process of natural evolution. As defined by Koza (see [60]), “The genetic algorithm simulates Darwinian evolutionary processes and naturally occurring genetic operations on chromosomes”..."The genetic algorithm is a highly parallel mathematical algorithm that transforms a set (population) of individual mathematical objects, each with an associated fitness value, into a new population using operators patterned after the Darwinian principle of reproduction and survival of the fittest and after naturally occurring genetic operations...”.

The first approximations to GAs appeared in late 1950 by evolutionary biologists who were explicitly seeking to model aspects of natural evolution. Since then, renowned researchers such as G.E.P. Box, G.J. Friedman, W.W. Bledsoe and H.J. Bremermann developed evolution-inspired algorithms without much success. In the sixties John Holland invented the concept of GAs and in the subsequent years they were further developed by his research group at the University of Michigan [50]. While Holland developed GAs, the same concepts were used in Germany where Schwefel and Rechemberg developed the Evolutionary Strategies (ES) [104]. However, there was no population or crossover in this technique ([49] p.146) until later versions. Later, L.J. Fogel among others introduced in [42] the Evolutionary Programming (EP), like ESs work by randomly mutating individuals and keeping the better of the two [78] and [46].

The Holland methodology evolves the process of transforming a population (set) of individuals (solutions) represented by chromosomes (strings of bits representing solutions to a problem) to a new population. Each chromosome is composed of genes (items), where each gene has a specified set of alleles (alternatives, i.e 0 or 1 in binary representation). The encoding in chromosomes is an advantage that allows GAs to manipulate multiple parameters simultaneously. The use of parallelism allows them to produce multiple solutions, with equally fitness, to the same problem (i.e allowing the possibility that an external supervisor can select one of these candidates). The GAs commonly use single chromosomes comprising a bit string of a fixed length. The fitness function assigns to each individual a value which measures the quality of the solution. This measurable quality allows to simulate the concept of better adapted individuals. Therefore, individuals with better fitness values provide higher probabilities to be selected to pass their properties to the next generation or even to be directly integrated into the next generation of the population. Thus, genetic algorithms are based on a basic principle of evolution: the best individuals are more likely to survive and reproduce than other individuals less adapted to the environment. The algorithm mimic the natural selection and evolution by the so-called operators of selection, crossover and mutation.

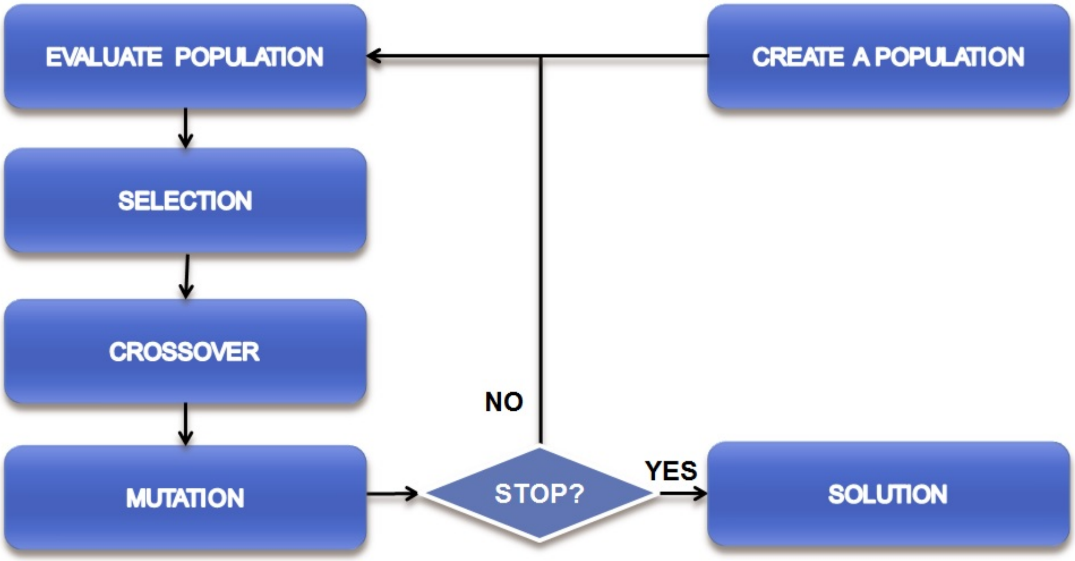


Figure 3.1: Typical genetic algorithm work flow

Figure 3.1 represents the mainly work flow in GAs. The first step is to create an initial population to work with. Each individual of this population is usually generated randomly. Once the initial population is generated, the methodology performs a repetitive process until the algorithm reaches the stop condition, as such a maximum number of generations or the population's convergence. The process starts by assessing the population and selecting the individuals who will be involved into the next generation (based on their fitness value).

Then the crossover operator is applied to obtain an offspring for the next generation of individuals. Finally, the mutation operator is applied to modify some individuals. At this point the algorithm has evolved to a new population, therefore the same steps will be repeated to move forward in the search process. We want to emphasize that this figure describe a classical GA. The variants of GAs are countless, there are implementations using a local search to initialise the population [47], problem's encodes using various chromosomes to represent a solution or variable length chromosomes [58], etc.

The functioning of the GA methodology is driven by different parameters: the crossover and mutation probabilities, the scaling of fitness function in the reproduction process, the initial population size and the numbers of generations. We consider important to mention that these parameters depend on a particular problem, this means that a good results for a combination of specific values cannot be extrapolated to others genetic algorithm problems.

To implement a genetic algorithm is necessary to define:

- The type of encoding that allows to represent solutions.
- The selection operator.
- The crossover operator.
- The mutation operator.
- A fitness function to evaluate the individuals.
- The size of the population and the number of generations to evolve.
- The probabilities of the applied operators.

The current state of the art contains many works that implement GAs as the engine of ATSs. For example Allen and Karjalaein [3] proposed one of the first works on using GAs to find technical trading Rules (this is precisely the title of the work). They applied a GA to obtain technical trading rules and compare the results with other models and the Buy and Hold (B&H) Strategy. In [83] Nuñez designed four GAs models incorporating different factors (e.g. risk, transaction costs, etc.) to obtain financial investment strategies. It is showed that all four GA models generate superior daily returns of long positions with lower risk than B&H strategy for 1987-1996 share price data from the Madrid Stock Exchange

(IBEX 35). In [54] the authors describe a trading system designed with GAs that uses different kind of rules with market and companies information. The system is applied to trade, on a daily base, to companies belonging to the S&P 500 index. They propose to apply the methodology using technical analysis and fundamental analysis separately and the authors claim that the main problem is the computational time required for training the trading system with daily data of stocks prices. This restriction is partially solved in 5.2 and 5.3 with the implementation on parallel computer architectures to speed up the functioning of a GA-based trading system to invest in stocks.

3.2 Grammatical Evolution

Grammatical Evolution (GE) is a relatively new evolutionary alternative. This evolutionary computation technique was promoted by C.Ryan, JJ. Collins and M. O'Neill in 1998 [24]. We can summarise the definition of GE as a type of Evolutionary Algorithm designed to evolve computer programs defined by a grammar, usually in Backus Normal Form (BNF notation). The most similar procedure is the Genetic Programing (GP) [60], which is also able to evolve computers programs. Although GP originally use Lisp as evolving language, the approaches using others languages are countless. The main dissimilarity that makes the GE an attractive and elegant solution is that GE does not perform the evolutionary process on a specific language. GE evolves individuals as GA does, and performs a mapping process to generate programs in any language. The GE approach becomes an attractive method thanks to its flexibility. This feature is the main advantage of the GE solutions, and is closely correlated by the great modularity that a well-structured grammar provides.

3.2.1 Backus Normal Form

BNF is a notation technique for expressing context-free grammars. The BNF can be any specification of a complete language or a subset of a problem-oriented language. A BNF specification is a set of derivation rules, expressed in the form:

$$< symbol >::=< expression >$$

The rules are composed of sequences of terminals and non-terminals. Symbols that appear at the left are non-terminals while terminals never appear on a left side. In this case we can affirm that $< symbol >$ is a non-terminal, and although this is not a complete BNF specification, we can affirm also that $< expression >$ will be also a non-terminal since those are always enclosed between the pair <>. So, in this case the non-terminal $< symbol >$ will be replaced (indicated by ::=) by an expression. The rest of the grammar must indicate the different possibilities. A grammar could be represented by the 4-Tuple N, T, P, S, being N the non-terminal set, T is the terminal set, P the Production rules for the assignment of elements on N and T, and S a start symbol which should appear in N. The options within a production rule are separated by a "|" symbol.

```
N= {<code>, <indicator>, <short>, <long>, <digito>,<onefour><fivenine> }

T= { ");" , "," , "1" , "2" , "3" , "4" , "5" , "6" , "7" , "8" , "9"
 , "MACD(" , "RSI(" , "MAC(" , "SR(" }

S= { <code> }

P= {I, II ,III ,IV ,V ,VI ,VII}

I  <code>      ::=  <indicator> <code>
                    | <indicator>

II  <indicator> ::= "MACD("<short>","<long>","<long>");"
        | "MAC("<short>","<long>");"
        | "SR("<long>","<long>");"
        | "RSI(" <short> ","<short>");"

III  <short> ::=  <digito>
        |<onefour><digito>

IV  <long>  ::=  <fivenine><digito>

V <digito>::= <onefour>
        |<fivenine>

VI  <onefour>::=  "1" | "2" | "3" | "4"
VII  <fivenine>::= "5" | "6" | "7" | "8" | "9"
```

Figure 3.2: Basic grammar to build a simple trading strategy

As we have mentioned above we will use an EA to evolve genotypes, i.e. a string of values. We use the individual genotype to map the start symbol onto terminals by reading codons of 8 bits. Each codon of 8 bit is represented by an integer value on the genotype. The mapping process is the transformation from the genotype to the phenotype. Thus, instead of representing the programs as a tree-solution, GE presents a chromosome composed by codons (genes in GA) that are directly connected with a specific rule of the grammar. The chromosome itself is considered the genotype and the real code derived from the codons is called phenotype. With the aim to understand its main advantage, we show a brief example. We specify a complete and simple grammar with the previously presented BNF notation in Figure 3.2.

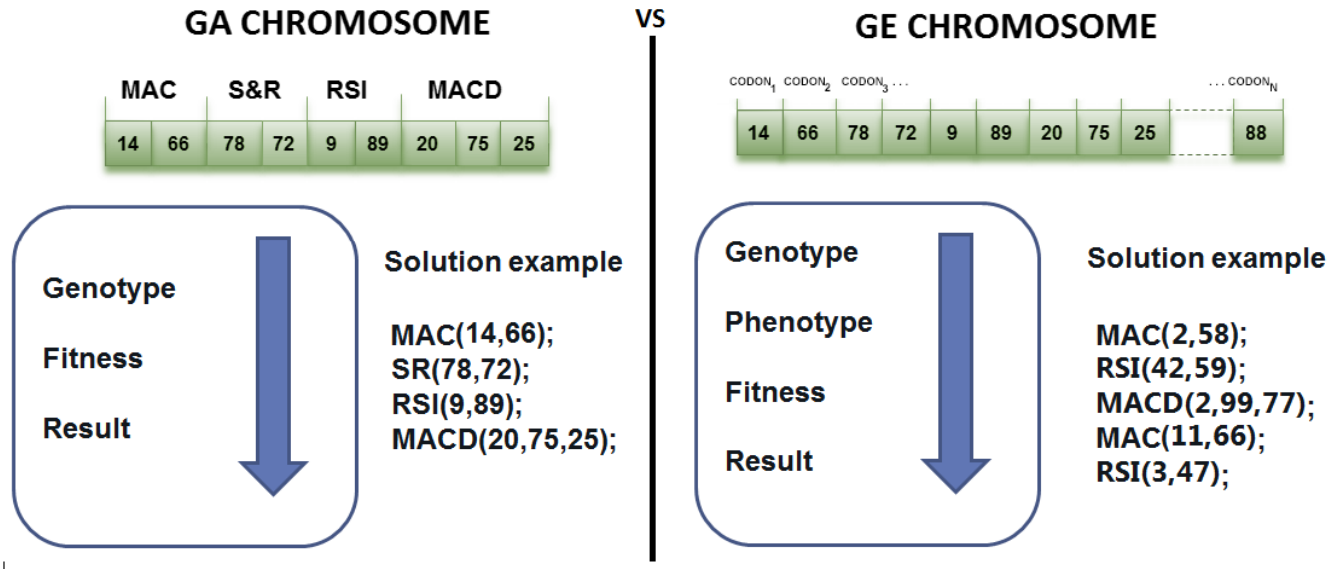


Figure 3.3: EAs chromosomes : Main differences between GA and GE

The target of this grammar is to obtain a trading system based on four technical indicators and to optimise its parameters assuming that we already implemented the functions MAC, MACD, SR and RSI. Therefore, the code solution consist in a a combination of elements of the set of terminals T which are chosen by mapping the individual to the grammar. With the purpose to understand this new features of a GE chromosome, we present the main structure of a GA and GE individuals in Figure 3.3. The figure faces two similar chromosomes, one per methodology, and presents the workflows of the decodification processes of each EA. Our mapping operator is the well-know modulo operator[1](MOD) :

$$[Solutionrule] = [Codonvalue]\ MOD\ [\#productionofasymbol]$$

[1] The modulo operator calculates the remainder of division of one number by another

Thus, each codon is linked to an specific rule of the grammar. Next, we perform the first steps of the mapping procedure related with the GE chromosome showed in Figure 3.3 and the grammar presented in Figure 3.2. We begin with the start symbol of the grammar : S, and the first codon, $codon_1$:

[Solution rule] = [14] MOD [4] = [0]

- [14] : First codon ($condon_1$).
- MOD : Method of mapping.
- [2] : Number of productions of the first non terminal which is the start symbol:

$$< code >$$

- [0] : Number of production to choose. Our partial solution corresponding to :

$$< indicator >< code >$$

After the first step of mapping we obtain the first partial solution corresponding to [<indicator><code>]. While our partial solution contains a non-terminal, the algorithm keeps reading codons and follows the same mapping process. Continuing the example, the algorithm reads the second codon [16] related to Figure 3.3 and the first non terminal of our partial solution: [<indicator>].

[Solution rule] = [66] MOD [2] = [0]

- [66] : Second codon ($codon_2$).
- MOD : Method of mapping.
- [4] : Number of productions of the first non-terminal in our partial solution which is the symbol:

$$< indicator >$$

- [2] : Number of production to choose corresponding to :

$$"MAC(" < short > "," < long > ");"$$

Thus, we update our partial solution to ["MAC("<short>","<long>");"<code>]. We must follow the process until, finally, we do not have more non-terminals and we can not continue.

During the process of mapping the genotype we could reach the end of the chromosome and we run out of codons before all non–terminals are turn into terminals. At this point, there are two main options. First, we can consider the individual as invalid, and assign it a very low fitness value to stop reproducing. Or, second, we can wrap the individual, that is, reuse the chromosome using the codons again. This technique is performed by the called wrapping operator. The wrapping operator is inspired int gene-overlapping phenomenon that has been observed in many organism [64]. The wrapping allows reuse chromosome structures to obtain broader rules, thereby influencing in the quality and diversity of generated individuals becoming an advantageous operator as Huggoson shows in the results of [51] (although some authors disagree).

GE is presented as a smart solution that fits perfectly in the context of the complexity of Stock Markets. Thus, other previous works already use the GE as a methodology to optimise investments. The BDS Group at the University of Limerick has performed an excellent work about this topic, for example in [32] the authors propose a GE to evolve a financial trading system. In this approach the authors show an adaptive grammar with a variant of the moving window. The different rules of the grammar are in constant evolution during the execution of the trading system while new data is uploading. Other works of this group show the proficiency of GE in foreign-exchange markets or in different indexes of the stock market as [12] [33] among others. Other authors proposed trading system based in GE, as in [1], where the authors build a system that uses the co-evolution of the entries, exits and stop loss for long positions, and short positions. These authors update their work in [2] where they change the fitness function with a complex fitness proposed by P.Saks [99]. Those works inspired our GE implementation that we present in Chapter 6.

Although GE fits properly in the context of the complexity of the Stock Markets, we note that GE are not limited to the financial environment and they have been used in many topics. For example Moore uses GE to generate optimal biochemical network models [80]. Also we can find GEs to solve trigonometric identities [98], to optimise dynamic memory

[23], even to compose automatically music [30].

3.3 Fitness Functions

In the field of the EAs there is a function considerate the most essential piece to guide the evolution towards more optimal solutions. The fitness function is responsible of assigning a value representing the merit of a particular individual. As the algorithm determines the workflow to follow through this fitness value, the design of the fitness function is a very important process in EAs.

The development of a fitness function depends exclusively on the problem to face. Thus, EAs implementing different type of problems are guided by different fitness funtions.The partnership of finances and EAs can use a wide range of fitness functions, indeed any trading strategy performance criterion can serve as a method of assessing individuals. For example, the K Ratio [57], the maximum drawdown [86] or the Pessimistic return on margin (PROM) [86]. In this work we use several functions to assess the individuals by the so-called fitness functions. Each one of our investments is represented by a operation signal (buy, sell or keep) that is evaluated by a fitness function. The functions are used individually in different sections of the thesis, or jointly in the same section to compare performance results between them, even the fitness functions are combined using multi-objective optimisation in the Chapter 6. Through this thesis, we have used three of these fitness functions to guide our trading system:

- The Accumulated return (AR)
- The Sharpe Index
- The Correlation coefficient between the equity curve of a strategy and the perfect profit of the market .

The first option we use to evaluate an individual is calculating the Accumulated Return (AR) obtained when applying the trading system to the sample data computed as described by equations 8.6 and 8.7. AR_f is the accumulated return at the end of the trading period and DR_i is the daily return. It is noteworthy that when the ATS does not provide signal, we could use a risk-free returns such as the risk-free return given daily by the U.S. Treasury Bills. Investors usually use this methodology, so that, the results we present throughout the thesis are slightly undervalued since we use 0 in these cases instead risk-free return. Regardless to the fitness function used in the trading system , final returns (profits) will be provide by the AR.

$$AR_f = \prod_{f}^{i=1}(1 + DR_i) \tag{3.1}$$

Where DR_i denote the return of the day_i as follow:

$$DR_i = \begin{cases} \frac{P_i - P_{i-1}}{P_{i-1}} & \text{if the ATS gives a long signal} \\ \\ \frac{-(P_i - P_{i-1})}{P_{i-1}} & \text{if the ATS signal is short selling} \\ \\ 0 & \text{if the ATS signal is neutral} \end{cases} \tag{3.2}$$

And where P_i is the close price of the day_i.

The next objective function is the well known Sharpe Index (SI) or Sharpe ratio 3.3 proposed by the Nobel William Sharpe [105], which intend to help investors figure out the returns they will get in exchange for the level of risk they will take on. Specifically, this index measures the excess return per unit of deviation in an investment asset or a trading strategy. The top half of the index is related to the funds returned over a set period RA, and subtracts the R_f as the return that an investor could have earned in a risk-free investment, typically defined as the return of the Treasury bills over the same period. The denominator is the standard deviation of the returns (σ), which measures how much is deviate the profits from its average performance,that means, the volatility of the strategy.

$$SI = \frac{RA - FR}{\sigma} \tag{3.3}$$

Where:

- AR is the accumulated return
- FR is the free risk return.
- σ represent the standard deviation

In order to asses successfully the population of EAs we need to include a modification in the classic Sharpe index 3.3. When the calculated value of the SI is negative, the fitness function presents an issue: negative values are better Sharpe ratios as more far they are from zero. Therefore, we include a modification in the calculation of SIs, thus, solving the problem in equation 3.3.

$$IS = \frac{AR - FR}{\sigma^{(R_i - R_f)/|(R_i - R_f)|}} \tag{3.4}$$

Finally, we use the correlation Coefficient between the Equity Curve of a strategy and the Perfect Profit of the market(CECPP). This objective function is presented in the book "The Evaluation and Optimization of Trading Strategies" from Pardo [86].

On one hand, the perfect equity is a theoretical measure of market potential which is related with the perfect investment in a period, therefore it obtains the maximum return possible. On the other hand, the equity curve is just the representation of the investment strategy to be evaluated. Thus, the correlation coefficient between the perfect profit and the equity curve is the relationship between both. The CECPP will range between -1 and +1. Best strategies will have values near +1, while poor strategies will produce values near 0. If a strategy gives negative values it must be disregarded.

We formulate the CECPP as follow:

$$CECPP_i = \sum_{i=1}^{n} \frac{(X_i - \overline{X})(Y_i - \overline{Y})}{((n-1)(\sigma_X)(\sigma_Y))} \tag{3.5}$$

Where:

- X is the perfect equity curve
- Y is the real equity curve
- σ represent the standard deviation

It is worth mentioning, as we can see in equation 8.7, that a neutral signal do not provide any return. That issue represents a disadvantage with respect to the practitioners or other trading systems. This is because usually the return of a neutral signal is a risk-free return given by the country target of the analysis (e.g the US Treasury Bills). Furthermore, we compute into the fitness functions the **transaction costs**. The trading costs are the expenses incurred when buying or selling securities. Transaction costs are important due to its relation to the net returns. Usually the more operations the more expenses. Transaction costs depend on the brokers, the capital invested and the stock exchanges where we invest. Thus, on one hand implementing the real trading costs in our trading systems is inefficient and in the other hand eliminating transaction costs would provide an unfair advantage to our experiments. We solve the problem by setting a fixed cost for each transaction, we simulate these additional costs calculating on each buying/selling operation a commission fee of 0.1%.

Chapter Conclusions This chapter has given a basic overview to the main methodologies used as optimisation engines of the automatic trading systems presented in this work: genetic algorithms and grammatical evolution. In addition, this chapter has introduced the fitness functions which will guide the automatic trading system in the investment process.

Chapter 4

The Predictability of Stock Prices

In this chapter we propose to use a novel approach to test the existence of undercover common and uncommon patterns in stock prices by using only information about historical prices. Extracting a common pattern typically involves recognizing a group of entities of specific types that have particular relationships between them. We use a procedure derived from an approximation to the non-computable Kolmogorov complexity which is the measure of the computational resources needed to specify an object. Our procedure, based on clusters, is capable of building hierarchical distance trees in a blind manner for a set of historical prices. This procedure performs this task by using a novel algorithm described by Contreras et al in [25] to measure the Kolmogorov complexity. Thus, this chapter stablish two objetives:

- In first sections (4.1, 4.2 and 4.4), the work frames within the weak form of the Efficient Market Hypothesis (EMH) and the random walk behaviour of stock prices. The first aim seeks to demonstrate that stock markets do not follow a random walk, and therefore they are somewhat predictable.

- In last sections (4.5 and 8.3.1) we face the industrial classifications of stock markets. Thus, the second aim wants to show the existence of trends in industrial sectors by studding companies developing similar industrial occupations.

4.1 The Random Walk Hypothesis

"...the theory of random walks says that the future path of the price level of a security is no more predictable than the path of a series of cumulated random numbers. In statistical terms the theory says that successive price changes are independent, identically distributed random variables. Most simply, this implies that the series of price changes has no memory, that is, the past cannot be used to predict the future in any meaningful way"; Fama [39].

The theory of random walks in stock prices actually involves two separate hypotheses:

- Successive price changes are independent.
- The price changes conform to some probability distribution.

But as Fama [39] recognizes "we can probably never hope to find a time series that is characterized by perfect independence. Thus, strictly speaking, the random walk theory cannot be a completely accurate description of reality". The first report of the behavior of stock market returns which suggests a random walk model for stock price changes, dates back to the work by Louis Bachelier [5] which presented (in his PhD thesis "The Theory of Speculation") a stochastic analysis of the stock and option markets. Subsequent empirical works by different authors during the first half of the XX Century probed into the adequacy of the randomness assumption: [27]; [56]; [94]; and [73]. For instance, [94] showed that a time series generated from a sequence of random numbers was indistinguishable from a record of US stock prices-the raw material used by market technicians to predict future price levels.

The mid-1960s was a turning point in research on the random character of stock prices with the publication of new more rigorous studies. In 1964, Cootner published his collection of papers on that topic in a book [26]. The main issue of the volume was whether stock prices followed a pure random walk or, alternatively had recognizable non-random patterns, perhaps profitably exploitable, with arguments on both sides.[96] concluded that the net impression left by Cootner collection of essays was that the random walk hypothesis probably does not accurately describes the movement of stock prices, but that properly modified, it is a pretty fair approximation. One year later, Fama's [39] doctoral dissertation, published in the Journal of Business, focused on the controversy of the possibility of predicting stock price changes from the history of stock prices. Examining the existing literature and carrying out several tests on stock prices behaviour, he concluded that the data seem to present consistent and strong support for the random-walk model, although he recognized that there were some slight departures from it. In 1973 the first edition of the book of Malkiel "A Random Walk Down Wall Street" [75] appeared.

During the seventies and eighties several papers reporting market anomalies and rejecting the random walk model were published ([20], [19]; [109]), and it was proved that the distributions of the daily log returns of common stocks displayed heavy tails and was asymmetric. During this time the nascent of the Behavioural Finance Paradigm offers an alternative approach to analyse market prices. More recently one of the most known books that argue against the Random Walk hypothesis "A Non-Random Walk Down Wall Street" [68] has been published. The authors pointed to the fact that there have been many long rises and long declines in the market which they believe is a clear indication that the market is not random since these rises and declines are too often and large to be explained by the arrive of new information. [106] underlines also that volatility in stocks is too large to support the EMH. Therefore, these characteristics are in contradiction with the assumption that the underlying model is a geometric Brownian motion.

4.2 Stock Market and Randomness

The first step in our set of experiments consists in testing whether our methodology can discern between prices series generated by a random walk process and real stock prices time series taken from the market. Stock prices are often modeled as the sum of the deterministic drift rate and a random number proportional to the last known price. We use the Geometric Brownian Motion (GBM) to generate series of random prices. GBM is the common model to define stock price paths. In mathematics, it is one of the best known L'évy processes (cádlág stochastic processes with stationary independent increments) and occurs frequently in pure and applied mathematics, economics and physics. The GBM is technically a Markov process that uses the sum of the deterministic growth rate and a random number with a mean of zero and a variance that is proportional to time intervals (dt). This means that the generated stock prices follows a random walk and is consistent with the called weak-form of the Efficient Market Hypothesis (EMH), that implies past price information has already been built-in and the next price movement is "conditionally independent" of past price movements. Brownian motion is described by the Wiener process; a continuous-time stochastic process named in honour of Norbert Wiener. It is defined by the following stochastic differential equation.

$$\begin{cases} dS_t = S_t\mu dt + S_t\sigma dW_t \\ dW_t = \epsilon\sqrt{dt} \end{cases} \tag{4.1}$$

Where St is the stock price at time t, dt is the time step, μ is the drift, σ is the volatility, Wt is a Weiner process, and ϵ is a normal distribution with a mean of zero and standard deviation of one. Hence dSt is the sum of a general trend, and a term that represents uncertainty. Converting Equation 8.4 into a finite difference form gives Equation 8.5:

$$\Delta S_{t+\Delta t} = S_t\mu\Delta t + S_t\sigma\epsilon\sqrt{\Delta t} \tag{4.2}$$

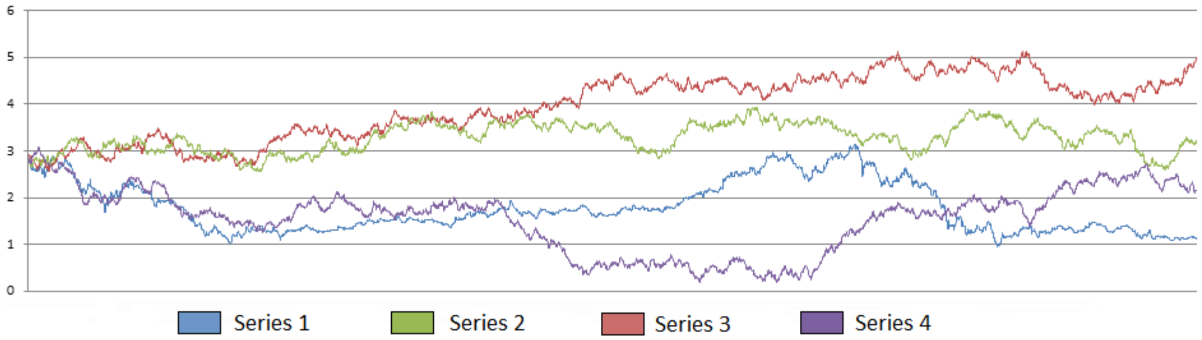


Figure 4.1: Random series versus price series: Series 1 in blue shows daily prices of A2A S.P.A (Italy) over the period 2002 - 2011. The remaining series are modelled price time series by the Geometric Brownian Motion

Figure 8.1 shows four series of historical stock daily prices covering ten years. On one hand Y axis represents the price of the company's share. On the other hand the X axis represents the period of time between 2002 and 2012, thus covering ten years. Only one of the series belongs to a real company; the other three series have been artificially generated by the GBM. The Figure shows the complexity involved in the task of distinguish historical price series of artificially modelled price series. The behaviour of the three modelled companies is indistinguishable, at first glance, of the behaviour of the real company.

Some authors have found that information about fundamentals has limited ability to predict stock prices on short horizons, but strong and significant ability to forecast prices at longer horizons ([92]; [15]). Thus why following the same previous skeleton we represent in Figure 8.2 four series of historical prices, but in this case with monthly prices. Again, only one of the series belongs to a real company and again the figure shows four series of

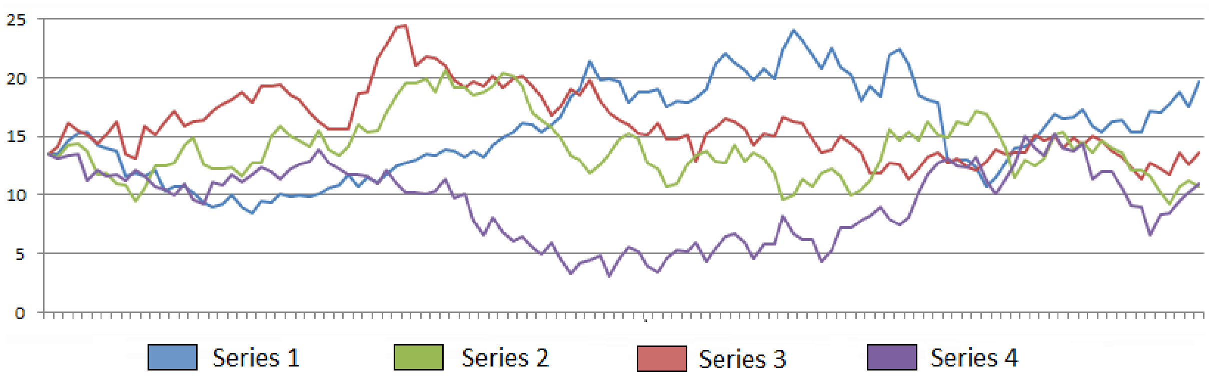


Figure 4.2: Random series versus real series: Series 1 in blue shows monthly prices of Repsol (Spain) over the period 2002 - 2011. The remaining series are modelled price time series by the Geometric Brownian Motion

indistinguishable behaviours.

The historical series follow apparently the random walk, however many studies argue against this idea, as we have exposed above. With the purpose to clarify experimentally this issue, in section 4.4 we will try to classify the real and modeled series with the only information provided by the price series itself by making use of a novel algorithm described in Section 8.2.1

4.3 Methodology

With the aim of showing that despite the seemingly random behaviour of the stock market, price movements have been built following different patterns than the random walk modelled series, we use a clusterisation procedure, which is capable to build hierarchical distance trees in a blind manner for a set of historical price series. We perform this task by an approximation to the not computable Kolmogorov complexity.

Cluster Analysis

The term cluster analysis (first used by [112]) encompasses different algorithms and methods for grouping objects of similar type into respective categories. The aim of clustering is to understand the macroscopic structure and relations between objects. Thus, clustering focuses on segmenting the complete set of information in homogeneous subgroups.

Clustering could be considered the most important unsupervised learning problem. In other words, cluster analysis simply discovers structures in data without explaining why they exist.

Just like static data clustering, time series clustering requires a clustering algorithm or procedure to form clusters given a set of unlabelled data objects. However, unlike static data, the time series of a feature comprise values changed with time. Given a set of unlabelled time series, it is often desirable to determine groups of similar time series. These unlabelled time series could be monitoring data collected during different periods from a particular process or from more than one process. There are other techniques applied to time series, however clustering is one of the most frequently used. Furthermore, there is a trend of increasing activity in the researches related to the clustering of time series, although the number of studies in this topic is relatively scarce compared against those focusing on static data.

Time series data are a topic of interest because its presence in a large variety of areas, like science, engineering, business, finances or biological. For example, in the field of finances time series clustering have been used by chartists to examine stock market data, searching for certain shapes, which are thought to be indicative of a stock's future performance [18] [79].

Normalised Compression Distance

The relationship between computation, information, and randomness is studied in the field of algorithmic information theory. An important topic in this field is the Kolmogorov Complexity [59] of an object, that is, broadly speaking, the measure of computational resources needed to describe an object. The idea is that the complexity of an object can be seen as an absolute and objective quantification of the amount of information in it. Since the Kolmogorov complexity is not computable, there have been raised several distance approximation measures. These measures are based on the comparison of lengths of compressions, between the objects and one of the most well-known is the Normalised Compression distance (NCD).

The NCD is a compression-based similarity distance that determines the similarity in terms of information distance between pairs of objects according to the most dominant common features. In previous works [115], [9], [90], [110], [61] we have demonstrated that the NCD is a reliable tool for classification on a number of domains. Furthermore, NCD has been applied successfully in many areas, NCD concerns to the classification of genomes, music pieces, plagiarism of computer programs, image registration, letters phylogeny, protein structure comparison, genotyping, tumor subclassification, virus detection, etc.([21],[61],[97], [45], [6]).

In this chapter we use a novel version of the NCD to measure the differences among the prices series. The measure that we use was proposed by Contreras et. al. in [25] as an innovative approach to the Kolmogorov complexity that exploits the management of the different dictionaries used by a compressor to reduce the redundancy. The proposed distance, called mNCD (modified Normalised Compression Distance), is the following:

$$mNCD(x, y) = \frac{Max\{C(x|D_y), C(y|D_x)\}}{Max\{C(x|D_x), C(y|D_y)\}} \tag{4.3}$$

Where C represents a measurable way to approximate the Kolmogorov Complexity using a compressor program, C(i) is the compressed size of i and $C(i|D_j)$ means the compression size of i using the compression dictionary of j.

In the same way that songs of the same music style share patterns, or malicious software and virus share common features, we expect multiple price series can share patterns that are invisible at usual analysis. In fact, our approach is based on the assumption that the patterns formed in the random price series are more similar to each other than to the real price series. Thus, we expect that patterns shared by two price series will be translated into a high degree of similarity.

In order to discretise the real numbers that represents the price of each day and be able to calculate the NCD; we simplify the information transforming the prices in daily or monthly increments and assessing threshold levels in terms of prices investments instead of prices alone:

$$return = \frac{Price_{t+1}}{Price_t} \tag{4.4}$$

Finally, we obtain series of daily returns in the next way:

$$Day(return) \begin{cases} return < 0.8 = "A" \\ 0.8 < return < 0.9 = "B" \\ 0.9 < return < 0.95 = "C" \\ 0.95 < return < 1.0 = "D" \\ 1.0 < return < 1.05 = "E" \\ 1.05 < return < 1.1 = "F" \\ 1.1 < return < 1.2 = "G" \\ return > 1.2 = "H" \end{cases} \tag{4.5}$$

Hierarchical Clustering

We can find different classifications, depending on the criteria used to define the types of clustering algorithms [53]. To show the degree of similarity of the price series, we perform hierarchical clusterings in which the historical series of prices are regrouped in a tree structure in an automatic and blind manner. The so-called hierarchical methods produce nested partitions and are represented by dendrograms. Thus, we consider a set of N price time series to be clustered and a Distance Matrix, also called dis-similarity matrix, with N*N measurements. The employed hierarchical clustering process, defined in [55], can be summarized as a method that builds a binary tree from individual elements by progressively merging the clusters containing the two closest elements (according to the Distance Matrix). This algorithm is able to divide any cluster further to observe its underlying structure. The specific type of hierarchical algorithm chosen in this work to perform the clustering is the complete linkage method [29] due to compromise between simplicity, ease of analysis and its ability to obtain quality solutions. Although other non- hierarchical methods are also possible, they will not be discussed in this work.

4.4 Experimental results: Real and modelled price time series clusterings

In the next experiments we use undirected graphs using a spring model in order to be able to show large graphs. These graphs are based on the force-directed approach [43] and show the results of binary hierarchical trees building by the mNCD. Each leaf node is linked with a specific price time series. The remaining nodes (branch nodes) are intermediate nodes which measure the distances between each branch. Thus, the smaller distance between them, the higher similarity among leafs.

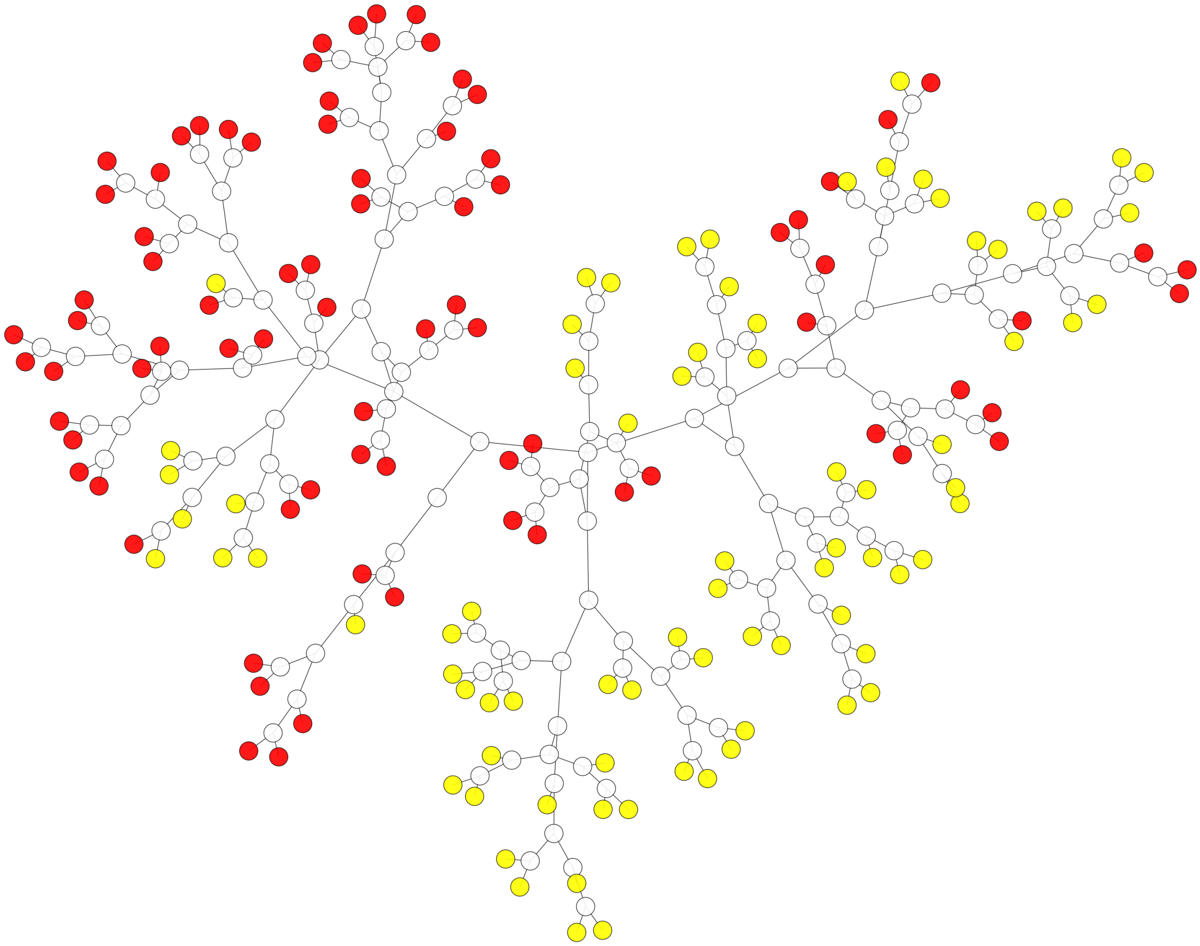


Figure 4.3: Random versus real price time series: Clustering of the monthly prices of 80 companies listed in Europe over the period 2001-2012, and 80 price time series modelled by the GBM

Figure 8.3 presents the clusterisation between monthly series of real and modelled prices. Red nodes represent the modelled prices time series and yellow nodes the real price time series of the companies. The results of the clustering in this sample display significantly clusters of the two types of series. However, there are price series misclassified belonging to the two types of source. If we divide the graph in two sections: first in the left side we obtain a 13,43% (9 yellow nodes) of misclassification and a 86,57% (58 red nodes) of success. In the right side we obtain a 23,65% (22 red nodes) of misclassification and a 76,35% (71 yellow nodes) of success. The periods of the used price time series are composed by 10 years, therefore the lower number of observations (120 months) and the misclassification errors could be correlated.

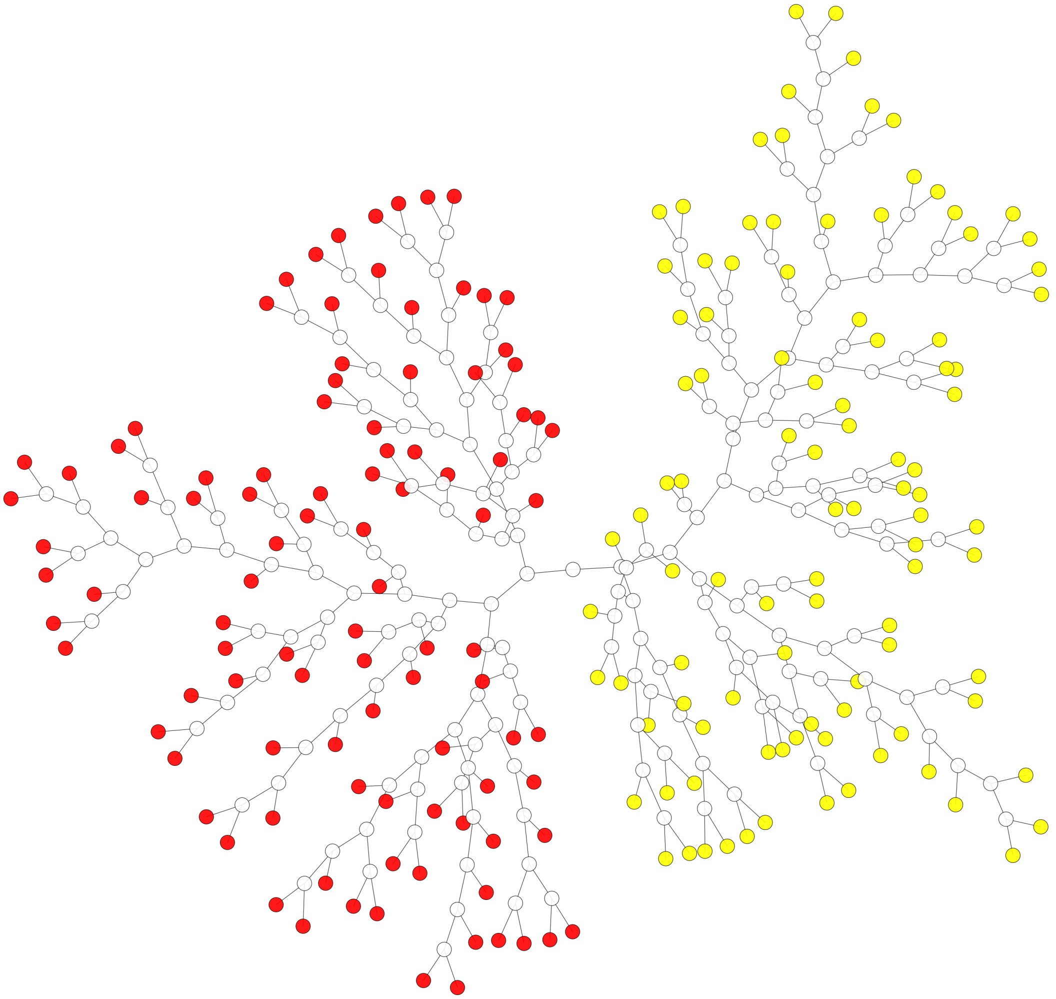


Figure 4.4: Random versus real time price series: Clustering of the daily prices of 80 companies listed in Europe over the period 2001-2012, and 80 price time series modeled by the GBM

Figure 8.4 presents the clusterisation between daily series of real and modelled prices. Instead of the clusterisation showed in the last graph, Figure 8.4 shows a clear division between the two type of series. The red nodes belong to one side of the graph and the yellow ones to the other side. Thus, the clustering performed is able to differentiate successfully the two types of series in a blind manner. As we commented above, the total success of this graph could be thanks to the amount of observations, several thousands of daily observations against one hundred of monthly observations. The patterns found by the compressor are very different in both cases, although we can not identify them in a usual way. The way to build the modelled series with the GBM (following a random walk) is different to the way that the stock market build the real price series of the companies. With this method we identify that stock prices show common patters that clearly differ from random walks driving to the conclusion that stock market prices do not follow the same path than random walk prices. Likewise some natural phenomena present common patterns of weather (for instance hurricanes) or some disorders exhibit common genetic patterns, stock prices also share some common patterns.

4.5 Undercover Patterns Among Industries

Macroeconomics analysis is the method which studies the performance, structure, behaviour, and decision-making of the economy as a whole. Instead of focus on individuals and how they make economic decisions. This type of analysis is considered very complex due the great amount of factors that influence it.

We present evidence that stock prices of companies belonging to different industries differ also on their patters. The breakdown of our analysis by industries is in line with the arguments exposed by Moskowitz and Grinblatt in [81], who underline the important role for industries in understanding financial markets, "firms within an industry tend to be highly correlated; they operate in the same regulatory environment, exhibit similar behaviour in the corporate finance arena, are similarly sensitive to macroeconomic shocks, and are exposed to similar supply and demand fluctuations" (pp 1288).

Although earlier studies have shown relatively little impact of industries on asset prices, with the exception of Roll [95] who concluded that a significant portion of the correlations among country returns was induced by the industrial compositions of the country indexes, part of the more recent financial literature has highlighted the specialization and segmentation of equity markets: Tripati in [91] find that different sectors behave differently when it comes to the risk-reward relationship in the stock market. Menzly and Ozbas in [66] claim that among the analysts who are the most important producers of information, there is specialization along industries. They find evidence that, firm and industry level returns are cross-predictable based on lagged returns in supplier and customer industries. In [48], Hong point out that some industries are related to the macroeconomic activity in an opposite way, that is, performance in some industries is positively (negatively) cross-serially correlated with the economic activity. For instance, high stock returns for some industries like retail mean good news for current or future economic activity, while high returns for other industries such as petroleum could mean just the opposite.

Consequently, the premise that guides our search for common and uncommon patterns in stock prices among industries is based on this recent branch of the literature. Although these common or uncommon patterns are not always easy to identify with a visual inspection, more sophisticated algorithms as the one we propose here are able to detect them. In addition, by examining uncommon patterns among stock prices of different industries, we provide direct evidence on the plausible predictability of information about fundamentals. In this way this work goes beyond the weak form of the EMH to the borders of the semi-strong form of the EMH. More generally, our empirical framework provides a novel measure of common and uncommon patterns that can be used to expand existing work on other forms of return predictability.

Therefore in this section we will use our methodology for searching similarities and patterns among companies belonging to common business sectors. Thus, we analyse whether the companies belonging to business sectors that perform similar tasks, remain with behaviors alike than those companies that perform tasks in different sectors.

The stock market is composed by many companies engaged in diverse activities. Events like the current recession, the "boom" of the brick, the rising cost of oil, the droughts, etc. differently affect each of them. However, there are companies working more correlated with each other. That is, the trends and behaviours of BMW and Volkswagen are more dependent on each other than those that BMW itself could have with Telefónica. In order to perform the test, the different activities are divided in the called sectors. An industry group or sector consists of companies in closely related businesses such as mining and quarrying, construction, agriculture, telecommunications, etc. It can be argued that if the companies are independent companies, all stocks will trade independently from each other. However the stocks within a sector tend to move together because companies within the same industry group are affected in similar ways by market and economic conditions. Actually, no matter what the overall market is doing, we always can find some industry sector moving up, and others heading down. The fact is that individual stocks follow other stock in the same sector over 70% of the time [84].

The stock sector analysis is considered one of the best approaches from a long term perspective to invest. Nowadays we can find a lot of tools and institutions performing this type of analysis (`http://Briefing.com` , `http://www.bloomberg.com/news-research/industries/` , `http://biz.yahoo.com/p/`). We try to analyse this behaviour and we check if the most close sectors share similar patterns. We could demonstrate the links among the companies in the same sector and similar sectors. Furthermore, if we can find this type of relation between companies of the same sector or even between different sectors, we may say that the similar or even the same strategies about buying and selling shares in these sectors will have a similar performance results. So, if we get a good strategy for a given sector we could expand this strategy to other sectors of the same behaviour. On these grounds it is very interesting and enlightening to demonstrate these relations among price series. To this end we will perform a series of experiments using the methodology presented in Section 8.2.1.

We set a database of the active listed companies of Europe over the period 2001-2013 and with at least 5 last years of historical data in the stock market. We split the companies of our data base in different sectors according to NACE Rev. 2 classification. The Statistical classification of economic activities in the European Community, abbreviated as NACE, is the nomenclature of economic activities in the European Union (EU); This classification is based on statistical units corresponding to a specific economic activity (or group of similar activities). The proposed division is quite general, there are companies settled in the same group performing different activities and with big differences about its impact in the stock market. Due to the general classification used and the complexity related to cluster companies with uneven impact factors , we do not expect perfect clusterings as in the experiments presented before. However we look for another formations as little clusters or clusters with tendencies to one particular activity.

4.6 Experimental Results: Clusterings of Price Series by Sectors

The experiments performed in this section have been carried out with the same methodology used in the last experiments: first we use the mNCD to calculate the similitude between pairs of price series. Second we perform a hierarchical clustering with the values obtained. And in the last place, we build suitable graphs to show the results.

Figure 8.5 shows instances of companies working in sectors with unrelated occupations. The figure is composed by three sub-figures (graphs A, B and C). Each graph shows a clustering of two different industrial activities. The graph 8.5.A shows a clustering of companies related to extractive industries and oil refining, and the companies related to the manufacture of food, beverages and tobacco, green and blue respectively. The graph 8.5.B shows a clustering of companies related to telecommunications and information technologies, and the companies related to the manufacture of sewing and textile, gray and blue. The graph 8.5.C shows a clustering of companies related to the supply and management of gas , electricity, water and waste, and the companies related to the manufacture of machinery, equipment, vehicles and transports, pink and black respectively. Furthermore the root node of each tree is yellow.

In this three graphs we can observe some degree of clusterisation. Broadly speaking, it seems that companies performing different economic activities generate price series that do not keep the same similarity between them than the firms that act within the same economic sector. With the purpose to probe the success of our clustering, we split the tree in subgroups (Sub-clusters). We divide the tree beginning in the level 0 (the root), thus obtaining two sub-cluster. With the aim of obtaining a balanced sub-clusters, we continue splitting the tree in next levels until we obtain sub-clusters with a number of companies (leafs) lower than the number of companies of the bigger sector. Figure 8.5 shows the sub-clusters shaded and numbered. In this case, all the graphs are divided in three sub-clusters. Although it is not the best method to split the cluster, we obtain a meaningful division of the companies. Table 8.1 shows a summary of the results obtained showing two values: the number of companies of a particular sector in each cluster, and the percent of each sector in the clusters. Both values are important to assess the clusters due to the uncorrelated sizes. Thus, for example we can see that although the sub-cluster 3 in the graph 8.5.C achieves

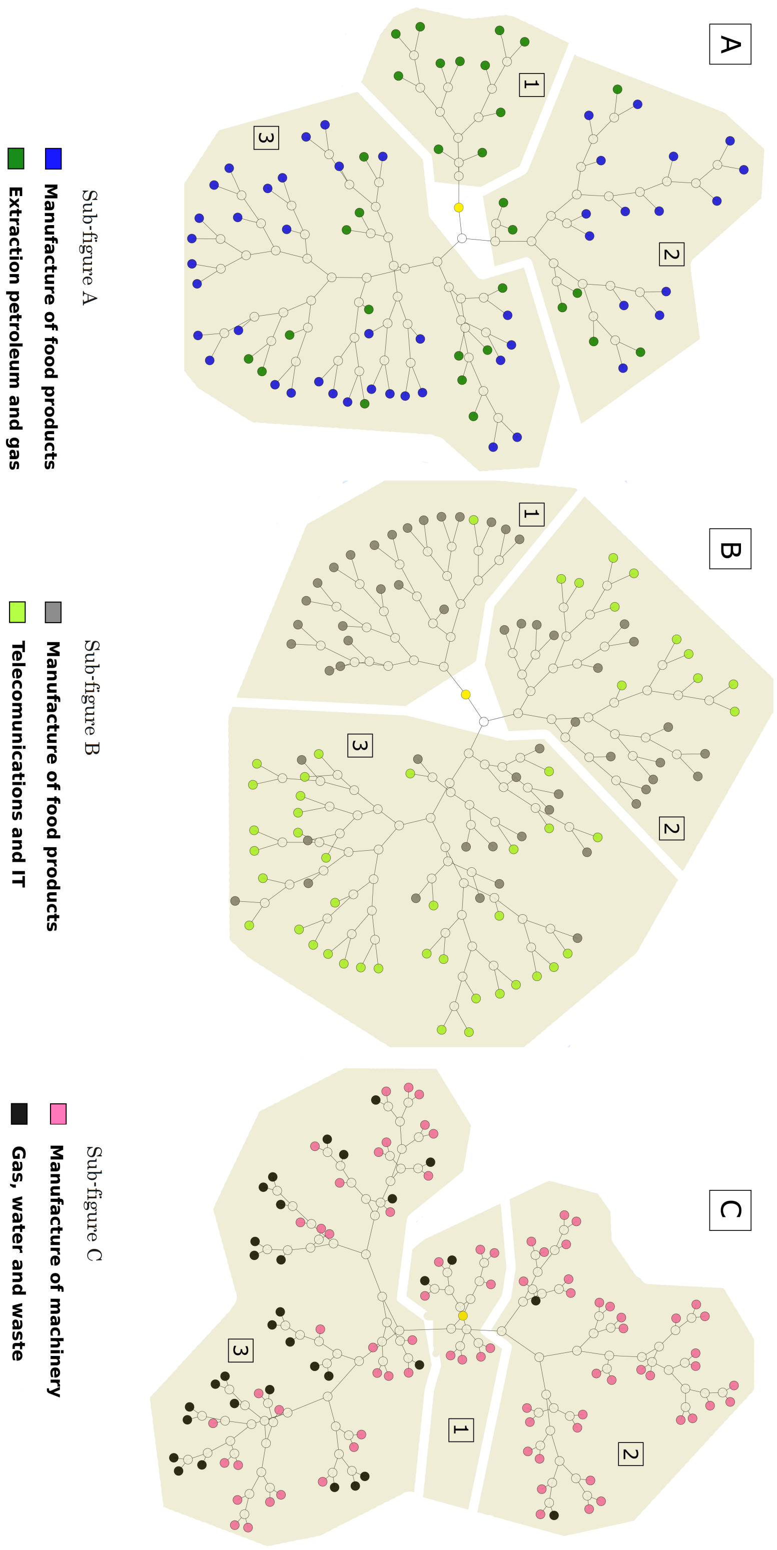


Figure 4.5: Clusters among companies operating in different economic activities.

only a 47% of black nodes, the same cluster includes 29 of 33 black nodes.

Table 4.1: Percents of success and number of companies per cluster of Figure 8.5

Graph		**A**		**B**		**C**	
Sector (Color)		**Green**	**Blue**	**Gray**	**Green**	**Pink**	**Black**
Total Companies		48	31	54	48	79	33
Sub-Cluster 1	Companies	11	0	22	1	9	2
	Percent	**100%**	**0%**	**96%**	**4%**	**82%**	**18%**
Sub-Cluster 2	Companies	7	15	16	11	37	2
	Percent	**31%**	**69%**	**60%**	**40%**	**95%**	**5%**
Sub-Cluster 3	Companies	13	33	16	36	33	29
	Percent	**29%**	**71%**	**31%**	**69%**	**53%**	**47%**

We can state that even with the low detail and the extreme generalization of the NACE classification, the system can roughly group the companies that develop similar economic activities only using historical prices time series. Furthermore, we want to note that the used classification is too general, and we could attribute to the same sector the ability to generate sub-activities, i.e the same industry can be divided into smaller groups that behave in a different ways to each other.

In order to face the results, we perform the same experiments in sectors developing similar activities. Figure 4.6 shows instances of companies working in sectors developing similar activities. The figure is divided on two sub-figures, each one linked to one graph (A and B). Each graph shows a clustering of two sectors. The first graph 4.6.A shows a clustering of companies related to real estate activities, and the companies related to the construction of buildings and civil engineering, gray and blue respectively. And the second graph 4.6.B shows the clustering of companies related to the manufacture of metallurgical products in blue colour, and companies related to the manufacture of machinery, equipment and vehicles in pink color.

Following the same methodology as in the previous experiment, we split the graphs to measure the clusters obtained. Thus, we obtain three sub-clusters in the Figure 4.6.A and four sub-clusters in Figure 4.6.B . We show the results in Table 4.2. Results show companies of different sectors scattered over the graphs, as we also can observe visually in the Figure 4.6 where the two graphs show a very low degree of clusterisation.

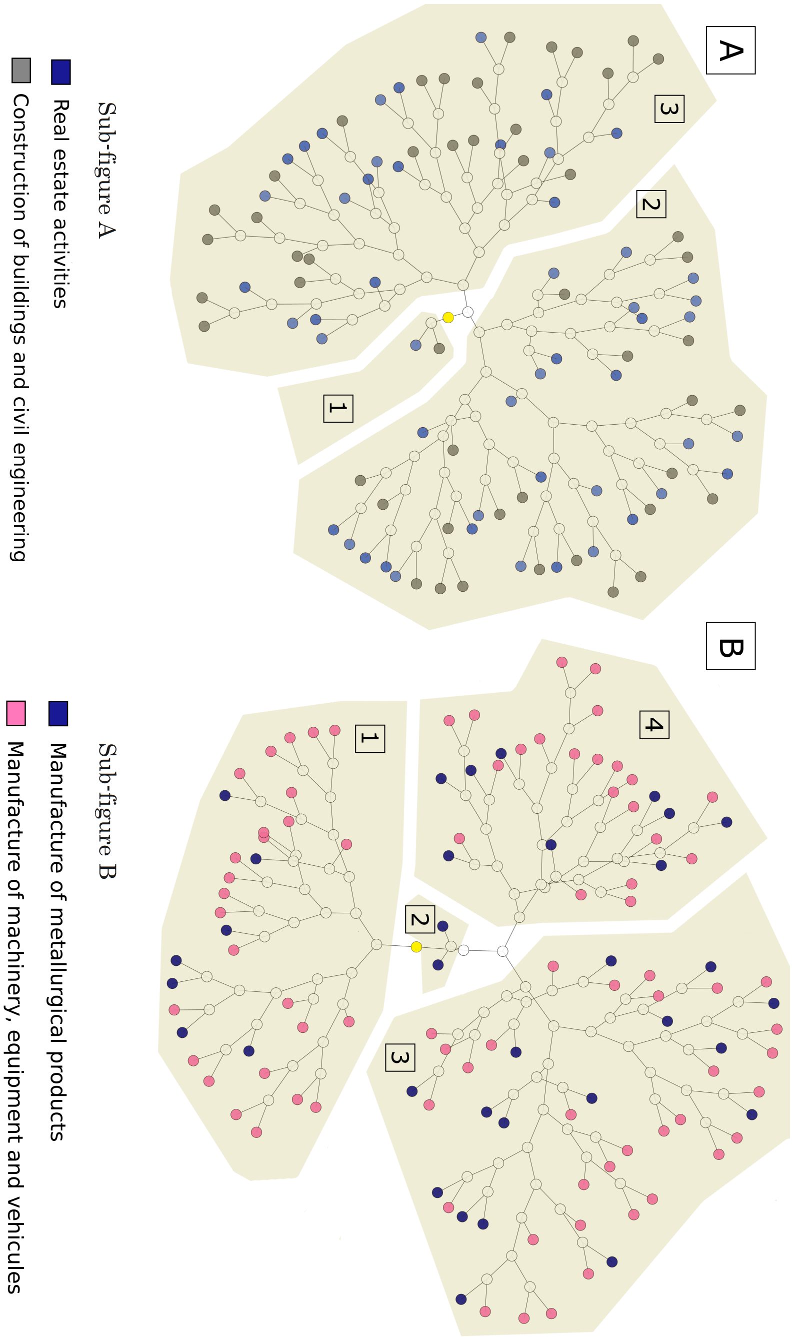

A
1
2
3
Sub-figure A
Real estate activities
Construction of buildings and civil engineering
B
1
2
3
4
Sub-figure B
Manufacture of metallurgical products
Manufacture of machinery, equipment and vehicules

Figure 4.6: Clusters among companies operating in similar economic activities.

Table 4.2: Percents of success and number of companies per cluster of Figure 4.6

Graph		A		B	
Sector (Color)		**Blue**	**Gray**	**Blue**	**Pink**
Total Companies		51	49	34	78
Sub-Cluster 1	Companies	1	1	7	26
	Percent	**50**	**50**	**21%**	**79%**
Sub-Cluster 2	Companies	29	23	2	0
	Percent	**56%**	**44%**	**100%**	**0%**
Sub-Cluster 3	Companies	21	25	16	31
	Percent	**46%**	**54%**	**34%**	**66%**
Sub-Cluster 4	Companies	–	–	9	21
	Percent	–	–	**30%**	**70%**

Chapter Conclusions This chapter delves into the behavioural models and in particular into the random walk model. After we analysed the background and related work, we have presented the next two contributions.

- First, we have modelled time series of prices following a Random Walk and we have faced them against real time series of historical stock market prices. We have shown that the series are indistinguishable in a simple way, nevertheless we have found disguised patterns in the series. Using a novel clustering methodology we have demonstrate that modelled series and real series are fully distinguishable. Therefore, we could use a time series prices of real companies as source of information.

- Second, we have used the same methodology showing connexions between clusters of companies developing alike industrial activities. Thus, we have pointed how the companies belonging to the same industrial sectors of the stock markets are linked in somehow. Therefore, a macroeconomic analysis to select group of companies could be useful to achieve better profits.

Chapter 5

An Automated Trading System Based on a Genetic Algorithm

As we widely explained in Chapter 2 there are two types of financial analysis oriented to design trading systems: fundamental and technical. And as we also mention in the same chapter, the use of both types of analysis is not necessarily exclusive. In this chapter we propose the use of a GA as engine of an ATS. As we show in 3, GAs are advantageous and quality tools to optimise countless number of applications and problems. In this chapter we propose a trading system based on a GA that performs operations in an exchange optimising the parameters related with ratios and indicators of Fundamental and Technical analysis. The aim of this approach is to obtain a set of trading signals per each evaluated security indicating the most suitable procedure for the next operation, *buy*, *sell* or *keep*.

Section 5.1 describes implementation details of the trading system and the proposed methodology including a novel operator called Filling operator (Section sec:GAwFO) which helps to preserve the diversity of the population and solves the problem of premature convergence. The section includes experimental results evaluating the profitability of the system with real data and testing the implemented improvements. The results indicate that the proposed methodology is positive in both terms of quality of the solutions and in terms of the convergence of the algorithm. In this chapter we also propose to combine the use of two parallel computer architectures to speed up the functioning of the GA proposed. First, we have used a corporative grid in Section 5.2 , and second, a graphics device in Section 5.3. Each section includes implementations details related to both platforms and experimental results which show how the combination of GAs and parallel architectures allows us to obtain solutions for real time (or intra-day data) investment decision. Finally, we conclude the chapter in Section 5.4 where we compare the two implemented parallel architectures.

5.1 Trading System Based on Genetic Algorithms: Implementation

In this section we propose the use of GAs to optimize jointly the parameters of FA and TA. We have implemented two GA versions:

- A classical and simple GA (sGA) with standard operators.
- A optimised GA with a new operator, namely Filling Operator (GAwFO).

First we implemented a classic version of a GA. Although the results are acceptable, a well-known problem can arise with the use of sGAs, it is known as premature convergence. Premature convergence is likely to occur when an individual is fitter the most of its competitors by a significant difference and it is found in a early stage of the execution. Due to its fitness value the algorithm tends to transfer the genetic code of the individual to the entire population in a few generations undermining the diversity of the population too quickly. The algorithm converge to a local optimum that represents the fitter individual. This would not be a problem if the solution was good enough, however, if the algorithm has not been left to evolve sufficiently the solution also could not be. Different techniques exist to prevent this. Usually, the most common methods implemented by researchers in GAs for solving this problem involve controlling the selective force or to increase the size of the population (it has a high computational cost in our case). There are also strategies for regeneration, a mating strategy called incest prevention, to increase the value of the probability of mutation, etc. In order to preserve the diversity of the population we propose a GA with a Filling operator (GAwFO). Basically, the GAwFO approach consists of a sGA with a modification of the selection and crossover operators, and as a consequence of this change it applies an operator driven to the generation of new random individuals.

5.1.1 Automatic Trading System functioning

One solution is given by a set of values, which indicate weights, parameters and thresholds for a selected set of technical and fundamental indicators. The operative of the ATS works as follows:

1. The investor selects a set of Technical Indicators for TA
2. The investor selects a set of Fundamental ratios for FA
3. Establish $Threshold_{buy}$ and $Threshold_{sell}$ ranges and $Weights$ ranges
4. For each company i
    (a) Apply GAwFO over a period of X years to obtain the solution A
    (b) Apply TA using the parameters given by A to the target Days
        i. For each indicator I_j calculate the signals S_js (Buy=1, Sell=−1, Neutral=0)
    (c) Apply FA using the thresholds given by A to the target year
        i. For each ratio R_j calculate the signals S_js (Buy=1, Sell=−1, Neutral=0)
    (d) Compute the Raw Trading System Signals by adding the indicators signals weighted by their weights $RTS_s = \sum_{j=1}^{n} S_j \cdot W_j$
    (e) Compute the Net Trading System Signal choosing values for X and Y as follows:
        i. If $RTS_s = or > Threshold_{buy}$ then $TS_s = 1$
        ii. elseif $RTS_s = or < Threshold_{sell}$ then $TS_s = -1$
        iii. else $RTS_s = 0$
    (f) Compute the profit given by the Trading System

The solution provides a combination value of fundamental and technical trading signals per each day of investment. This value is compared versus the $Threshold_{buy}$ and $Threshold_{sell}$. The thresholds must be previously established in the system. If the value exceeds the value of the $Threshold_{buy}$, the system will give a buy signal. Otherwise, if the value is lower than the $Threshold_{sell}$, the system provides a sell signal. Finally, if the value is between both thresholds, the system provides a neutral signal. The thresholds can be increased or decreased to profile the system behavior. Therefore, at low ranges the program will be more sensitive to the trading signals becoming more aggressive. However, the program will be less sensitive at high ranges becoming conservative.

TA is formulated by four Technical Indicators (TI) selected by the investor and four weights corresponding to each indicator (W_1, W_2, W_3 and W_4) . In this chapter the selected technical indicators are the MA, the VPCI, the RSID and the SR presented in Chapter 2.

Each one of these technical indicators gives us a signal of buy, sell or neutral. The values of W_1, W_2, W_3 and W_4 weigh the importance of each indicator in obtaining buy or sell signals. FA uses also four fundamental ratios and four weights linked to each indicator (W_5, W_6, W_7 and W_8). We use the the fundamental ratios $Ratio_0, Ratio_3, Ratio_9, Ratio_{10}$ explained in Chapter 2. Those indicators have been selected following the historical usefulness in the literature on investments (we refer the interested reader to [7],[10],[14],[17],[40],[38] and [93]).

5.1.2 Genetic Algorithm based on the Filling Operator

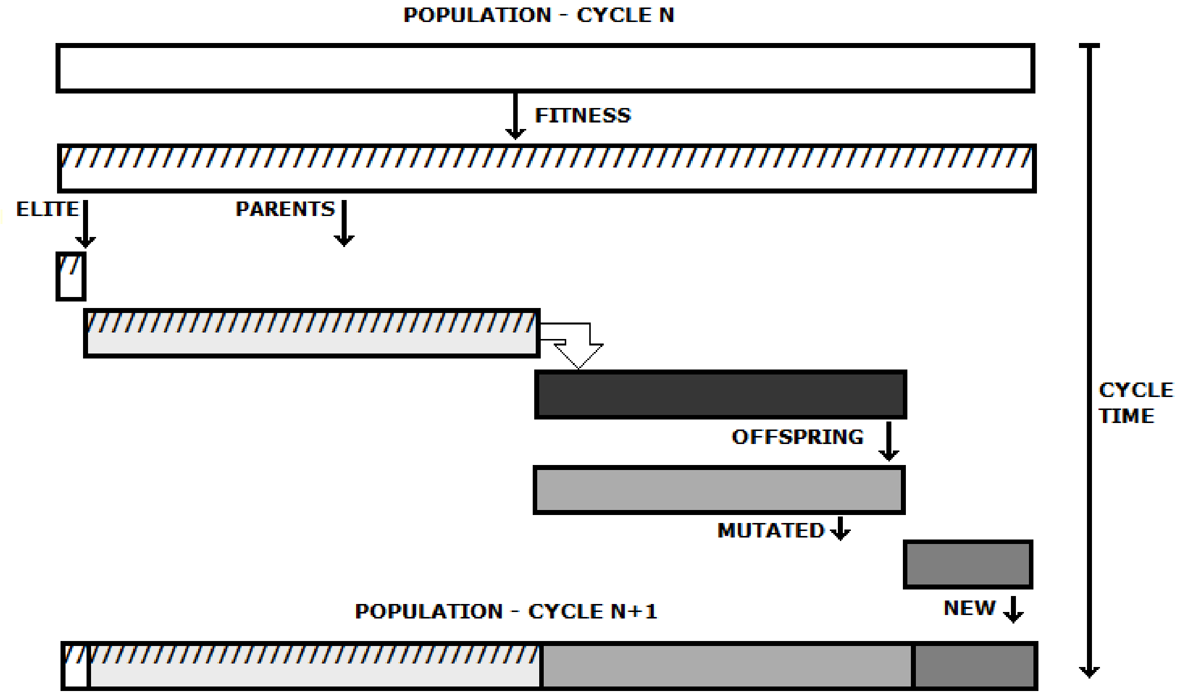
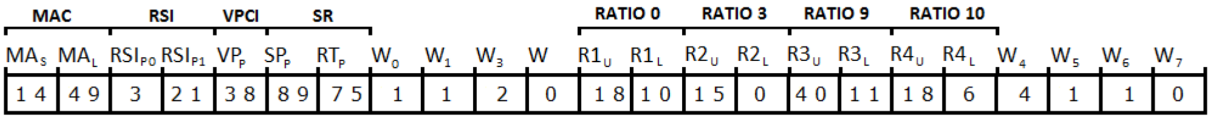


Figure 5.1: General work-flow of the GAwFO in a generation (cycle) N: evaluation, elitism, crossover, mutation and filling operator

The implemented algorithm takes advantage of the operator GAwFO. The main advantages of GAwFO is to preserve the diversity of the population solving the possibility of a premature convergence. In addition GAwFO uses a particular structure of the population specifically designed to facilitate the implementation on GPUs and to reduce the number of evaluations. Next, we present the basic flow of GAwFO represented in Figure 5.1.

- The algorithm starts from a population of n individuals obtained in previous generations or generated randomly in first generation.
- The fitness function evaluates the population and selects the best individual which is preserved throughout the execution without transformations (elitism).
- The AG selects by tournament $s = (n/2) - 1$ individuals which join for creating the rest of the population.
- With this s individuals, the algorithm applies a version of an uniform crossover operator (detailed in next paragraphs) with a crossover probability (p_c), thus generating m new individuals. Since $p_c < 1$, in most generations we will get $m < (n/2)$. The crossover operator provides two main features:
  - First, the generated offspring is probabilistically more similar to the parent which has higher fitness value, that is the new individual can obtain more genes from the best parent.
  - Second, the crossovers leading to already existing individuals are not allowed. Therefore, the more convergence in the population, the fewer individuals generated by the crossover operator.
- In order to preserve the size of the population (n individuals) we will need to generate $k = n - s - m$ individuals. The algorithm generates k random individuals in each generation (note that k is not a constant value for all generations, as it depends on the number of crossings made by the crossover operator).
- On this new population of $n = 1 + s + m + k$ individuals we apply a classical mutation operator with probability p_m only to the offspring of m individuals.
- With a new population fully generated, the GA steps back to the second point repeating the process until the number of generations will be reach.



MAC		RSI		VPCI	SR						RATIO 0		RATIO 3		RATIO 9		RATIO 10					
MA_S	MA_L	RSI_{P0}	RSI_{P1}	VP_P	SP_P	RT_P	W_0	W_1	W_3	W	$R1_U$	$R1_L$	$R2_U$	$R2_L$	$R3_U$	$R3_L$	$R4_U$	$R4_L$	W_4	W_5	W_6	W_7
14	49	3	21	38	89	75	1	1	2	0	18	10	15	0	40	11	18	6	4	1	1	0

Figure 5.2: Genetic encoding: Chromosome example for the GAwFO

As can be deduced from the previous explanation, this implementation does not use effectively the entire size of the population, since there are s individuals which do not vary for a generation. However, individuals do transmit their information to the next generation because they are precisely those involved in the crossover. Given the nature of the problem this can be useful to get several solutions. Furthermore this issue provide two more advantages. First we obtain a reduction of the computation time, being that it is not necessary to evaluate $n/2$ of the population in each generation. Second, the structure of the population is also interesting for future implementations on GPU architectures that allow for faster execution.

Table 5.1: Chromosome codification: parameters, ranges and decodification process

Gene	**Name**		**Range**			**Example**	
#	**Short**	**Long**	**Lower**	**Upper**	**Jumps**	**Coded**	**Decoded**
1	$\mathbf{MA_S}$	Short MA	1	50	1	14	15
2	$\mathbf{MA_L}$	Long MA	51	150	1	49	100
3	$\mathbf{RSI_{P0}}$	RSI period	1	50	1	3	4
4	$\mathbf{RSI_{P1}}$	RSI MA period	1	50	1	21	22
5	$\mathbf{VP_p}$	VPCI period	1	50	1	38	39
6	$\mathbf{SP_p}$	Support period	1	100	1	89	90
7	$\mathbf{RT_p}$	Resistance period	1	100	1	75	76
8	$\mathbf{W_1}$	Weight MA	0	4	1	1	1
9	$\mathbf{W_2}$	Weight RSI	0	4	1	1	1
10	$\mathbf{W_3}$	Weight VPCI	0	4	1	2	2
11	$\mathbf{W_4}$	Weight SR	0	4	1	0	0
12	$\mathbf{R0_U}$	Ratio$_0$: $Threshold_{buy}$	9	18	0.5	18	18
13	$\mathbf{R0_L}$	Ratio$_0$: $Threshold_{sell}$	20	29	0.5	10	25
14	$\mathbf{R1_U}$	Ratio$_3$: $Threshold_{buy}$	1.5	3	0.1	15	3
15	$\mathbf{R1_L}$	Ratio$_3$: $Threshold_{sell}$	3.25	4.75	0.1	0	3.25
16	$\mathbf{R2_U}$	Ratio$_9$: $Threshold_{buy}$	6	10	0.1	40	10
17	$\mathbf{R2_L}$	Ratio$_9$: $Threshold_{sell}$	1	5	0.1	11	2.1
18	$\mathbf{R3_U}$	Ratio$_{10}$: $Threshold_{buy}$	6	10	0.1	18	7.8
19	$\mathbf{R3_L}$	Ratio$_{10}$: $Threshold_{sell}$	1	5	0.1	6	1.6
20	$\mathbf{W_5}$	Weight Ratio$_0$	0	4	1	4	4
21	$\mathbf{W_6}$	Weight Ratio$_3$	0	4	1	1	1
22	$\mathbf{W_7}$	Weight Ratio$_9$	0	4	1	1	1
23	$\mathbf{W_8}$	Weight Ratio$_{10}$	0	4	1	0	0

We use an integer codification with a chromosome with a total of 23 genes separated into two parts. The first 11 genes represent values that are interpreted by TA and the remainder (12 genes) affect FA indicators. Using the set of technical and fundamental indicators (previously selected by the investor) with the encoded values, we get a signal to buy, sell or remain inactive for each of the companies. With all the indicators (TA and FA) we obtain a value S_i for buying and selling (for example if we obtain 3 buying an 4 selling signals, the value will be $S_i = 3 - 4 = -1$. Once we have the sum of all trading signals, we use a specific range to buy, sell or do nothing ($Threshold_{buy}$ and $Threshold_{sell}$). The higher the absolute module of the thresholds are, the more conservative the trading system will be (less movements of trading). Then we compute the benefit of buying and selling following the signals indicated by the chromosome.

Figure 5.2 represents an example of the chromosome of a member of the population. The example is decoded on Table 5.1 which shows the ranges of each indicator [**Lower**,**Upper**] and the accuracy into the range (the number of possible **Jumps** into the range). Thus, Table 5.1 outlines the parameters and presents a decodification example related to Figure 5.2. For example, the value of the gen MA_L is 49, however the range of the indicator is from 51 to 150 in steps of 1 , therefore we obtain the decoded valules in next way:

$$\mathbf{Decoded}_{value} : (\mathbf{Coded}_{value} * \mathbf{Jumps} + \mathbf{Lower}_{Range}) = 100$$

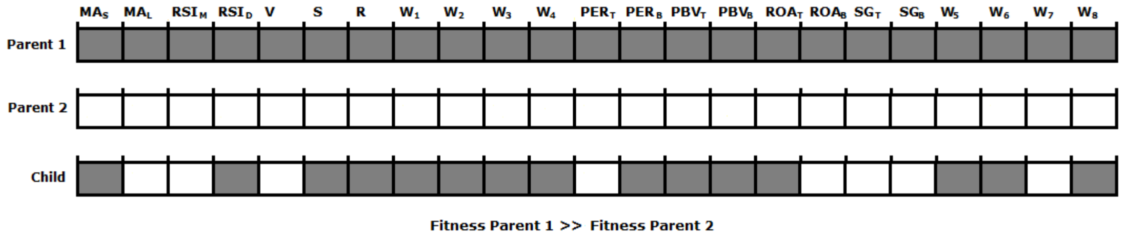


Figure 5.3: Operation of crossover with dominance

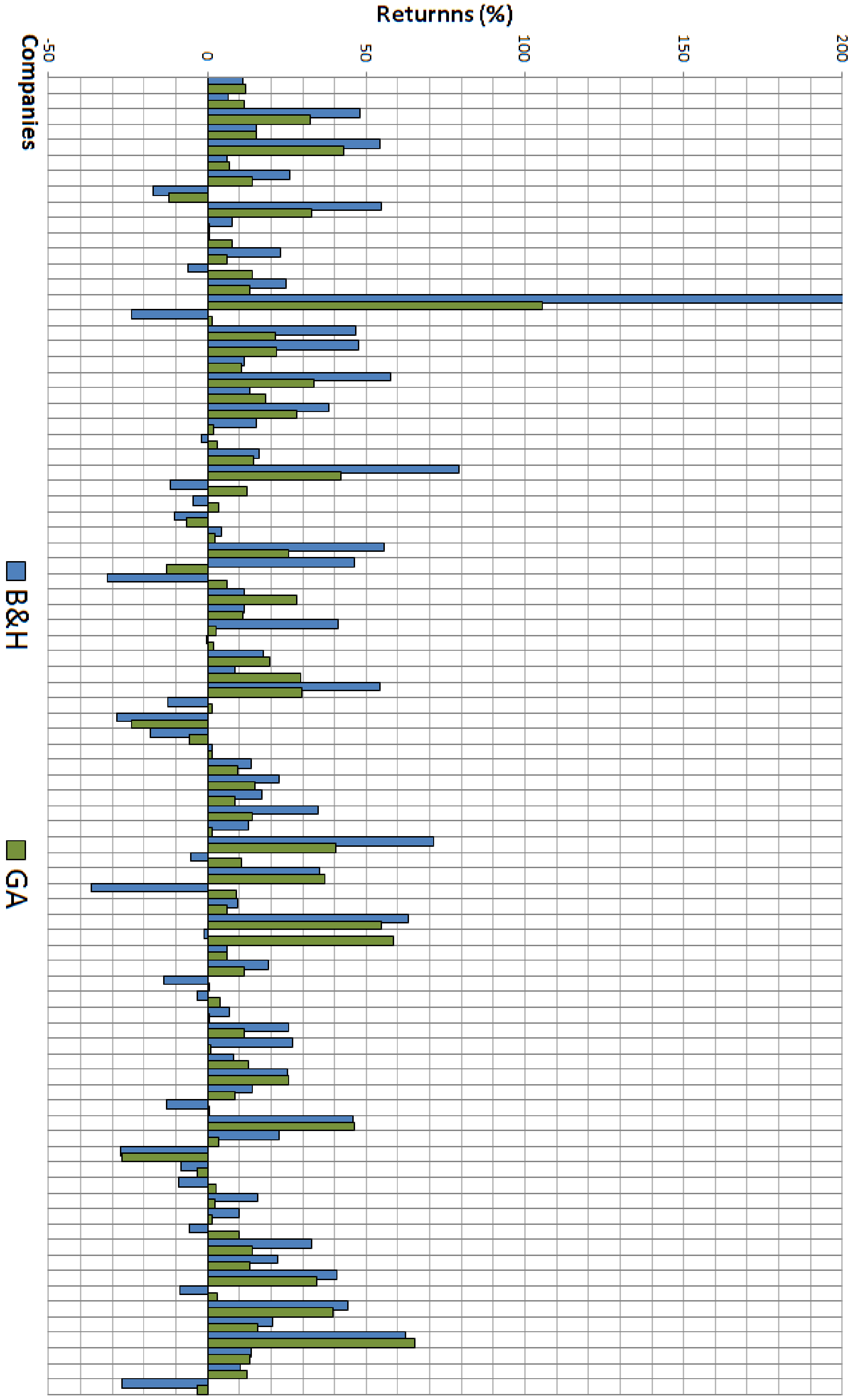


Figure 5.4: Experimental Results for 85 companies listed in the S&P 500. Average of the accumulated return for the B&H =17.47 and for the ATS based on GAwFO (average of 20 runs)=14.31.

The crossover follows a modified implementation of the Uniform Point Crossover (UPX)[1]. The implemented operator provides a special feature to select the inherited genes. The generated offspring will be more similar to the parent with higher fitness value. Thus, those that provide to the individual the greater fitness are more likely to be inherited to offspring. Following the genetic inspiration we called them dominant genes. For example, if a particular individual has a combination of three genes which provide to the individual a higher fitness value, these dominant genes have higher probabilities of being transferred to next generations by the crossover operator. The crossover operator can exchange any number of variables, in any order. We show an example in Figure 5.3.

We use a classic mutation operator which replaces the value of a random gene with another random value. The fitness function used to evaluate an individual is the AR, the accumulated return obtained as it is described in the Chapter 3, Equation 8.7. It is also possible to use a fitness function based on return and risk as the Sharpe ratio, however at this point of the thsis we considered that technical indicators are supposed to control risk in some way (e.g. break of support levels).

5.1.3 Experimental Results: Automated Trading System Based on Genetic Algorithms

The experimental results has been obtained on a Intel i5 processor running at 2,67 GHz under Windows7 and with a memory capacity of 4GB RAM. The data represented in the figures are the arithmetic average of 20 runs. First, we test our ATS with real data to measure its financial results, and second, we perform several experiments with the aim to evaluate the proficiency of the GAwFO.

Figure 8.6 presents the financial results. GAwFO and sGA were executed with a sample of 85 companies included in the S&P 500 Index Constituent List during the period January 1994 to December 2003. We apply FA and TA with the solutions obtained during the year 2004. The data provides around 35000 observations for quarterly fundamental data and 41000 observations for monthly technical data. The source of the data are Compustat and CRSP databases. The ATS reaches high returns and minimises the losses of the B&H strategy, even, it turns the losses into gains. However our trading system rarely exceeds

[1] Genes are randomly copied from the first or from the second parent [108]

the highest returns given by the market itself, maybe due to the good financial year where investments are performed. Average returns of the ATS reached a 14.31% with 8 negative and 77 positive operations, while the average returns of the B%H strategy reaches a 17.37% profit with 24 negative and 61 positive investments.. It is also important to note, that a neutral signal do not provide any return (as we can see in equation 8.7). That issue represents a disadvantage with respect to the practitioners or other trading systems. This is because usually the return of a neutral signal is a risk-free return given by the country target of the analysis (e.g the US Treasury Bills of EE.UU).

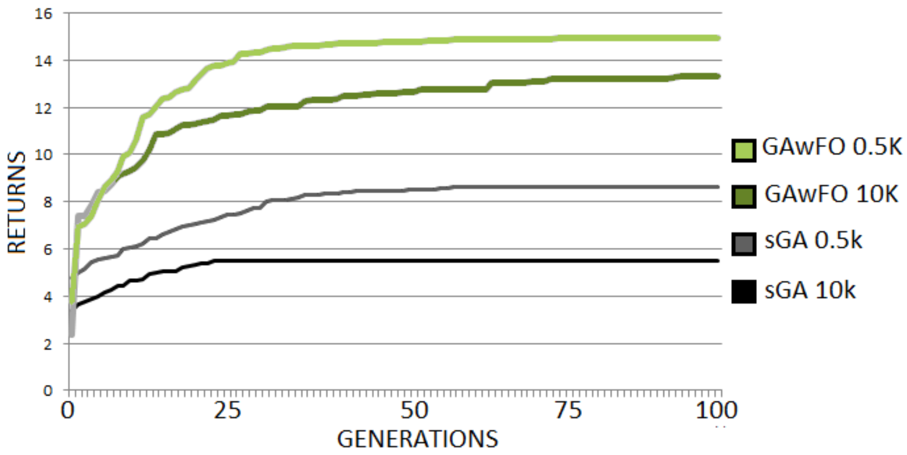


Figure 5.5: Convergency of the algorithm (profit of company ID1356) for GAwFO and sGA for 100 generations and several amounts of individuals

Regarding the convergence of the GAs we perform two experiments over one of the companies of our database (GVKEY: ID1356 from CRSP databases). As we mentioned in previous sections, the main objectives of a GAwFO is to preserve the diversity of the population and to improve the convergence of the algorithm.

- First we analyse the elite of the population and its fitness value. Figure 5.5 graphically compares the evolution of the fitness of the best individual in both the sGA and the GAwFO. The results in the Y axis represent the in-sample values of the trading system with one the companies used in the first experiment (ID 1356) Black lines correspond to the sGA and grey lines to the GAwFO. Results are for 100 generations with a populations size of 10,000 individuals (GAwFO-10K and sGA-10K) and 500 individuals (GAwFO-0.5K and sGA-0.5K). Figure shows the success of the GAwFO

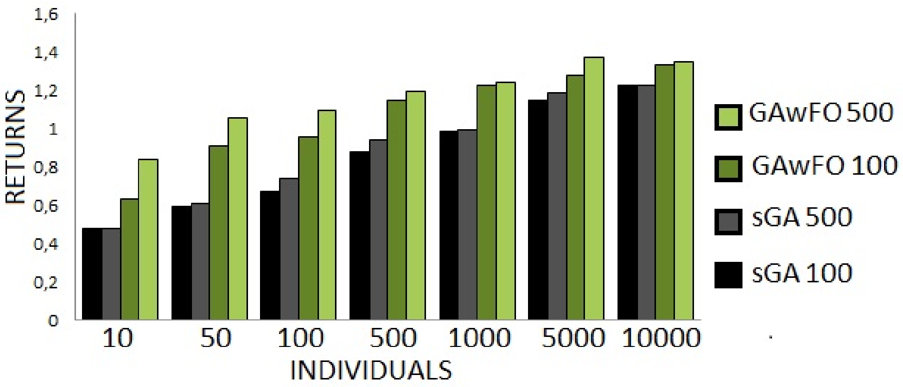


Figure 5.6: In-sample returns (company ID 1356) for the GA and the GAwFO for 100 and 500 generations and several amounts of individuals (X axis)

in preserving the diversity of the population, thus, the algorithm converges to higher fitness values.

- Second, Figure 8.7 presents the results in-sample of the sGA and the GAwFO for100 and 500 generations and an incremental number of individuals (X axis). The Y axis represents the average profit reached by our trading system, i.e. the quality of the best solution found by the algorithm. The experiment shows as GAwFO obtains better results than GA for all the configurations. Here we can also observe better convergence of GAwFO since the final value of the fitness is different for 100 generations and 500 generations, indicating that the algorithm is preserving the diversity along the generations.

Finally, we perform two experiments regarding the execution time of both GAs. The experiments are also performed on the company ID1356 (GVKEY from CRSP databases). As we mentioned in previous sections, one of the characteristic of the GAwFO is the reduction of the execution time.

- First, Figure 8.8 presents, on one hand, the executions of the sGA and the GAwFO for100 and 500 generations and an incremental number of individuals on the X-axis, on the other hand, the in-sample results on the Y-axis. Figure 8.8 shows that sGA needs more time than GAwFO and also getting worse solutions (next figure). This fact is due to the reduction in the number of fitness evaluations. As we mentioned we are reducing to $n/2$ the evaluations on each generation, thus achieving an approximate speed-up of 2 when comparing GA and GAwFO.

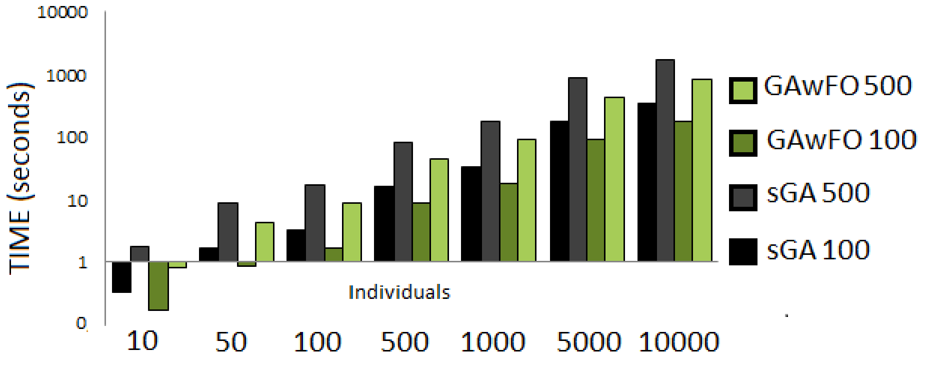


Figure 5.7: Execution time (Y axis) for the GA and the GAwFO for 100 and 500 generations and several amounts of individuals (X axis)

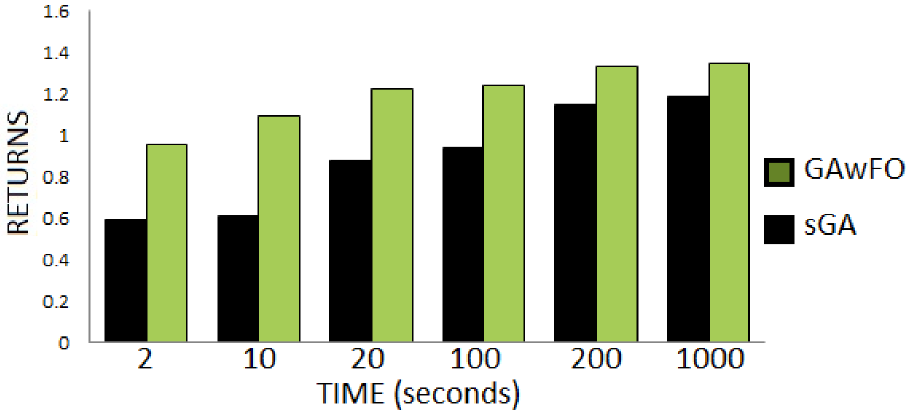


Figure 5.8: Profit (Y axis is the in-sample return) and execution time (the X axis is the delimited time) for the GA and the GAwFO for 500 generations and 500 individuals.

- Second, with the aim to experiment in a more realistic environment where we are limited by the time of the next price movement (seconds, minutes, days, etc.), we measure the fitness values (in-sample results) obtained by the algorithm in a limited execution time. In Figure 5.8 the data on the Y axis represent the in-sample values of the algorithms with one the companies used in the first experiment (ID 1356), and the bars of the X axis represents the used times to delimit the execution in 2, 10, 20, 100, 200 and 1000 seconds.

Although this method has given promising results in short times, we do not have a time windows with enough lengths to run our system with intra-day data. So, the main problem is the execution time. Thus, we are not able to apply our proposal in a sequential execution of the GA to real time problems with intra-day data. In the following sections we explain how we can approach this problem by using several parallel implementations. For this purpose we made first an analysis of the execution time for the whole program and, based on the results, we implemented several important changes not only in the structure of the program, but also in the genetic operators.

As the implementation details of the parallelisation are related to the programming code, we need to specify the software and environments used in the next sections. Each section introduces the parallel architecture used, however we notice that the original language used in the remaining of the chapter is the Matlab language[2].*Matlab* is a mathematical software widely used in universities and research centres. Matlab provides an integrated develop environment (IDE) with a own language of programming (M language). *Matlab* integrates numerical analysis, matrix computation, signal processing and graphics in an easy-to-use environment where problems and solutions are expressed just as they are written mathematically.

5.2 Parallelisation on a Grid Architecture

Nowadays, more and more personal computers are connected to the internet, the power accumulated by linking millions of computers for the same task can be impressing. That will give you a glimpse of the quake taking place in the computer industry these days. Cooperative computing takes advantage of widespread broadband connections and new concepts such as grid, peer-to-peer and the associative string processors to raise Internet to its next level.

Some applications require such high levels of computing power that they require the use of expensive supercomputers. These applications include big science (astronomy and physics), finance and biochemistry. Industry also needs more and more computing power as it shifts from real world experiments to simulation, whether for designing aircraft or for

[2] Section 5.1 presents all the GAwFO experiments with C language and Chapter 6 performs all the experiments with a trading system implemented in Java language. It is worthy of mention that environmental changes have not been motivated by shortcomings in any of the languages, but issues licenses and libraries available.

assessing a cars safety through virtual crash tests. All these applications require intensive computer resources. Many sectors as science, medical research or business are using the grid computing. For example, Stanford University is managing a program aimed at studying genomes and protein synthesis [89].

A high amount of business sectors require enormous computer power, including finance, which we are interested. Buying supercomputers requires heavy investment that can be avoided by setting up computer grids. Take the example of a bank. In order to carry out its complex financial operations, it will be able to use the idle time of computers on its Local Area Network (LAN). This solution has many advantages. First, it is relatively cheap. Second, it is scalable. If the bank needs more computing power, it will only have to tighten its grid by adding more computers to it. Some companies already provide these services, thus these allow hiring his services when another company require it, for instance the enterprise *Grid Systems* (www.gridsystems.com/). We have tested the grid advantages developing an approach of our trading system in the Boinc software. The brand Boinc is the acronym for Berkeley Open Infrastructure for Network Computing and can be found as open software for non commercial use. The software was developed with the main intent of achieving a massive computing capacity. This feature carry out with the interconnection of computers trough Ethernet, either LAN or WAN, like a grid system.

Projects linked to Boinc has a hight exigence in capacity of computing as we can see, for example, in Seti (setiathome.ssl.berkeley.edu/) (Search for ExtraTerrestrial Intelligence), where according to SETI officials boincstats.com/, there are more than 3.5 million of host participating in the program, which have the processing power of a 654.7 teraFLOPS machine in 2014. The total computation power of Boinc reach 7 702 teraFLOPS (boinc.berkeley.edu/). Indeed, it is recognized by the Guinness World Records as the largest computing process in history. Four years after its beginning in 1999 the total cost of the program does not exceed 500000 dollars. By comparison, only the cost to develop the proyect RIKEN from the Advanced Institute for Computational Science (AICS), the most powerful computer in 2011, exceeds the 7.5 million of dollars with a similar computational power (www.top500.org/lists/2013/11/).

Boinc architecture

The tremendous computation power of Boinc falls in the combined computation power of huge sets of computers. Thus, the Boinc framework consists of two layers which operate under the client-server architecture as the Figure5.9 shows. First, the Boinc server is responsible for planning and scheduling tasks for the projects sending works units (tasks) to client computers, which once they have completed their job report the results to the server where are processed and combined.

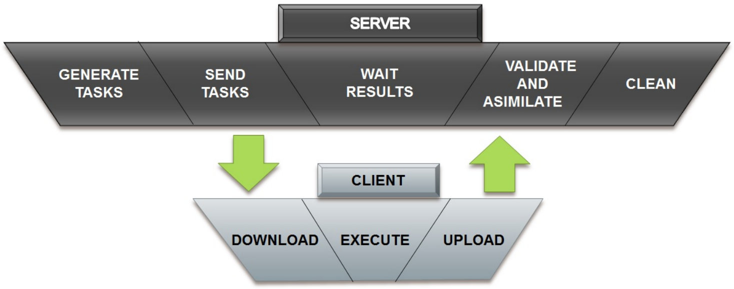


Figure 5.9: The client-server model server work-flow of Boinc

The Boinc platform can be used for private purposes and over computers fully dedicated to a specific project, however, the original idea of Boinc is to provide a platform for anyone interested in supporting scientific projects, the volunteers. Boinc allows the interconnection of any computer using different operating systems (Unix, Windows or Mac) and was conceived for exploiting the lost cycles of a processor unit, that is to say, the spare time where processors remains idle. Thus, Boinc allows to volunteers collaborate easily and without monetary expenses, furthermore there are compensation systems for users as Gridcoin (www.gridcoin.us/).

Parallelization Tasks

With the aim to execute our program in Boinc, we need to transform the code.

- The original code is a unique program that takes as input a matrix with data of companies in the S&P 500 in the period of twenty years. Once the program loads the input data, it executes optimisation engine based on a GA multiple times, one for each company, approximately reaching the 4500 executions. The GA executions are independent among them, therefore we split the program in 4500 sub-modules for executing the program in the grid. Thus, we execute the GA without divide the algorithm itself meaning less complexity in the process of parallelisation. Furthermore, the 4500 sub-modules, which divides the serial program, are enough amount of work-units to test intensively our grid and achieve the best configurations. Each work-unit is large enough to avoid the server overload because a large number of requests, and also they are small enough to not overload the computers with long executions. Thus, the number of changes needed in the code are drastically reduced to change the main loop where the GAs are processed.

- Next, we cope with the management of the input and output data. The original code reads input data from a unique and large file. The size of this file causes an inefficient behaviour on sending data as a unique block. As we did with the programming code, we pre-process the input data dividing it into smaller blocks which are only related with one company, thus, executing the trading system with just the necessary data[3]. Finally, we have around 10000 files; 5000 for technical analysis and the same number for the fundamental analysis. The same process can be extended to the output but in reverse order. Briefly we need to develop two small scripts allowing, to divide the data and, to combine all the results in the same once more.

- Once the input and outputs are fixed, we create a script able to read data, execute the algorithm and save the results. In this order and receiving an unique string with the year and the company ID.

- We build the related executables files (one per operating system) that will be executed in the client computers. The executables files needs the free distribution libraries of Matlab called MCR (Matlab Component Runtime) . Thus we do not limit the proportion of volunteers due to not licensing reasons, or version incompatibilities. MCR libraries are sent to the clients as the first task of the server. Furthermore, besides of the executable, we need to create an additional script (sh) in Linux to enable the libraries of the environment and run our application.

- *Matlab* is not compatible with the Boinc API, therefore the ATS is executed on Boinc by an intermediary called Wrapper. Wrapper is a free distribution software provided

[3] Inputs and outputs changes trigger a domino effect in the source code due to the different dimensions of the data.

Table 5.2: Main characteristics of the Server used in the experiments

Server	IBM xSeries 236 Type 8841
Processor	Intel Xeon 2,8 GHz
RAM	2 Gb
HDD	x4 70Gb - Raid 5

by Boinc. It encapsulates and executes any type of application in the Boinc framework. Thus, Wrapper is a very useful tool when the source code is not reachable, the code does not support changes, or the source code is not developed by languages supported by Boinc, like our system. Boinc provides the source code of Wrapper which we have modified to our own purposes which also is sent as first task.

- The following step is to configure the Boinc framework with the next necessary tasks:
  - We create the structure of our project and applications
  - We change the set up files with our preferences of the server, client and grid policies (project.xml, config.xml and job.xml).
  - We develop a new web interface for promoting and managing the project.
  - We implement a work generator responsible to generate all the tasks.
  - We developed a script to generate templates for all the work-units and results of all executions of our program.

5.2.1 The Experimental Results

In this section we present a description of the experimental tests performing in the Boinc environment.

Metrics

The experimental results are performed in a grid belonging to the CES Felipe II *(A Computer University College of Aranjuez, Madrid, Spain)*, where we used the computers of multiple laboratories and some well-known volunteers (http//:falua.cesfelipesegundo.com).

We use an independent Boinc server for an optimal use of the software. Almost any computer can be used as a Boinc server, however, when the Boinc project uses large number

Table 5.3: Main characteristics of the grid used in the experiments

Group	**PCs**	**CPU**	**Operative System**	GFLOPS	GIPS	**Total** GFLOPS	**Total** GIPS
Lab. ITIS 1	20	2 x Intel P4 3GHz	Windows XP x86	2,744	5,101	**54,88**	**102,02**
Lab. ITIS 2	21	2x Intel P4 3GHz	Windows XP x86	2,744	5,194	**57,624**	**109,074**
Lab. ITIS 3	22	2x Intel P4 3GHz	Windows XP x86	2,744	5,01	**60,368**	**111,22**
Lab. I4	1	2x Intel E2200 2GHz	Ubuntu Linux x86	1,853	5,436	**5,905**	**13,204**
	2	2x Intel P4 3GHz	Ubuntu Linux x86	2,026	3,884		
Lab. DOSI I+D	2	2x AMD Athlon 4600+	Windows XP x86	4,906	8,974	**11,84**	**21,556**
	1	2x Intel P4 3GHz	Ubuntu Linux x86	2,028	3,608		
Volunteers	6	2x Intel P4 3GHz	XP x86 XP x86	2,722	4,747	**35,128**	**77,53**
	2	2x Intel P4 3GHz	Ubuntu Linux x86	1,847	2,876		
	1	4x Intel i5 750 2.7GHz	Windows 7 x64	11,492	36,736		
	1	AMD Athlon 2600+	Windows XP x86	2,129	3,578		
	1	2x Intel T2450 2GHz	Windows XP x86	0,802	1,508		
	1	Pentium III Coppermine	Ubuntu Linux x86	0,679	1,474		
TOTAL						**225,745**	**433,604**

of computers, it is highly recommended to use an independent server. The features of our server are exposed in Table 5.2. The computer is a professional server, specially designed for this purpose and ensures performance, availability and security. Together with the mentioned characteristics of the server we should have an internet connection with adequate performance and a static IP address.

Before starting the experiments of our ATS, we estimate the grid computing power (the capability to support volunteers carries a non constant computing power). The estimation is performed by a set of tests sending to each node of the grid. The test measure computing times in each processing unit. We review our grid with two tests of performance in each computer to determine both magnitudes, gigaFLOPS and GMIPS [4]. Table 5.3 summarizes the main characteristics of the different groups of computers that comprise the system (processor and operating system) and the measures undertaken to estimate the computing capacity of the grid. In short, the grid Falua has about 225 gigaFLOPS and 433 gigaMIPS. These numbers supposed the grid at full capacity, with all the computers active and available.

On one hand the execution times of the experiments related to the grid have been measured by the Boinc itself, On the other hand the execution times executed in one CPU have been measured with the *Matlab* Profiler tool in a standard computer of our grid (Pentium 4, see 5.3).

Other software packages are used for managing the Boinc grid. The manipulation of the computers independently is very uncomfortable. Therefore, we should manage all our network (not the volunteers) in groups and with abilities to operate remotely. The software EMCO Remote Shutdown fits in that propose (`emcosoftware.com/`). We also need a manager for the Boinc client, it is inefficient to configure each Boinc client independently, so we use the Boinc View (supplied by Boinc) which facilitates the management of the Boinc client in the grid.

Execution Time analysis of the Algorithm in the Grid

Figure 8.9 shows the related data with a set of executions where the parameters used to determine the number of individuals iteratively get more weight. These series of executions have been carried out with 500 generations. The number of individuals is represented on the x-axis. The increase intervals in this parameter are not identical along the graphic. The first five intervals raise the amount of 100 individuals each one, and the next 10 intervals raise with 1000 individuals each. On the other hand, the y-axis shows the execution time for the algorithm, and is represented in a 10-base logarithm scale.

[4] The tests are called *Whetstone* and *Dhrystone* and they are provided by Boinc

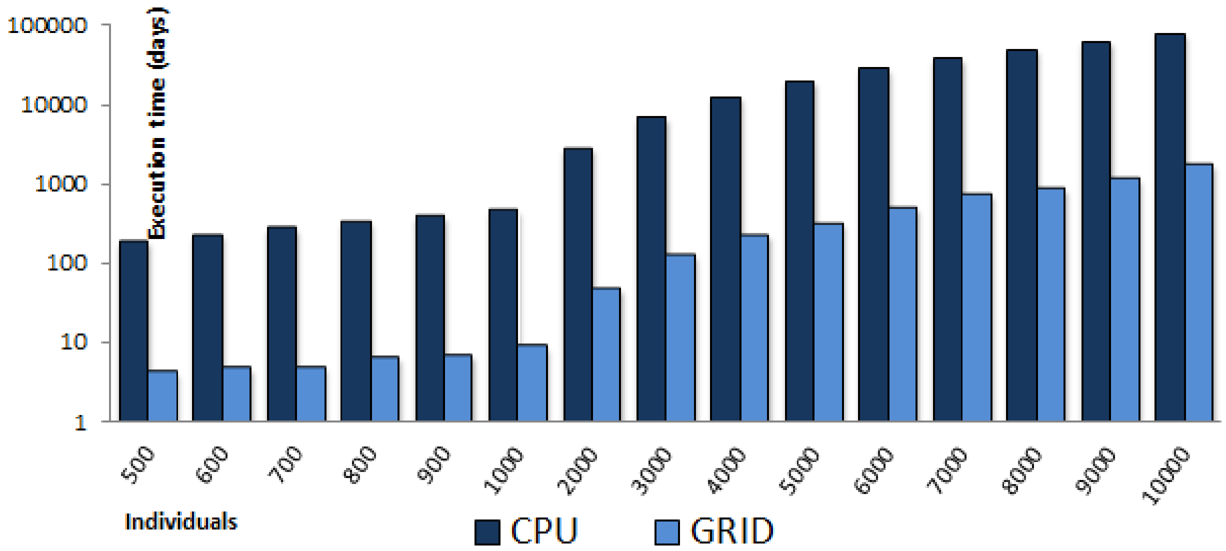


Figure 5.10: Execution times in the grid (y- axis) for 500 generations and different number of individuals (x- axis)

From the beginning, the implemented application in the grid starts to provide beneficial execution times. We can observe the inefficient times reached in an independent CPU. For example, we can analyse the first bars of Figure 8.9, where the times for the CPU with 500 individuals are higher than 100 days (184,9 days), while the grid achieves the execution on approximately one week (4,11 days). Hundred of days of execution is a very high time, fully uneasy, even more if we think that the execution depends only on one machine, with an execution without checkpoints. In this way, the system becomes easily susceptible to any danger or event, like a software crash or a power failure.

Figure 5.11 represents the power of the grid respect to the independent computer version, in other words, we represent the speed-up of the grid. We can observe that numbers are approximately constants. The achieved speed-up is around 50X and this ratio remains throughout all tests realized. The speed-up is not continue at all, this is because the computers in the grid do not have a continued availability, maybe one computer is off by hours or it have a failure or another event out of our control.

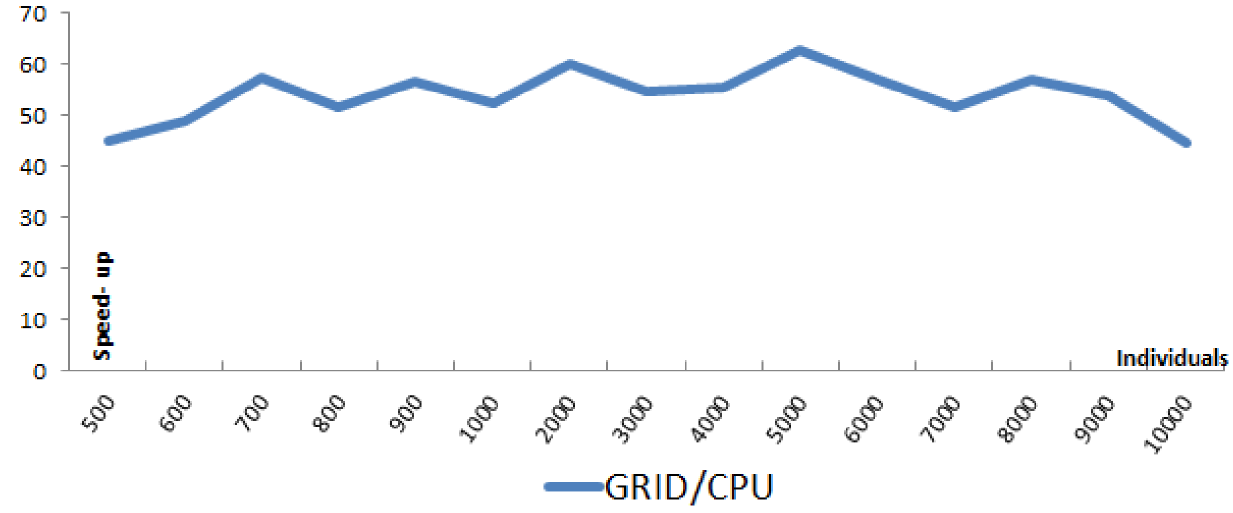


Figure 5.11: Speed-up in the grid

5.3 Parallelization on General-Purpose Computing on Graphics Processing Units

This section introduces a new parallel approach with the development of our ATS with the called General-Purpose Computing on Graphics Processing Units (GPGPU). Section 5.2 was focused in the parallelisation of an ATS analysing all the companies of our data base related the S&P500. Thus, we parallelised the ATS dividing the program into independent modules per company. This chapter implements a GPGPU approach focused on a parallelisation of each one of this independent modules. So far we have used a grid system to obtain a hight advantage in computation times, but we had not been able to raise the number of individuals.

We can find in the literature several approximations for implementing evolutionary algorithms on Graphics Processing Units (GPUs, for example [74]; [62]; [88]; [8]; [82]; [116] among others. Most of this works rely on the CUDA architecture (`www.nvidia.com/cuda.html`) and provide detailed information of how to configure the parameters to obtain an efficient implementation. These works show that doing an ad-hoc implementation require a good level of knowledge on a set of computer architecture and programming issues. However, our proposal tries to offer an adaptable tool for investor with no special knowledge on computer architecture, although familiar with the *Matlab* tools. Thus,

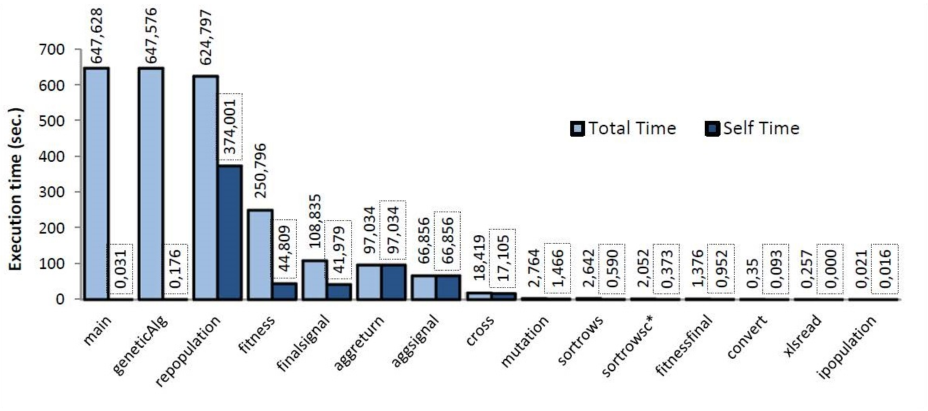


Figure 5.12: Execution time analysis of the genetic algorithm in the CPU. (One company, 5000 individuals, 500 generations). Total Time = 648 seconds.

we propose an implementation based on a software tool called *Jacket* by AceelerEyes (www.accelereyes.com/). In this section we present the general architecture of a graphics device, the motivations to select it, and several parallelisation and implementation details.

5.3.1 Execution Time Analysis in CPU

The ATS executed for one company is based on a GA which already was showed in Section 5.2. The main structure of the GA is a loop with a fixed number of generations. The direct parallelisation of this loop in different execution threads is infeasible, since this would prevent the populations evolution and the concept of genetic algorithm would lose its meaning. In fact, there are other methodologies that originally emerged from approaches that parallelise this loop, as such the island models ([16]). In this we focus in the parallelisation of the GA's basic operators: selection, evaluation, crossover, etc. We analyse the functions forming the trading system in order to determine the critical elements . We use the *Matlab Profiler* to measure the execution times, which shows execution times of each function of the source code. Therefore, we can analyse the time measurements, and propose the parallelisation of the most time-consuming functions.

Figure 5.12 shows the 15 functions with greater weight in the GA computation time. Their names are listed in the base of each column. For example, <main> is the main program and <geneticAlg> is the main genetic loop. The functions marked with an asterisk are MEX functions which have been written in C language (These functions are commonly used to manage external libraries, in this case supports the <sortrows> function). *Total Time* , as the name notes, is the overall time consumed by the program, as can be expected <main> is operative during the whole program execution. *Self time* is the total time minus the shared time that the function itself spends in other functions calls. For example, the <repopulation> function is the part of the GA code which evaluates the population and which selects a individuals for a further crossover. Therefore, the figure shows as <repopulation> spends some time on other functions as the <fitness> (population assessment) and <repopulation> (the selection algorithm). Thus, thanks to the knowledge of the structure of the program and the measurements of 5.12, we observe that most of the time is devoted to run the selection algorithm which takes more than half of the overall time of our ATS. Following to the selection algorithm, the most time consuming functions are the fitness function (<finalsignal>, <aggreturn> and <aggsignal>), and the crossover <cross> and mutation operator <mutation>.

GPGPU Architecture

Multi-threading in processors of general purpose is a common technique which is used for taking full advantage of the available resources. The processor in collaboration with the operating system processes instructions of two or more threads simultaneously. The idea of parallelism and multi-threading is essential in the design of current GPUs. GPU perform highly parallel tasks which process each vertex or fragment graphics, which entails constantly repeat the same operations with different data. Thus, instead the CPU which devote lot of transistors for cache memory and for control flow, GPU uses most of the chip area for Arithmetic Logic Units (ALUS) processing the memory accesses and thread changes much faster than the CPU.

Despite the high computation power the GPUs have boundaries related to the memory bandwidth because they consume too much on the exchange of information. Programming languages for GPU used to be a low level languages and they used an API to transform the data to images, or to transform any type of algorithm in a image processing methodologies. CUDA was born as a general purpose language and parallel software development environment. CUDA represented a new step in the development of applications GPU-based allowing to programmers to use a version of *C* instead a graphics API. Despite the facilities that CUDA provides to programmers, CUDA is a very complex environment for people who rarely developed software. Therefore, we propose to use a easier tool based on CUDA, the *Jacket* software.*Jacket* is a numerical calculation software solution developed by *AccelerEyes*. The choice of this software is motivated by several characteristics of the tool listed below:

- The main characteristic of this tool lies in its ability to accelerate codes by a GPU based on *Matlab*.
- *Jacket* offers the possibility of manipulating matrices easily in the GPU.
- The implementation includes interfaces with other popular programming languages such as C, C++, and CUDA.
- *Jacket* provides a set of GPU versions of most of *Matlab* functions, which facilitates programming if compared to any programming language for CPU. A good example is the generation of random numbers, while in most of the GPU languages are not immediate and can generate problems ([63]), in *Jacket* it is as simple as the use of a single command. Random numbers are a key factor in the correct functioning of the evolutionary process.
- *Jacket* introduces new specific data types from GPU in *Matlab*. Once a GPU data structure has been created, any operation in which this matrix is used will be implemented in the GPU instead of the CPU. In order to stop the GPU computation, we simply must pass the data to the CPU using one of the data types from *Matlab* (*double*, for instance). *Jacket* includes a graphic library, which completes its main characteristic, because with this library computing visualizations in the CPU can be generated. Its use is based on commands, that are versions of the visualization common commands of *Matlab*, as for example *gplot* o *gsurf*.

Thus, *Jacket* gives us the enough capability to compute *Matlab* program with no specific knowledge of the structure of the GPU, which is highly desirable in order to create a useful investment tool since the final user of the methodology probably will not be a computer science expert. Figure 5.13 shows how this software abstract the developer from the complex process of building specific code for the GPU (PTX code).

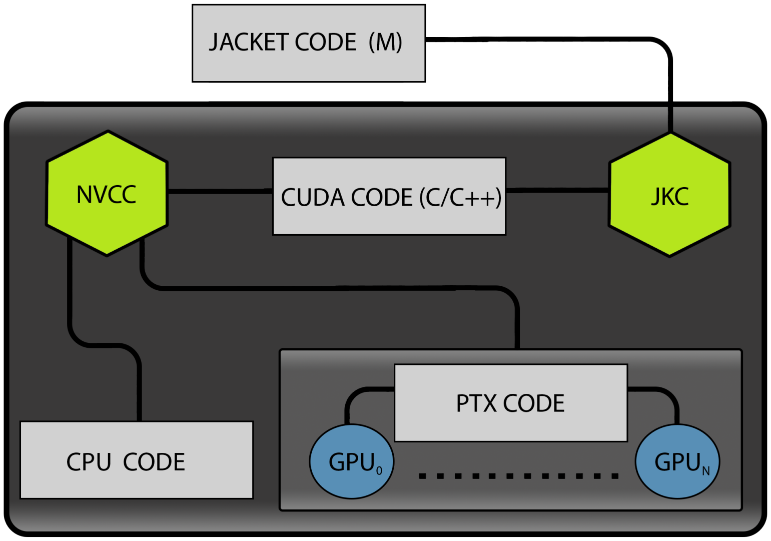


Figure 5.13: General Jacket architecture

5.3.2 Parallelization Tasks

Any program designed for its execution on a CPU should be modified in order to be able to carry out its execution by a GPU. This is due to some limitations existing in the GPU programming languages and to architectural differences between both processing units. The code transformation for its execution in a GPU usually requires drastic changes, and it is here where *Jacket* shows its strong points. *Jacket* is conceived to gain efficiency in the programming execution without having to resort to large modifications. Still, while keeping the basic structure of the code, it is necessary to apply a series of changes to the original code. The magnitude of these changes will depend on the kind of parallelism we want to apply and on the code itself. Reviewed below are the changes considered more important and useful for the readers.

- We pre-locate the data on the memory of the graphic card. In our case, we have two ways of pre-locating the data.
    - When variables are not used outside parallelised sections, we create easily the data directly on memory of the GPU with functions such as *grand(params)* or *gzeros(params)*, which are equivalents to their counterparts for CPU. We want to

note that specially the grand function simplifies and solves one of the difficulties on GPU programming, the random numbers.

  - When we needs to transfer data read by the CPU to the GPU or vice-versa we perform a casting of the data.

- The remaining changes are linked to the incompatibilities that *Jacket* present among CPU and GPU code. The most significant changes are located in those parts of the code that are included inside a *gfor* loop. Remember that a *gfor* loop is identical to a *for* loop in its general functionality, however the first causes that the iterations of the loop are executed in parallel in the GPU.

  - The *grand* command is not compatible with the generation of random numbers, therefore the source code have assumed slight variations in the structure.
  - Accumulators will not be able to be used inside this type of loops, because are incompatible with this command.
  - Branches are not allowed, therefore it could not be possible to use the *if* command inside this loops. Next, we show an example which works as general and easy method to avoid the issue.

    Let us suppose the following code fragment:

$$
\begin{aligned}
&if(x > y)\\
&\qquad A = B;\\
&else\\
&\qquad A = C;
\end{aligned}
$$

    It can be rewritten as:

$$
\begin{aligned}
&Condition = x > y;\\
&A = condition * B + \ condition * C;
\end{aligned}
$$

  - Other functions or commands such as *break*, *return* or the “:”must be replaced by their equivalent codes.

- Renaming the iterators also has been necessary because *Jacket* reserve some names as i and j iterators for complex numbers.

5.3.3 Experimental Results

Next, we present a description of the metric used for experimental tests performing by GPGPU.

Metrics

The experimental results presented in this section are based upon a series of tests executed in both CPU and GPU. These tests consist of a series of computing time measurements in both processing units. Time has been measured using *Matlab* software. Due to the stochastic nature of GAs, all experimental tests have been executed 30 times. Once obtained the required data, a graphic is presented to show and interpret the data in a simple way. The data used for graphs were obtained by the arithmetic average of all previous tests. A speed-up graphic is also included to evaluate the improvement of the execution time in the GPU.

Three Different CPU architectures (see table 5.4) have been used to compare the execution time with the GPU. These CPUs have been chosen due to their great variety of characteristics, for instance the P4 is the oldest CPU and has only capacity to execute one thread, whereas the i7-860 processor is a modern one with a capacity to execute up to 8 threads simultaneously. The SU4100 CPU is an intermediate architecture with a capacity to process two different threads simultaneously.

Jacket GPU programming is only compatible with *nVidia* graphic cards with CUDA technology. For the tests conducted here the GPU 460GTX and the 570GTX has been used. These are modern hardwares, a range normally used for entertainment and with prices of 300 and 150 euros respectively. These graphic cards have been assembled in the third computer of the previous table, that is, the i7-860 CPU computer. Table 5.5 summarizes the main features of the GPU. The data previously presented on Figure 5.12 has been obtained for the architecture corresponding to the third column in Table 5.4. Figures 5.14 and 5.17 have been executed only with the GPU 570GTX.

Table 5.4: CPU Architectures used for comparison

Processor	Intel Pentium 4	Intel Pentium SU4100	Intel Core i7-860
Number of cores	1	2	4
Number of threads	1	2	8
Max. Frequency	2.8 GHz	1.3 GHz	3.46 GHz
Cache	512 KB L2 Cache	2 MB L2 Cache	8 MB Intel Smart Cache
System Bus	533 MHz	800 MHz	2.5 GT/s
Operating system	Windows XP-32-bit	Windows 7 64-bit	Windows 7 64-bit
RAM -Memory	768 MB	4GB	8GB

Table 5.5: Main characteristics of the GPUs used in the experiments

Graphic Card	MSI nVidia 460 GTX OC	Gigabyte nVidia 570 GTX
CUDA Cores	336	480
Memory	768 MB	1280 MB
Clock for graphics	725 MHz	732 MHz
Clock for processor	1350 MHz	1464 MHz

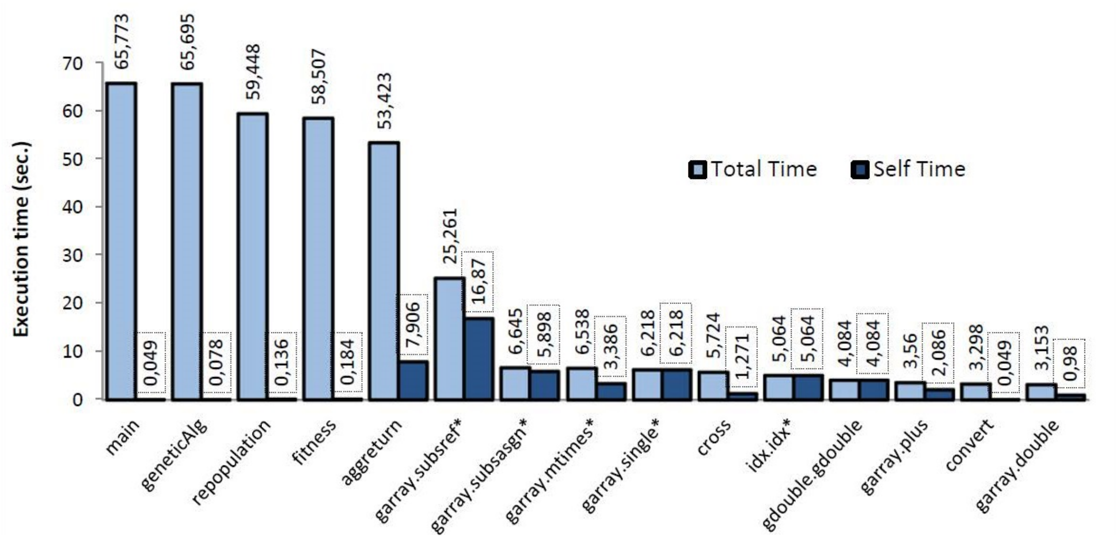


Figure 5.14: Genetic algorithm execution in the GPU. It has been run for a company with 5000 individuals, 500 generation. Total Time= 66 seconds

Runtime Analysis of the Algorithm in the GPU

Figure 5.14 contains the 15 functions with the highest computation times related to the execution of our ATS on the GPU. The figure shows several changes in the weight of the time-consuming functions. This is due to the *MEX* functions that *Jacket* makes using our code. As mentioned above, those functions are written in C, so they become the best way to create an interface that converts the *Matlab* code *(.m)*, to CUDA, since these functions directly interact with the GPU.

By implementing the code in the GPU the searched objectives have been achieved, the selection function no longer is a bottleneck in the program. On Figure 5.14, we cannot appreciate the time reduction in the selection algorithm, this is due to the fact that the *Jacket* parallelisation has provoked the creation of new MEX functions that will be called from the selection algorithm. For instance *array subref* function is the one that consumes the greater part of the execution time. The program total execution time has been reduced, in this particular case, around 90% if compared to the CPU version.

Analysis of the Evolution of the Runtime

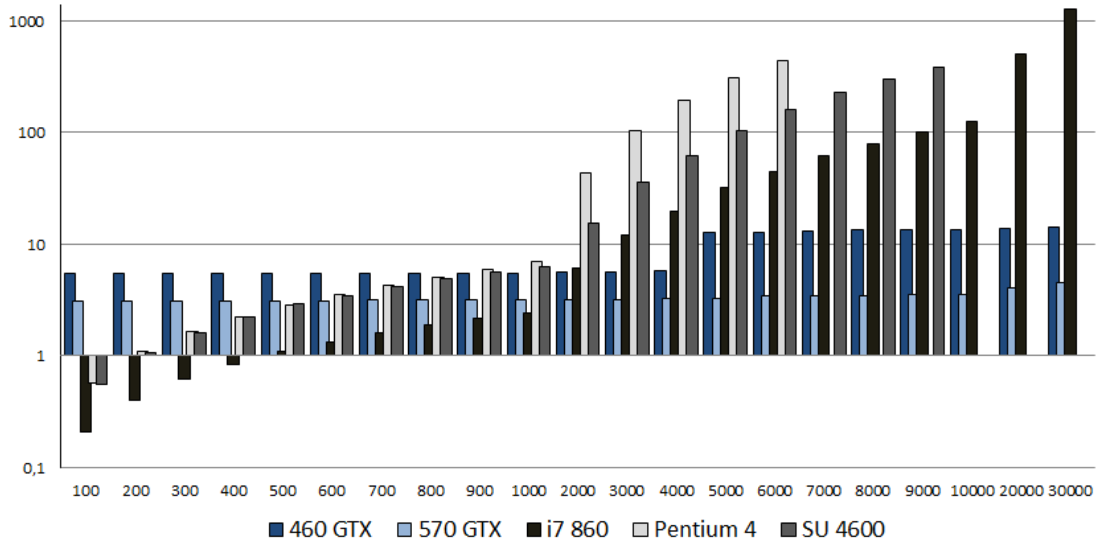


Figure 5.15: Execution times (y- axis) for 500 generations and different number of individuals (x- axis)

Figure 8.10 shows the time results obtained by a series of executions where the parameter used to set up the individuals iteratively increases (x-axis). These intervals raise in an exponential scale each 9 executions. The y-axis shows the runtimes of the ATS, and it is presented in a 10-base logarithm scale. The different colours show the process units which execute the trading system, a total of three CPUs and two GPUs. The Pentium 4, SU 4600 and i7 860 are no able to finish their execution at 7000, 10000 and 40000 individuals respectively. The runtime is intrinsically linked to the number of individuals and the number of generations of a particular execution. Therefore, the analysis of Figure 8.10 shows the turning point where the GPUs start to get better results than the evaluated CPUs. The figure shows than the GPU is slower than the CPU in all the first iterations mainly because the improvements provided by the parallelisation are not enough to exceed the speed at which the CPU executes the data. As the number of individuals increases, the gaps of time between the executions of GPUs and CPUs are reduced. Finally, the turning points where the GPUs improve the CPU computation power are reached:

- CPU Pentium 4: Next to the 500 individuals for the 570 GTX and 900 individuals for the 460GTX.
- CPU SU 4100: around the 600 individuals for the 570 GTX and 900 individuals for the 460GTX.
- CPU i7 860: Around 1500 individuals for the 570 GTX and 2000 individuals for the 460 GTX.

The population size of the GA reaches enough proportion for the execution of this particular problem. In fact with such a number of individuals to reach the convergence becomes an inconvenient. Actually, in our GAs, a large number of individuals cannot be a good choice in order to optimize this kind of problems. Section 5.3.4 shows how to take advantage of populations with high number of individuals without endangering the problem of convergence.

Figure 5.16 represents the speed-ups of the 570 GTX over the CPUs in our ATS. The plot is based on the data results of Figure 8.10, therefore the parameters used are identically. The y-axis represents the speed-up for a specific number of individuals. We achieve the next maximum speed-up values:

- CPU Pentium 4: A speed-up of 128x with 6000 individuals.

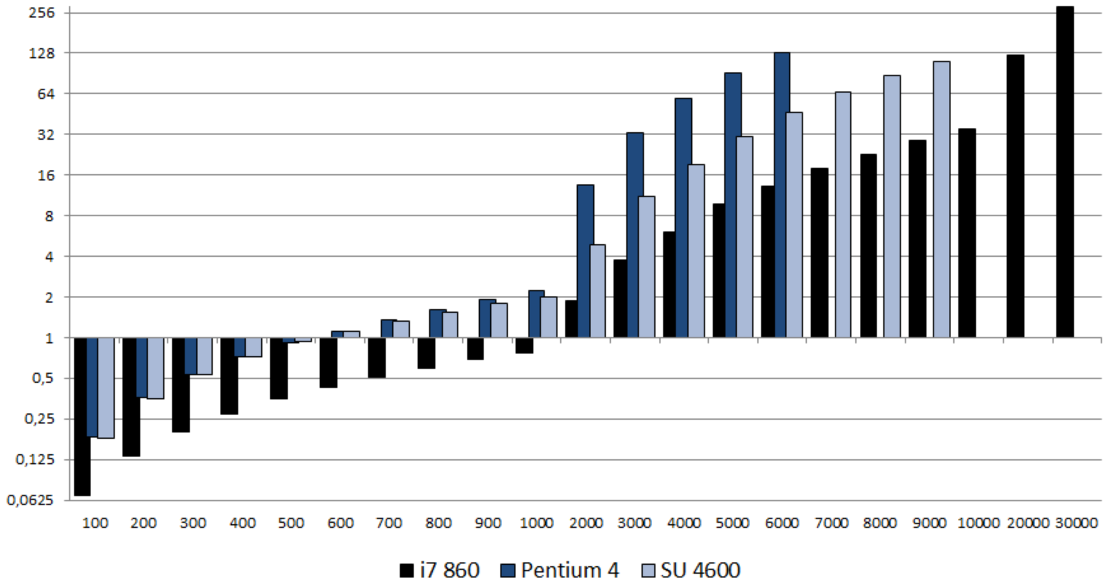


Figure 5.16: Speed-up of the 570 GTX over different CPUs (y-axis) with variable number of individuals (x-axis) and 500 generations

- CPU SU 4100: A speed-up around 100x with 9000 individuals.
- CPU i7 860: A speed-up close to 256x with 30000 individuals

5.3.4 Taking Advantage of the Parallelisation for Trading the Stock Market

In previous sections we showed that our implementation can manage large amount of individuals in acceptable execution times. However, the reader could have some concerns about the advantages of executing the GA with such a high number of individuals for our problem. If the number of generations has not increased in proportion to the number of the individuals, we could lose the algorithm convergence and, even increasing the number of generations, could be that the algorithm converged with lesser number of individuals and generations, therefore missing valuable execution time, especially on real time operations.

To take advantage of generating a high number of individuals, we changed the basic structure of the algorithm and a divided population was implemented. As in the island model ([16]), this model maintains several independent populations, however, in this case there is no collaboration between populations which are independent over the complete

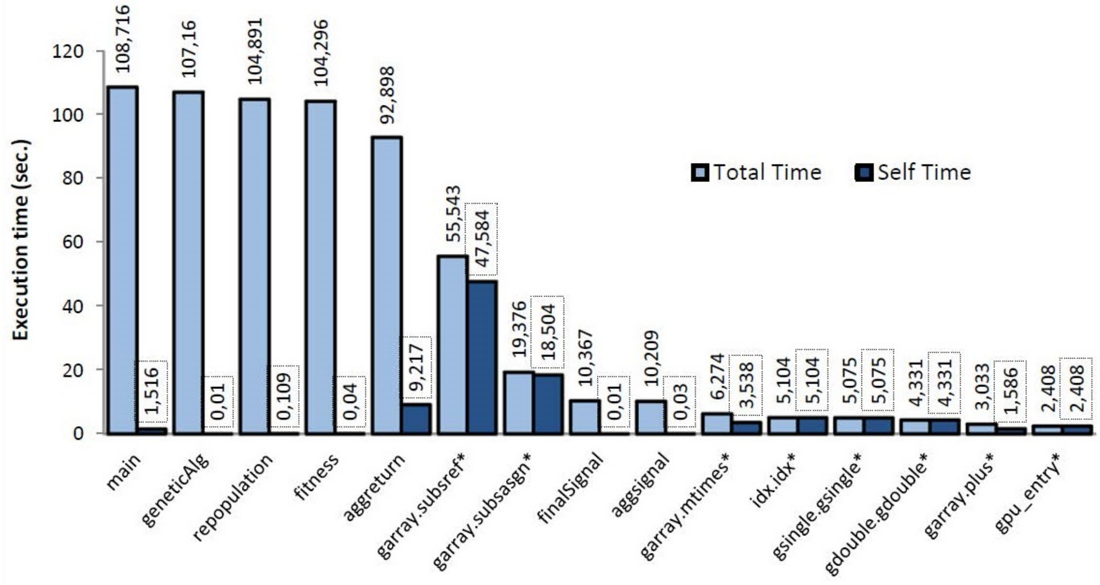


Figure 5.17: Execution of the GA in the GPU. It is run for 10 companies, with 50000 individuals, 500 generations and tournament selection algorithm. Total time =109 seconds, partial time for company =10.9 seconds

execution of the program. This methodology can be considered a new parallelisation of the already done previous parallelisation. For example, to test our data base of the S&P500, the GA must be executed for about 250 companies per year, in a total of twenty years. Finally, we need 5000 executions of the same GA with different data each time. To benefit from this characteristic, all the data from the companies in a given year are loaded and the population of the problem is divided by this number of companies. For example if we have 250 companies and we executed the GA with 125000 individuals (i.e a total population size of 125000 individuals) in which each company keeps 500 individuals.

It has been implemented in such way that all the basic processes of the GA as the selection or the crossover, is independent for each sub population. This way the consistency of the population is kept, allowing them to evolve independently. To avoid a population disproportionate to the number of companies the number of individuals fits automatically to the nearest multiple of the number of companies, so if there are 5000 individuals and 21 companies, the number of individuals the algorithm will implement will be the result of truncating (5000/21) and multiplying again by 21, total 4998 individuals.

Through this technique the number of executions can be simplified a lot, in the above mentioned example, the 5000 executions will be reduced to 20. Playing with the number of individuals and the number of companies that can be executed at the same time substantial advantage is achieved benefiting from the GPU capacity. Hence, that parallelisation allows us to implement and test previous GA approximations.

Figure **??** shows the first 15 most expensive computational functions of a series of executions related with the above described methodology. This test has been made simultaneously for the data of 10 companies. To be able to compare with the previous execution time analysis, the same execution parameters have been used but rising the number of individuals up to 50000. In this way, each company is linked to 5000 individuals remaining the proper proportion for their direct comparison. The execution time for company is given by the total execution time divided by the number of companies, which makes a total of 10.9 seconds. The execution time for company is reduced approximately the sixth of the total, if compared to the parallelised version in the GPU presented on Section 5.3.3.

5.4 Grid Computing Versus GPU Computing

We have used two different platforms for computing the same program and thus analyzed the benefits of the paralysed architectures. Now, we compare the two parallelisation technologies, *Jacket* and Boinc, the GPU and the grid.

First, we will evaluate the complexity of developing a project in these platforms. The Boinc platform is a very powerful system, however, we need a deep knowledge on IT to take advantage of Boinc. Since the installation of the Boinc server until the execution of the project, the developer needs amounts of software configurations and learns the monitoring of specific procedures. Therefore, the Boinc platform requires a larger knowledge than *Jacket*. Set up an application for Boinc request a middle level of C, SQL, Bash, etc. Nevertheless, you only need to know a middle Matlab language to develop an application in *Jacket*. Thus, the learning curve in *Jacket* is softer than in the Boinc system. Nevertheless, we should develop for Boinc if we already have developed an application not implemented in Matlab and we want to improve the performance in the execution times. Boinc allows the parallelisation in several languages, furthermore includes tools like Wrapper to encapsulate the not compatibles applications.

We can affirm that the Boinc platform is by far more stable and reliable. For example, the execution in this system can be stopped at any time and back to the execution whenever we want. The GPU cannot stop the execution. Boinc can send several works of the same company for comparing results, we can see the times of each execution or the percent of program executed. In short, Boinc is a mature platform which provides countless options, configurations and facilities.

5.4.1 Performance GPU vs Grid

With the aim to compare the performance of our trading system in both platforms, we present bellow the execution times in several graphics.

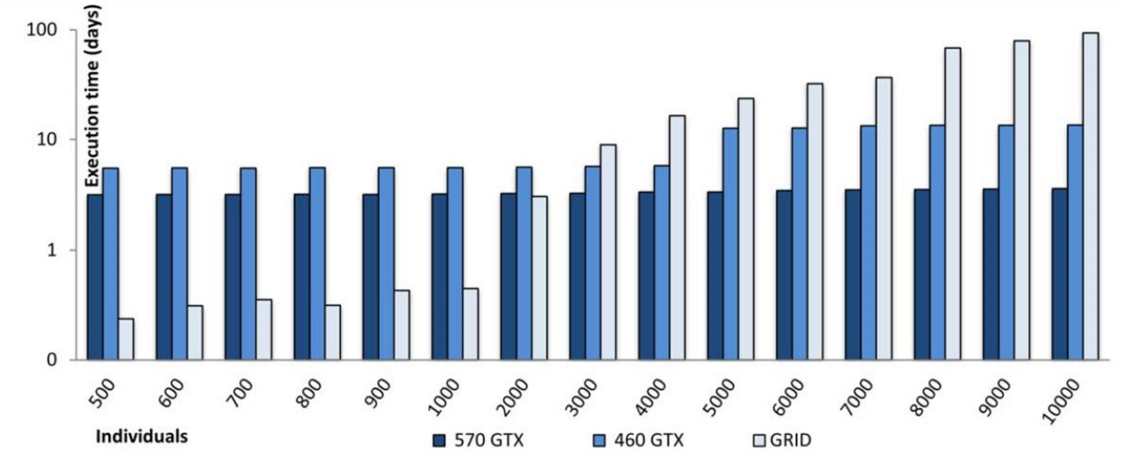


Figure 5.18: Execution times comparative (y- axis) for 500 generations and different number of individuals (x- axis)

Figure ?? shows a comparison chart between the different execution times of the platforms analyzed in the present document. The y-axis represents the execution time measure in days. The x-axis shows the different amount of individuals in each execution. The most important point of this graphic is to check the useful period of the executions; in other words, we can observe that the grid system becomes inefficient around the 4000 individuals, because we consider that more than 10 days of execution starts to be too large times. Nevertheless, the GPU system remains with a little increments throughout the test (all the time bellow the 10 days).

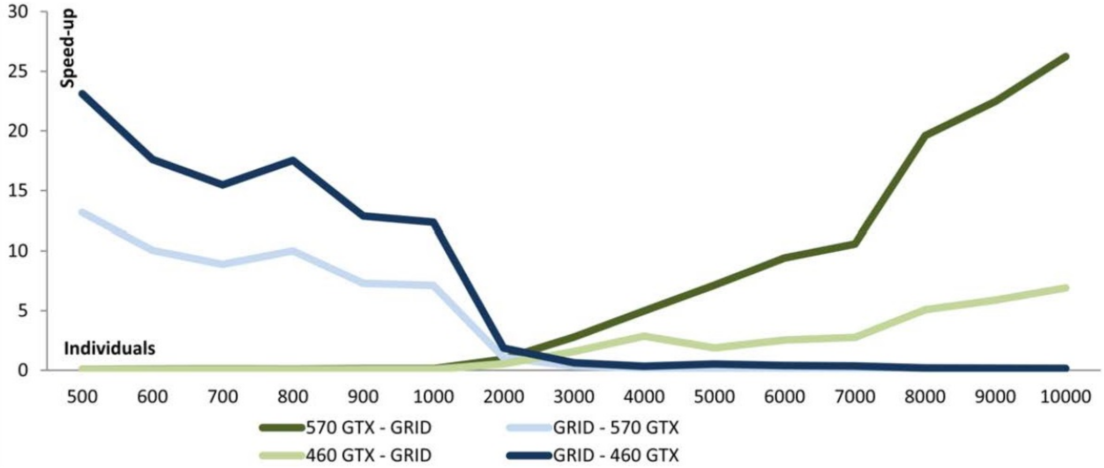


Figure 5.19: Speed-up on the different platforms (y-axis)with variable number of individuals (x-axis) and 500 generations

With the same tests used in Figure ??, we have develop a graphic of speed-up. Figure 5.19 shows the speed-up between the GPU and the grid, and also the reverse form. More specifically, the dark lines represent the relations: 570GTX/grid and grid/570GTX. The clear lines represent the relations: 460GTX/grid and the grid/460GTX. As in the above figure, x-axis represents the amount of individuals and y-axis shows the speed-up.

The first thing that takes our attention is that both platforms could be useful in different periods. At the beginning, the Boinc system has a better behaviour than the GPU, but this performance vanishes when the number of individuals in a population rises. Beyond 2000 individuals the GPU/grid speed-up grows up constantly. Maybe the critical point (where both technologies have the same performance) should be shifted to the right because the GPU uses a different algorithm, however the trend to raise the speed-up by the GPU is clear. After, we have analysed the above figures; we can conclude that the grid system works fine when the number of individuals is not too high.

Chapter Conclusions This Chapter, based on the conclusions of Chapter 4 about developing prediction systems based on the study of time series of historical prices, has introduced the next contributions:

- First, the chapter has presented an ATS based on GAs. We have used a novel version of a GA (GAwFO) to optimise the trading signals over a period. The GAwFO proved several advantages, both avoiding the premature convergence and improving the final results. The financial results achieve high return reducing the risk of losses.

- Second, we presented two implementations combining GAs and parallel architectures. We used a grip to paralelise the trading system using at company level and a GPGPU platforms to paralelise the system at individual level, thus we obtain solutions for real time (or intra-day data) investment decision. We presented high speed-ups in the two selected platforms achieving values up to 50x in the grid and almost 256x in the GPGPU.

Chapter 6

Hybrid Automated Trading System based on Grammatical Evolution

This chapter presents the final contributions of the thesis where we develop a new and improved version of our trading system implementing a new level of analysis. With this aim, we carry out two main steps. On one hand, we analyse new features of the methodology testing different ATS approaches with new data. On the other hand the chapter shows implementation details and experimental results of the new ATS analysing macroeconomics, fundamentals and techinicals.

Building on the already GA-based ATS tested in the previous chapter, and with the aim to offer more realistic results and put the trading system through its paces, Section 6.1 starts updating the data set with recent values. The same section introduces grammatical evolution to our trading system in order to provide greater flexibility to our solutions. Furthermore, the section explains the new work-flows and shows implementation details of multiple versions based in different the fitness functions including a multi-objective version.

Companies listed in the stock market behave in multiple ways and the operative of our ATS allowed us to invest in companies selected by the professional investors. Each company provides different returns and risks, however they are linked by the industrial sector where they develop its activities (as we demonstrate in Chapter 4). We propose and analyse the valuation of our trading system through various business activities. Thus, Section 6.2 presents experimental results related to:

- compare the two EAs implemented: Genetic algorithms and Grammatical Evolution.

- evaluate results of the new ATS trough three different fitness functions.
- test a multi-objective version of the system.
- analyse and evaluate returns by industrial activities

Finally this chapter presents the most complete trading system developed in this thesis in section 6.3. The new approach proposes a novel methodology to analyse and invest in multiples exchanges at the same time. The ATS performs an analysis of almost one thousand companies scattered over Europe. The methodology of the system uses fundamentals, technicals and macroeconomics in a hybrid analysis to locate the more attractive companies and trigger investment signals.

6.1 Trading the Stock Market: Automated Trading Systems Based on Grammatical Evolution

This section sets the initial stage for the development of our complete system in Section 6.3 where we perform an exhaustive analysis of the more profitable companies in major world regions.

New dataset: Europe in recession

The trading system is used to invest in the market through the analysis of historical data. The selection of historical data is an important decision in the development of the system, the quality of the results are biased by the stability and how profitable are the trends of the market in the analysed period. Thus, it should be easier to achieve better returns in a period of rising market than in a diminishing market. So, the already difficult and exciting challenge of building a trading system able to predict the intricate behaviour of the stock market becomes more difficult and interesting in a hostile environment. Thus, we test our trading system with current data and analyse its performance in one of the most hostile scenarios, the current economy recession. We updated our old dataset that comprised the daily data of the years ranging from 1994 to 2004. Thus, our current data is located between the 2001 and 2013, in the middle of one of the most important economy cracks. We are fixing the period, which answers the question *When*. Nevertheless there is another important parameter to choose the historical data, that is *where*. Our old data is retrieved from the SP&500 which is one of the most important indexes of the stock market. Now, we are considering a group of listed companies in Europe, one of the continents with more financial problems in this economic recession. The companies that form our dataset are listed in Orbis https://orbis.bvdinfo.com/ and have been chosen according to the next criteria:

- Publicly Listed companies in Europe.
- Active companies with at least 5 last years of historical data in the stock market.
- Current market capitalisation greater than 25.000.000 euros.
- Excluding pension fund.
- Excluding financial firms (Financial and insurance activities).
- Excluding Public administration.
- Excluding activities of extraterritorial organism.

The historical series of data have been downloaded by Bloomberg software http://www.bloomberg.com/. We use a daily frequency for the data.

6.1.1 Work-flow and implementation details of the Automated Trading System based on Grammatical Evolution

The GE trading system operate in a similar way than the GA-based system. Next we offer a outline:

1. The investor selects a set of Technical Indicators for TA
2. The investor establish $Threshold_{buy}$ and $Threshold_{sell}$ ranges
3. Define a Grammar in BNF
4. For each company i
    (a) Apply GE over a period of X days to obtain the grammar solution A_i.
    (b) Built the program P_i related to the grammar A_i.
    (c) Run the program P_i over a period of Y days.

(d) Compute the profit given by the Trading System

The GE trading system is managed by the implementation of six main technical indicators. The selected technical indicators are the Moving Average Crossover, Volume-Price, Moving Average Convergence Divergence, Support and Resistances (SR) and Relative Strength Index used in two ways, to spot divergences that show if a trend is fading (RSID), and to identify Overbought/Oversold levels (RSIO). Each one of these technical indicators gives us a signal of buy, sell or neutral. The indicators have been initially selected due to their utility in the professional and academic world of finances and they are widely explained in Chapter 2. However, thanks partly to the flexibility of grammars, we are not limited to six classical indicators. The GE optimizes the parameters related to each indicator, but additionally allows the building of new versions of the initial implemented indicators.

The solution rules built by the grammar provide the accumulated value of the trading signals per each day of investment. As in the GA approach (Chapter 5), the value is compared versus the pre-setted $Threshold_{buy}$ and $Threshold_{sell}$ ranges. The system provides a buy signal if the value exceeds the $Threshold_{buy}$, a sell signal if the value is lower than the $Threshold_{sell}$, a neutral signal if the value is between both thresholds. The thresholds can be increased or decreased to profile the system behaviour. Thus, initially the program becomes more aggressive or conservative when it performs investments. However the GE system ,unlike the GA approach, can increase the number of indicators involved in the solution indefinitely alleviating the liability of the thresholds. In order to the better understanding of this feature, next, we expose an example. Let us to consider the following threshold configurations

- {A} Buy threshold=5 Sell threshold= -4
- {B} Buy threshold=20 Sell threshold= -20

On one hand, the first configuration produces an initially aggressive trading system. The signals produced by the technical indicators easily exceed the thresholds, therefore, they provide a higher number of investments. However, if the fitness function assesses the conservative strategies as the most profitable operations, the solutions eventually will evolve through the generations into conservatives strategies. On the other hand, the second configuration may represent the opposite case. The accumulated value of the signals rarely exceed the thresholds, therefore, they provide a lower number of investments. However, if the fitness function assesses the aggressive strategies as the most profitable operations, the solutions eventually will evolve through the generations into aggressive strategies.

Grammatical Evolution: Seeking the best investments operations

As we have previously explained, GE uses grammars to build a set of rules that guide our trading system. With the aim to expose the main features of our grammar we present in Figure 6.1 a fragment of the grammar used in our automated trading sytem.

```
N= {<code>, <indicatorSet>, <combination>, <indicator>, <weight>,<range>,
<type>, <parameters>, <RSID>, <RSIO>, <SR>, <VOL>, <MAC> }

T= { ");" , "," , "MIX(" , "MAC(", "MACD(" , "SMA" , "WMA" , "EMA" , "HMA"
    ,... }

S= { <code> }

P= {I, II ,III ,IV ,V ,VI, VII ...}

I   <code> ::= <indicatorsSet>
       | <indicatorsSet><combination>

II   <indicatorsSet> ::= <indicatorsSet><indicator><weigh>
       | <indicator><weigh>

III   <combination> ::= "MIX("<range>","<indicadorSet>");"

IV   <indicator> ::= <MACD> | <RSID> | <RSIO> | <SR> | <VOL> | <MAC>

V   <MACD> ::="MACD("<type>","<type>","<type>","<parameters>");"

VI  <MAC> ::= "MAC("<type>","<type>","<parammeters>");"

VII   <type>  ::= "SMA" | "WMA"| "EMA"| "HMA"
...
...
...
```

Figure 6.1: Fragment of the grammar used in our ATS GE-based. The grammar has the aim to build programs providing investment signals on a specific period.

As the grammar 6.1 shows, in addition to the indicators we include a weighting factors. In the grammar we can see how each indicator is associated with a weight *(<indicator>*

<weight>). Their values represent the importance of each indicator, thus the importance of each technical indicator is modified depending on his weight value.

Generating Technical Indicators

With the aim to provide greater flexibility to our solutions, we implement four different moving averages which already were explained in Chapter 2. The objective is to offer a novel level of flexibility in our trading system. There are countless technical indicators and even nowadays new technical indicators are created. Most of them are modifications of other indicators, or combinations thereof. Furthermore, there is not a perfect formula to select and use them. Thus, we implemented several versions of moving averages in order to allow that our trading system creates new indicators. Therefore when we calculate the MAC, MACD, RSIO or RSIOD the system chooses any moving average. Next, we show an example related with the grammar showed.

Let us consider the following solution:

$$MACD(''WMA'',''HMA'',''EMA'',''9'',''18'',''11'');$$

Where 9, 18 and 11 are the parameters (<parameters>) optimised by the GE. And where WMA, HMA and EMA are the MA types (<type><type><type>) optimised by the GE. We calculate the MACD in the next two steps:

- First: The MACD line is the difference between two MAs: WMA(9) and HMA(18)
- Second: the signal line which is an MA of the MACD line: EMA(11)

Thus, the ATS creates a new indicator based in the old MACD but working together with the HMA, which is a novel and effective MA. Therefore, this contribution is an innovative feature easily expanded which allows to ATSs not be constrained to a default set of indicators building its own technical indicators.

The advantage of this is that EDDIE is not constrained to use prespecified indicators; instead, thanks to its grammar, it can choose any indicators within a pre-defined range, leading to new solutions that might have never been discovered before. However, a disadvantage of the above approach is that the algorithm's search space is dramatically larger, and as a result good solutions can sometimes be missed due to ineffective search.

Combining Indicators Over a Period

Following the line of the previous feature, we implemented the possibility of applying different set of indicators in the same period. The implementation allows the grammar to select branching strategies, i.e, it allows to choose different strategies depending on the behaviour of the performed investments. Thus, when a strategy is not working properly in a specific period, the ATS changes its strategy. Therefore, the trading system could be fit to sudden changes in the behaviour. Next, we study an instance of a solution with this feature.

Let A be the solution obtained when we apply our trading system in a time period **P**:

$$MACD(type1, type2, type3, \{parameters1\});$$
$$MIX("56", "MAC(type5, type6, \{parameters2\})");$$

Where $type_n$ and $parameters_k$ are the types and the parameters optimised by the GE algorithm. And where 56 is a sub-period (**SP**) (<range>) of the period analysed for our ATS.

Then, the return of the day_i according to the solution A is calculated with the next algorithm:

1. Calculate the return X of the system in the sub-period SP using A=MACD.
2. Calculate the return Y of the system in the sub-period SP using B=MAC.
3. If (X>Y)
   (a) Calculate the return of the day_i using A
4. Else
   (a) Calculate the return of the day_i using B

Thus, we calculated the MACD and the MAC values of the last 52 days, if MAC fits better with this period the system uses it, else the system uses the MACD. Thus the system could react to repetitive patterns in small periods which are not exploited for the general strategy. We note that the work-flow is generic to any solution being A and B any possible grammar combinations of indicators.

Furthermore this feature allows the system to optimise in some way the periods of training. Thus, when the ATS achieves high returns with short periods of training, the system tends to converge to unprofitable strategies in X, and profitable strategies in Y. Hence, the system could be optimised with short periods ceasing to take into account the fixed training period.

6.1.2 Operators and Fitness functions: Automated Trading System Versions

The population of our GE is codified by an integer codification and is evolved using classic operators

- We allow ours individuals to generate the offspring by the crossover operation (probability= 0.85), crucial for the right working of the GE [85].
- We use single point crossover which has already been demonstrated to be a successful crossover operator and capable to achieve good performance in terms of utility in the process of interchanging blocks in the chromosomes.
- The mutation operator is implemented using the well known Integer Flip Mutation (probability= 0.02).
- We use also the distinctive operator of Grammatical Evolution, the wrapping operator.

Finally, we need to focus with more detail on the function which guides our ATS, the fitness function. Through the literature we find some authors that claim the lack of proficiency of the AR function in the task of evaluating investments [86]. The AR function avoid a determinant factor in the evaluation of the investments, the risk. Although the trading system itself uses technical indicators supporting in some way the measure of the risk (e.g the support and resistances), the highly volatile period requires a better risk assessment. Therefore we introduce two new functions in the trading system supporting the risk assessment. The SI and the CECCP, both described in the Chapter 3 in equations 8.8 and 8.9. Therefore, we can test the trading system with each one of the three implemented versions, one per fitness function. The inclusion of the new fitness functions drives us to implement also a multi-objective approach.

A Multi-objective Approach

Trading systems can assess large amount of factors and objectives with the aim to optimise series of investments. Due to the complexity of the problem, different environments may require different factors and objectives. The GE evaluates multitude of possible combinations of these factors. However we are guided by an unique fitness function. The financial works differ in the assessment of the importance related to the objectives. Therefore, we include in our system a new feature able to provide an optimal tradeoff among different objectives. We integrate a multi-objective optimization, a combined approach which could provides solutions fitting better to different behaviours in the sector or companies and therefore to achieve better returns.

Others works have used multi-objective optimization to trade stocks. For example in [13] authors claim high profits with a multi-objective version of a trading system based in the RA and the SI. More recently, Bodas et al. in [?] and [107] show a multi-objective GA where the parameters of several technical indicators are optimised whenever a new data is received.

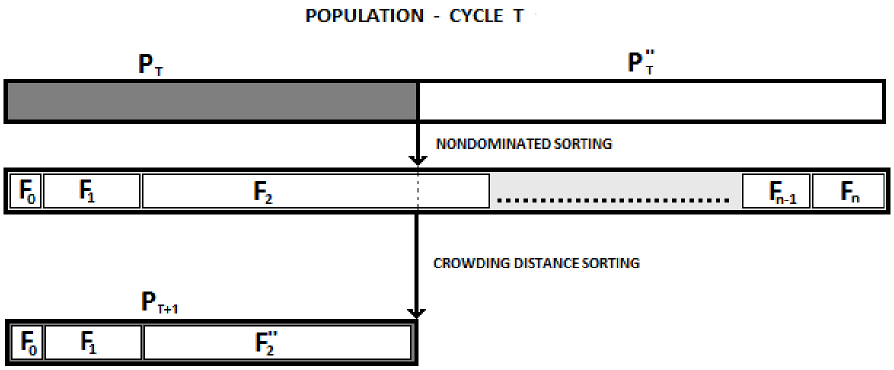


Figure 6.2: Flow of a NSGAII.

We implement a version of a well-known multi-objective algorithm, the updated version of the Non-dominated Sorting Genetic Algorithm [31] (NSGA -II) based in GE. We have chosen NSGA-II since it works with any number of objectives, which can be easily added or removed. Thus, we evaluate and sort by dominance obtaining a set of non-dominated solutions as better individuals. The procedure of NSGA-II is shown in Figure 8.13. First, a combined population of parents P_t and offspring P_t'' is formed. Then, the population is sorted in non-dominated frontiers (F_0 contains best solutions). Next, we transfer individuals from the frontiers to the new P_{t+1} until it be filled. If a full frontier cannot be inserted in P_{t+1} because overflows, we sort according to their crowding distance [31] (in descending order) choosing the necessary number of individuals to fill the new population. The created population P_{t+1} is then used for selection, crossover and mutation (operators work with same implementations and probabilities used in the previous approach) to create a new population. Once the algorithm reaches this point, the process is repeated until the maximum number of generations.

6.2 Experimental Results: Automated Trading Systems Based on Grammatical Evolution

The experimental results presented in this section cover the ATS versions described in the previous section. We noted that the companies are still selected previously, therefore we must diversify our investment in a wide range of companies or rely the decision on a expert opinion.

Facing Automated Trading Systems Based on Genetic Algorithms and Grammatical Evolution

In first experimental results of this this section we analyse the results provided by two trading systems within a recession economy period in one of the most affected countries: Spain. Just as we previously use in Chapter 5, the ATS GE-based implements the AR fitness function 8.7 (Chapter 2). At this point of the work we have already implemented other fitness functions, however we want to collate the methodologies of GA and GE trading systems, so it is more suitable in order to evaluate the differences.

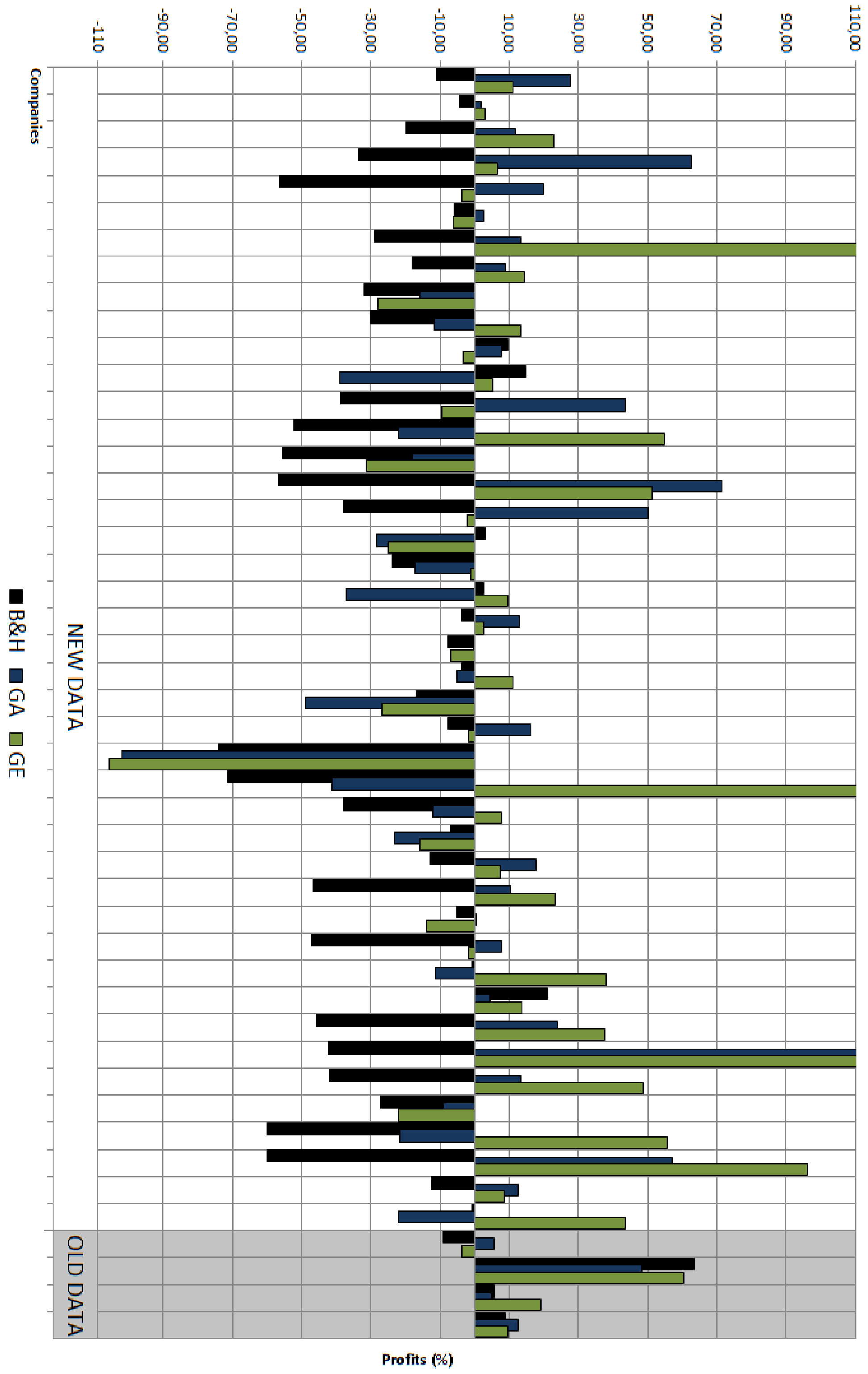


Figure 6.3: Returns of 43 spanish companies over 2012 (recent data) and 4 companies of the S&P500 over 2004 (old data)

The objectives of this section are not only to test the GE trading system with the new data, but also to compare it with the previous GA approach. As we want a fair play when comparing both systems, the set of indicators was adjusted. Therefore, we use the six technical indicators for both systems which already were defined in the last section.

Figure 8.11 shows the results of 43 spanish companies. The vertical edge shows the returns of the companies in the year 2012 after we build the rules and adjust the parameters of our ATS with data over the period 2001-2011. The horizontal axis shows the different strategies used grouped by the companies. Thus, we obtain the average returns of -23.63 % for the Buy and Hold strategy, a 5,89 % for the GA approach and finally a 21,08 % for the GE approach, furthermore we noted that the mean of operations with positive returns are 8, 28 and 29 respectively.

Testing the GE Automated Trading Systems

After facing the GE and GA ATSs over one of the most affected countries for the economy recession we focus our experiments in the GE approach. In the remaining experiments of the section we change our sample of Spanish companies, with the aim to get a more diverse sample, using 36 companies listed in the next countries: Germany, United Kingdom, Spain and France. Thus, in the next experiment we present a extension of our experiments with the new implemented fitness functions.

Figure 8.12 shows the results of a series of investments in 36 companies. The results are correlated to the investments over year 2012, and the process of optimisation was performed during the previous 10 years. The Y axis shows the returns obtained by the trading systems GE-based. The X axis represents the the different fitness functions (legend) and the companies which have contributed to the experiment. Each company comprises 3 bars, every one linked to a fitness function: the AR, the SI and the CECPP (already explained in Chapter 3). The averaged returns provided by the system are 10,94% for SI, 40,79% for the AR and 20,32% for the CECPP. SI, despite obtaining the lowest return, is the strategy with the least number of operations with negative returns with a total of 12 unprofitable investments reaching a total return-26.68% (sum of the losses of all the companies). AR obtains 16 operations with negative returns summing a total of -274% .Finally the CECPP shows the worst behaviour because the negative returns reach the -332% with 18 failed investments and, as we have noted, the averaged return slightly exceeds the 20% .

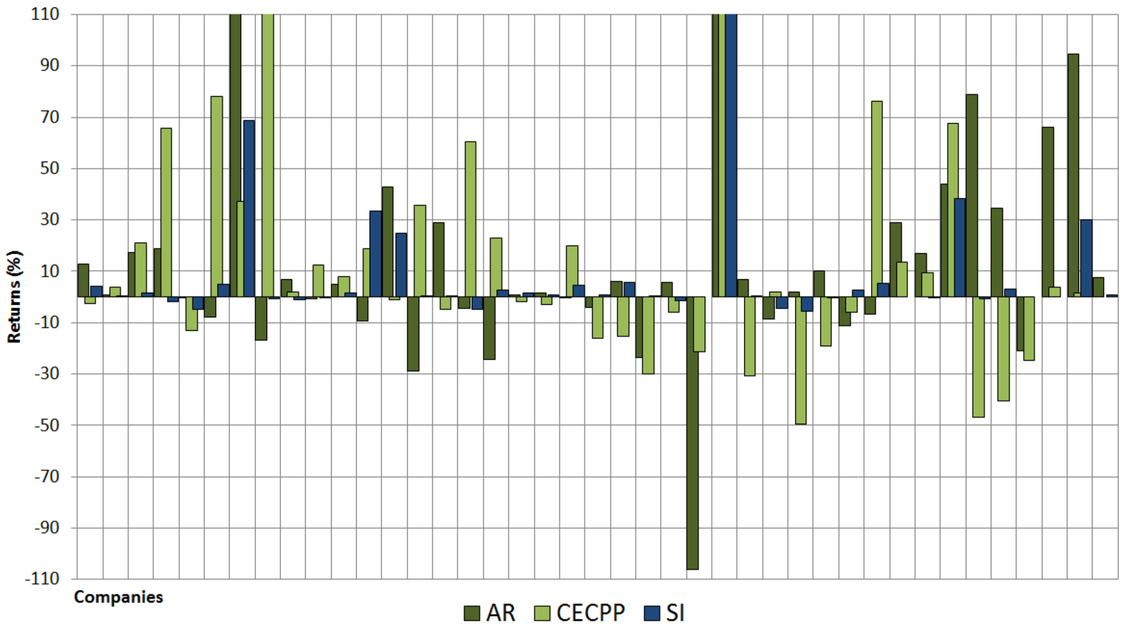


Figure 6.4: Returns in 2012 of the different fitness versions of the ATS based in GE for 36 European companies

The simplest conclusion about last results is that, first, we do not use CECPP in any case. Second, we use SI when the portfolio is composed by few companies or is selected without any criteria, and finally, if we diversify the investments in large set of companies or we have experts knowledge about the portfolio analysed. However, a detailed study of the figure shows that although in general terms SI and RA provide higher confidence, CECPP achieves good results in many cases where the others functions show worse results. Thus, the CECPP provides some features able to get returns where the SI and the RA are unprofitable. This observation encourages us to implement a system which tries to capture the advantages of various fitness functions in order to improve the final returns.

As far as we know, the system based on GE and NSGA II (described and defined in the previous section 8.7.2) is the first system using a multi-objective approach and a GE methodology. Figure 8.14 presents the returns of two multi-objective systems (Y-axis) over the 36 European companies used in the last experiment (X-axis). On one hand we use a

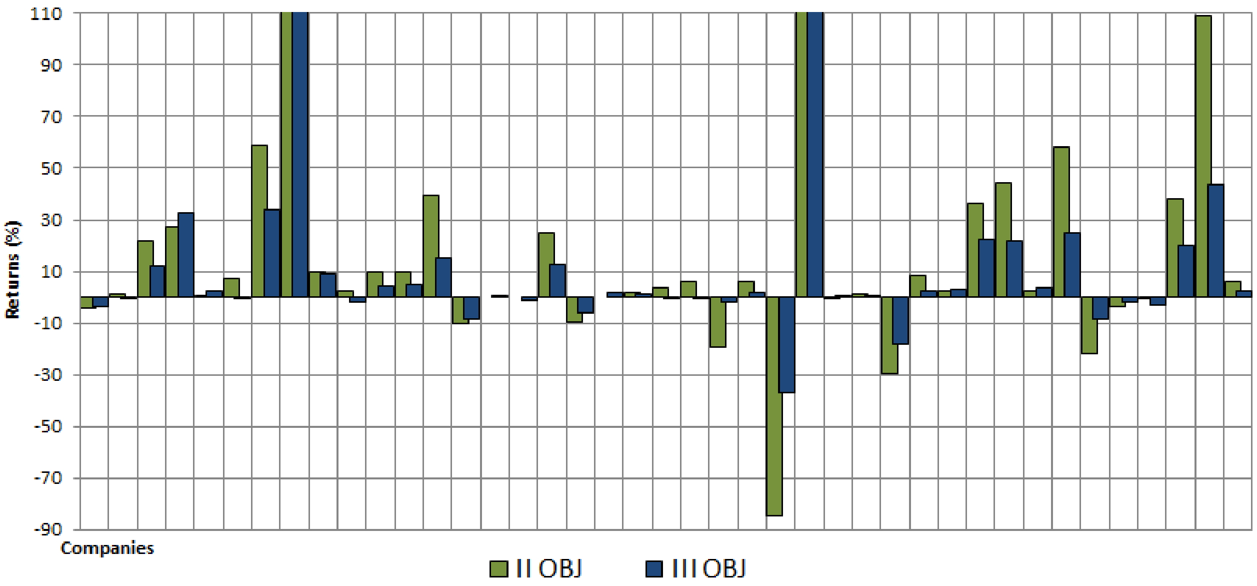


Figure 6.5: Returns of the multi-objective versions of the ATS based in GE for 36 European companies.

multi-objective approach based in all of the fitness functions used till now (3 objectives) and, on the other hand, we select two of the fitness functions to build another approach with 2 objectives, in this case we select RA and CECPP with the aim to take advantage of the main features of both functions. The 3 objectives approach reaches an average return of 11,07% and the 2 objectives version reaches a 23.31%. The version with all the fitness functions is slightly above of SI, but with 15 operations of negative returns obtaining a total of -92%. Therefore, this approach does not improve any feature from previous versions. Nevertheless the 2 objective version achieves the lower number of operation with negative returns, a total number of 10, and achieves the second position in terms of average returns. It is worth of mention that despite the lower number of operations with negative results, the total average of negative results exceeds the mono-objective approach with the SI reaching a -183.33% (largely because one company almost reaches the -90% of return).

Testing the Automated Trading System Throughout the Industrial Analysis

Stock Markets are formed by many companies engaged in the most disparate activities. Events like the current economic recession, the "boom" of the brick, the rising cost of oil, droughts, etc. affect differently to each industrial sector. However, as we demonstrated in Chapter 4, we can found similar patterns in the time series of historical prices among companies of the same industrial sector. Thus, there are companies more interrelated than others, for example the behaviour or trends, of BMW and Volkswagen are more dependent on each other than the BMW itself with Goggle.

Results of Chapter 4 encourage us to include an experimental results in order to analyse the advantages of identifying the most attractive sectors for investments. Therefore, we test our trading system upon some industrial sectors to verify the existence of trends among them. We will conduct the same experiments that we perform in the two last experiment but splitting the companies in different sectors according to NACE Rev. 2 classification. The sectors used in next experimental results are:

- J - Information and Communication
- B - Oil and other extractive industries.
- D - Electricity, gas, steam and air conditioning
- F - Building
- H - Transport and storage
- C - Manufacturing
- L - Real estate activities
- G - Wholesale and retail trade, except repair of motor vehicles.

Figure 6.6 shows the returns (Y axis) of the six industrial sectors (six companies per sector) used with all the approaches implemented until now (X axis). Furthermore, all the experiments have been performed by the three fitness functions used in the last experiment. Results show different returns depending on the sector. Therefore, if according to Chapter 4, there is a general trend of a specific industry and our trading system was able to analyse and choose the sector related with, we could obtain generous returns. In the figure the most profitable sector is the sector G related to wholesale where join companies as Adolfo Domiguez or Inditex. This brief experiment, although not conclusive, supports the idea developed in Chapter 4 about the trends in different industrial sectors and shows how a macroeconomic analysis in this case could be used to exploit the portfolio to obtain better returns with a trading system.

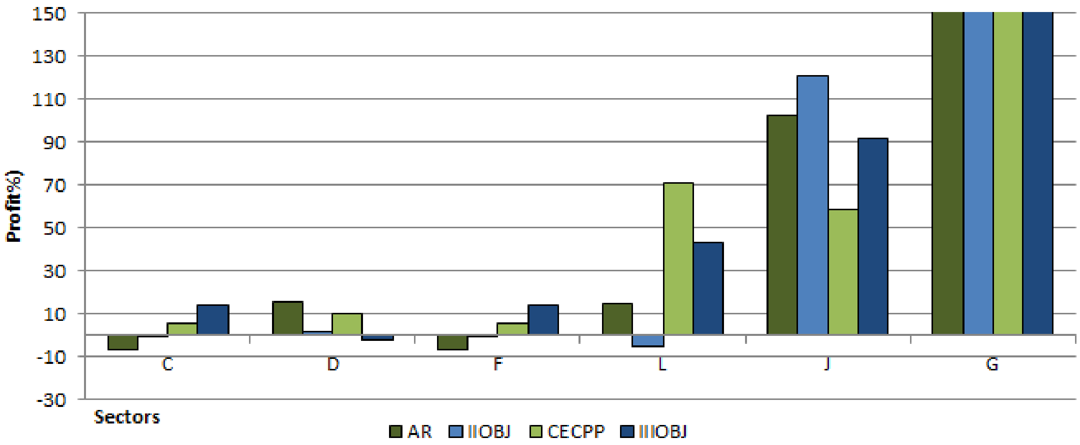


Figure 6.6: Returns of the multi-objective and mono-objective versions of the ATSs based on GE for 36 European companies divided in 6 sectors.

6.3 Trading the Stock Market: Macroeconomic, Fundamental and Technical Analysis

This section includes the final contribution of the thesis through the development of an ATS able to choose companies of different countries and exchanges. As we widely describe in Chapter 2 there are several ways to analyse the financial activities. Professional investors often use mixed techniques to invest in the exchanges, however this feature has not been transmitted to the trading systems.

In Chapter 4 we analysed an detected patterns into historical prices series of companies with related business activities. The companies forming the industrial sectors move in similar trends which can be exploited with the aim to invest in the best portfolios. In section 6.2 we test a GE-based trading system splitting the companies in different industrial activities. The trading system found sectors with more attractive profits than others. We want to take advantage of these trends, therefore we propose a novel analysis where we study macroeconomic and industry variables.

The mixture of different theories of analysis can be viewed as pieces of a puzzle whose combination can lead to predict the direction of the market most likely to succeed. We try to extend this symbiosis to our trading system combining the best features of each analysis. Nowadays most of the trading systems found in the literature are based on

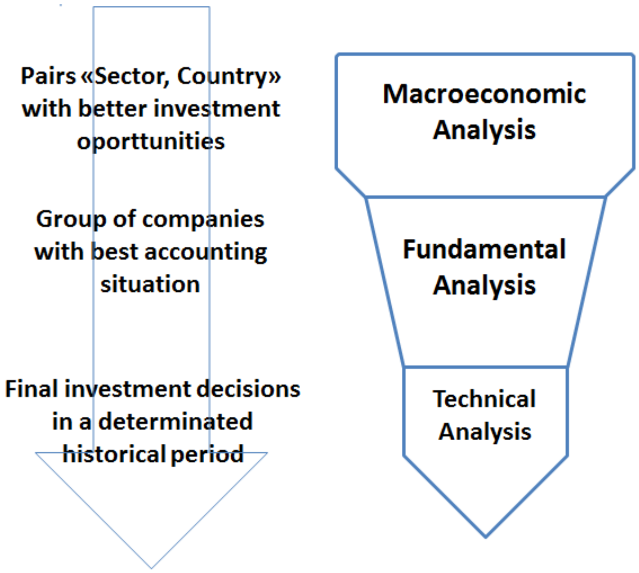

Pairs «Sector, Country» with better investment oporttunities
Group of companies with best accounting situation
Final investment decisions in a determinated historical period
Macroeconomic Analysis
Fundamental Analysis
Technical Analysis

Figure 6.7: The top-dow methodology

optimizations of indicators related to the technical analysis. Furthermore, we can find trading systems based on optimizations of fundamental rules (related to the fundamentals values of the companies). Even in last years, we can found some instances of hybrid analysis combining rules of fundamental ratios with technical indicators. However, we want to build a novel approach based on the fundamental analysis, both company and macroeconomic information (including industry), and the technical analysis. Our proposal implements a top-down methodology. The top-down method refers to make decisions from the general to the particular, that is, from the general, macroeconomic factors which substantially influence in the market securities, to finally reach a microeconomic analysis of the particular company, leading to set the target price at which the company should trade at the market. The whole hybrid analysis comprise four threads. First a macroeconomic analysis (fundamental) to chose a place among European countries. Second a industry analysis (fundamental) to select the specific industry to invest. Third a company analysis to choose the best portfolio of highly profitable companies among the specific industry. And finally, a technical analysis to optimise the time of the investment decisions.

Figure 6.7 shows the overview of the analysis implemented presenting the whole workflow split in 3 sub-modules. The first two use fundamental to locate the place where we will invest. First, through macroeconomic analysis we select a set of companies linked to a specific country and sector, and second, we use fundamentals of companies to narrow the set of companies provided by the first module. The final module, which corresponds with the trading system presented in the previous sections of this chapter, works to produce final investment decisions trough the technical analysis. Next, we describe the details of the implementation of our trading system proposal.

6.3.1 Work-flow and implementation details of the Automated Trading System based on a double Grammatical Evolution

As we have previously explained, GE uses grammars to build a set of rules that guide our trading system. With the aim of exposing the main features of our grammar we present in Figure 6.1 a fragment of the grammar used in our automated trading system.

As we mentioned, the implementation of this novel version of a trading system is developed based on two levels of analysis. One of the levels is already implemented and is based on the ATS explained and tested in the previous section of the chapter. The other level performs a macroeconomic analysis choosing the most promising companies. Next, we present the general work-flow of the whole ATS:

1. The investor selects a set of Technical Indicators
2. The investor selects a set of fundamental ratios
3. The investor selects a set of macroeconomic variables
4. The investor establish $Threshold_{buy}$ and $Threshold_{sell}$ ranges
5. Define a Grammar A in BNF to chose Countries, Sectors and companies.
6. Define a Grammar B in BNF to chose investment times.
7. Apply the ATS over a period of $T_{insample}$ days obtaining the solution rules A.
8. Built the program related to the rules A to obtain the portfolio X
9. For each company i in X
   (a) Apply GE over a period of $T_{insample}$ days to obtain the solution rules B_i.
   (b) Built the program P_i related to the rules B_i.

(c) Run the program P_i over a period of $T_{outsample}$ days.

(d) Compute the $Profit_i$ given by company

10. Compute the profit given by the Trading System

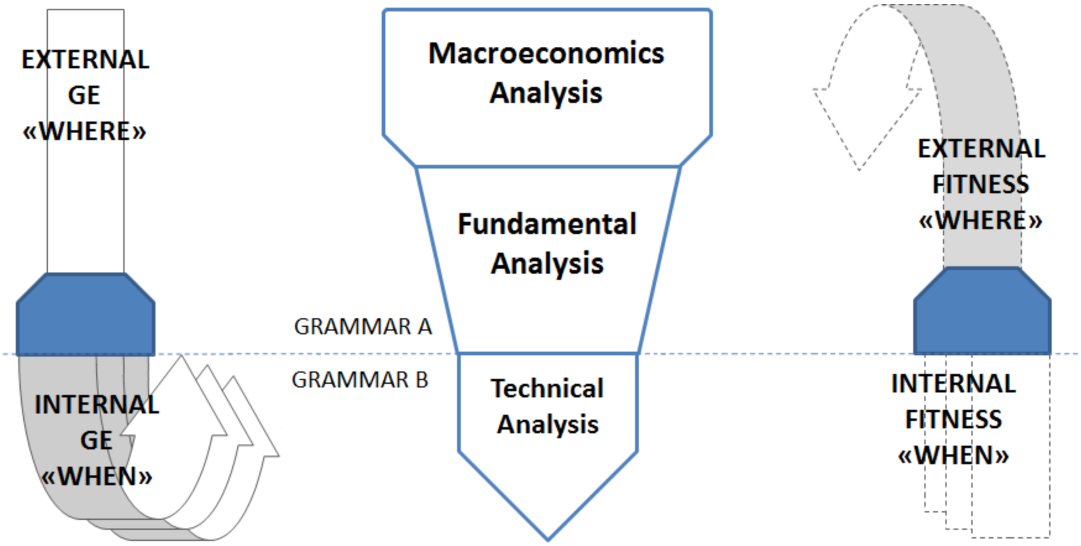


Figure 6.8: The double layer GE methodology

The work-flow can be divided in three well differentiated sections. First, we define the elements joining in the program. Second, we obtain a set of promising companies by the solution rules A (step 7 and 8), and third, we obtain the final investment signals and its returns by the solution rules B (step 9 and 10). The process of how we obtain these final signals has been widely detailed in previous sections. Next, we detail how the ATS obtains the solution rules A.

The ATS is based in a double layer of GE methodology. As the Figure 8.15 represents, the double layer methodology consists of a GE methodology (internal GE) working as the fitness function of other GE (external GE). Thus, the external GE evaluates the individuals by the following steps:

- For each individual of the population of the external GE:

1. Build the $program_i$ linked to the $individual_i$ using the Grammar A.

2. Run the $program_i$ providing a $portfolio_i$ of companies.
3. For each $company_j$ of the $portfolio_i$:
   (a) Evaluate the $Company_j$ by running the internal GE with the Grammar B.
   (b) Obtain a fitness value V_j representing the profitability of the $Company_j$.
4. Sum all the fitness values of the companies of the portfolio in a unique Value V_i.
5. Assign V_i as the fitness value of the individual representing the profitability of the portfolio.

Thus, the internal GE (the GE working as fitness function) provides a value for the portfolio (set of companies) evaluating each of its companies. Therefore, the external GE evolve its individuals seeking the most promising portfolio provided by the external grammar and evaluated by the internal GE.

Finally, we point that the codification of the population, the crossover, the mutation and the wrapper operator are inherited from the previous ATS, therefore the two ATS GE-based implement the same operators excluding the fitness function.

Grammatical Evolution: Seeking the Best Portfolio

In the Section 6.1 we present the internal grammar (Figure 6.1) which provided a set of rules to invest in a specific company. This section introduces a grammar which provides a set of rules to obtain a set of companies. Next, with the aim to present the main features of the analysis performed by this new layer, we present a fragment of the external grammar.

```
N= {<code>, <Country>, <Sector>, <Company>, <CountryVar>,
<SectorVar>, <CompanyVar>, <Period>, <Type>, <VarA>, <VarB>, ...}

T= { ");" , "," , "COUNTRY", "SECTOR", "PORTFOLIO(", "HVGDP", "ILN",
"ISN" , "NETN" , "B" , "C" , "D" , "E" , "R0", "R1", "R2", "R3",
"UVG2" , "PVG2" , "VPRI" , "QLMC", "MaxAV", "MaxDev", "MinDev"... }

S= { <code> }

P= {I, II ,III ,IV ,V ,VI, VII, VIII, IX, X ...}

I   <code> ::= <Country><Sector> <Company>

II   <Country> ::= <Country>"COUNTRY("<CountryVar>","<Period>");"
       | "BestCountry("<CountryVar>","<Period>");"

III <Sector> ::= <Sector>"SECTOR("<SectorVar>","<Period>");"
       | "BestSector("<SectorVar>","<Period>");"

IV <Company> ::= <Company>"PORTFOLIO("<CompaniesVar>","<Period>");"
       | "Bestportfolio("<CompaniesVar>","<period>", <type>);"

V   <CountryVar> ::= "HVGDP" | "ILN" | "ISN" | "NETN" | ....

VI   <SectorVar> ::= "B"<VarB> | "C"<VarC> |"D"<VarD> | "E"<VarE> | ...

VII   <CompanyVar> ::= "R0" | "R1" | "R2" | "R3" | ...

VIII   <Type> ::= "MaxAV" | "MaxDev" | "MinDev"

IX  <VarB> ::= "UVG2" | "PVG2" | ...

X   <VarC> ::= "VPRI" | "QLMC" | ...
...
...
...
```

Figure 6.9: Fragment of the external grammar used in our ATS GE-based. The grammar has the aim to build programs providing a portfolio of the most promising companies.

As the grammar introduces in Figure 6.9 the portfolios are composed by three different block of rules. The first block (<Country>) makes a portfolio of all the companies from the country which achieves better values of a economic variable (<CountryVar>) in a specific period (<Period>). The second block (<Sector>) makes a new portfolio selecting all the companies from a sector ("B", "C", "D", etc.) included in the previous portfolio which achieve better values of a industrial variable (<VarB>,<VarC>, etc.) in a specific period

(<Period>). The last block (Company) build the final portfolio selecting the companies with more favourable behaviour (<Type>) in a fundamental variable (<CompaniesVar>) over a specific period (<Period>).

We define that a company X provides the more favourable behaviour to create high returns when:

- The economic variable (<CompaniesVar>) of the company X obtains the maximum average over a period (<Period>).
- The economic variable (<CompaniesVar>) of the company X obtains the maximum standard deviation over a period (<Period>).
- The economic variable (<CompaniesVar>) of the company X obtains the minimum average over a period (<Period>).

6.3.2 Experimental results

Being the double layer ATS the last contribution, this section present results of the returns achieved by our ATS in the complete data-base. We want to empathise that the database stores information of the last 13 years of prices, volumes and fundamental ratios throughout approximately 1000 companies, it also contains macroeconomic data of approximately 40 variables over 30 countries. Furthermore, the architecture of the ATS based in a double GE trigger a complete execution of a simple GE methodology per each evaluation of the fitness function. For instance, if we execute the external GE with 500 generations, 100 individuals and we set the number of companies in the final portfolio in only 3, we obtain a total of 500*100*3 = 150 000 executions of the internal GE which executes its own methodology optimisation. Thus, the time spent by a complete execution of the ATS is very high and, therefore, it s highly recommendable the parallelisation of the system as we covered in Chapter 5, although the present thesis does not address this issue in this chapter.

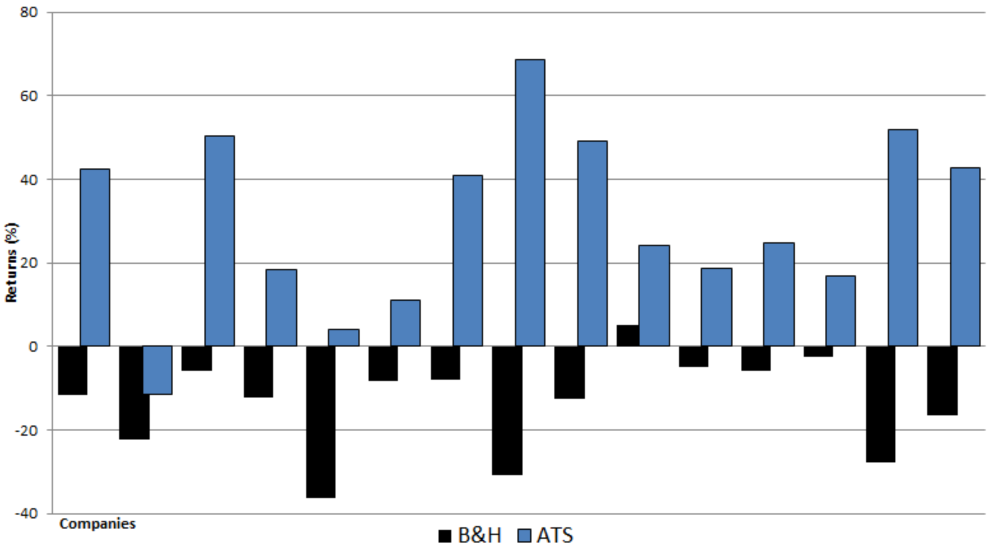


Figure 6.10: Returns of 15 operations in 2013 by the ATS performing a macroeconomic, fundamental and technical analysis. The ATS performs 1 negative investment versus 13 of the Buy and Hold

Figure 6.10 shows the returns of 15 executions of the ATS developed in the last section. The ATS has built the investments rules guided by the Sharpe Index function with data between 2003 and 2012, both the rules for building a portfolio and the rules for investment decisions, next the ATS operate in 2013 with the portfolio obtained and the built rules. The returns of the ATS are faced against the returns of the Buy and Hold strategy (B&H) over the same period. On one hand Figure **??** presents high returns in the ATS reaching an average return of 30.14% with only one portfolio with negative returns, on the other hand, the same portfolios in the B&H strategy obtain an average return of -13.35% an 13 portfolios with negative returns. Table 6.1 shows in detail the investments performed in each execution and the companies composing each portfolio (the number of companies per portfolio is fixed to 3 companies).

Table 6.1: Detailed returns of 45 companies in 2013 by the ATS performing a macroeconomic, fundamental and technical analysis. The average return reachs a 30.14% versus a -13.35% of a static strategy. The ATS performs 10 negative operations versus 39 of the B&H strategy.

COMPANY	COUNTRY	SECTOR	B&H	ATS
MCH GROUP AG	Switzerland	S	-8,38	48,78
SKANSKA AB	Sweden	F	-9,62	68,21
JM AB	Sweden	F	-17,04	10,33
ACCOR SA	France	I	-31,76	-41,43
GROUPE FLO SA	France	I	-16,91	-10,61
IMMOBILIERE DE BELGIQUE SA	Belgium	L	-18,14	17,58
GANGER ROLF ASA	Norway	B	-23,25	-2,39
Q-FREE ASA	Norway	C	13,89	21,76
FARSTAD SHIPPING ASA	Norway	H	-7,70	131,94
WILH. WILHELMSEN H. ASA	Norway	H	-13,30	-30,89
BEFIMMO SCA/CVA	Belgium	L	-14,60	28,62
MCH GROUP AG	Switzerland	S	-8,38	57,23
KGHM POLSKA MIEDZ SA	APoland	B	-26,45	34,39
GANGER ROLF ASA	Norway	B	-23,25	-0,11
REDERI A.B. TRANSATLANTIC	Sweden	H	-58,87	-22,28
SPADEL NV/SA	Belgium	C	-2,59	55,01
TESSENDERLO CHEMIE SA/NV	Belgium	C	-1,69	-18,91
BARCO NV	Belgium	C	-20,12	-2,93
MELEXIS NV	Belgium	C	-10,24	18,65
PICANOL NV	Belgium	C	-7,49	36,51
TELEKOMUNIKACJA POLSKA SA	Poland	J	-6,19	67,11
VIKING LINE ABP	Finland	H	-34,57	179,08
FINNLINES OYJ	Finland	H	-9,09	15,14
FINNAIR OYJ	Finland	H	-48,99	11,71
MCH GROUP AG	Switzerland	S	-8,38	92,68
HAFSLUND ASA	Norway	D	-16,19	30,19
WILH. WILHELMSEN H. ASA	Norway	H	-13,30	24,72
FUNESPANA SA	Spain	S	-0,57	20,58
TELEKOMUNIKACJA POLSKA SA	Poland	J	-6,19	29,48
BARRY CALLEBAUT AG	Switzerland	C	22,17	22,00
GEBERIT AG	Switzerland	G	1,96	-16,92
TAMEDIA AG	Switzerland	J	1,19	36,46
IMMOBILIERE DE BELGIQUE SA	Belgium	L	-18,14	36,23
MCH GROUP AG	Switzerland	S	-8,38	38,63
SVENSKA CELLULOSA AB SCA	Sweden	C	-1,24	-4,00

Table 6.1 continues in next page

COMPANY	**COUNTRY**	**SECTOR**	**B&H**	**ATS**
SOCIETA ELETTRICA SOPRACENERINA	Switzerland	D	-8,06	39,77
ARENDALS FOSSEKOMPANI ASA	Norway	D	6,78	11,65
CKW CENTRALSCHWEIZERISCHE K.	Switzerland	D	8,83	23,42
SOCIETA ELETTRICA SOPRACENERINA	Switzerland	D	-8,06	26,43
GANGER ROLF ASA	Norway	B	-23,25	15,32
SPONDA OYJ	Finland	L	-12,80	16,83
HURRIYET GAZETECILIK VE MAT. AS	Turkey	J	-62,75	111,89
GANGER ROLF ASA	Norway	B	-23,25	5,08
WILH. WILHELMSEN HOLDING ASA	Norway	H	-13,30	50,80
ANADOLU CAM SANAYII AS	Turkey	C	-13,28	72,41
TOTAL AV.			**-13,35**	**30,14**

Chapter conclusions In this chapter we have concluded the contributions of this work by presenting the following points:

- Using the skeleton of the automatic trading system introduced in Chapter 5, we developed a automated trading system based on grammatical evolution which implements complex features as the auto-generation of new technical indicators.

- We have tested several fitness functions including a novel multi-objective version. The results have pointed differents profiles and advantages for each fitness function suggesting advantages in its combined features.

- We have followed the conclusions of Chapter 4 on macroeconomic trends to develop a hybrid analysis that studies in different stages the macroeconomic, technical and fundamental values.

- We have tested the complete automated trading system in a hostile environment such as this economic recession reaching an average returns of a 30% with a low number of loss operations

Chapter 7

Final Conclusions and Future Research

In this thesis, we delved into the behavioural models analysing the background and related work. Thus, we modelled time series of historical prices following a Random Walk by the Geometric Brownian Motion which show us how the time series are indistinguishable in a simple way. However by a novel clustering methodology, we found patterns among sets of time series of modelled and historical prices which allowing the fully differentiation of the time series. Therefore, we present results of the divergence between the prices following random walk and the prices of stock markets. This conclusion supports the development of an Automated Trading System (ATS) able to analyse large amount of historical prices as source of information.

The clustering methodology presented allowed us to seek patterns among the companies composing the stock markets. We performed several experiments presenting similarities among sets of time series of historical prices belonging to companies developing alike economic activities. Therefore, the thesis points how the industrial sector of the Stock Markets are linked in somehow and are affected in similar ways by the economic environment. The idea of similar trends among companies of same industrial sectors makes more attractive to include a macroeconomic analysis with the aim to take advantage of these trends.

This thesis introduced the ATSs with the development of an ATS based on Genetic Algorithms (GAs) implementing an innovative version of the methodology using the named Filling Operator. The results of the experiments showed that the methodology provides several advantages, both avoiding premature convergences of the algorithm and improving final results. The ATS based on GAs was tested in 2004, where it achieved high returns and reduced the risk of losses showing as evolutionary methodologies fits perfectly with the investment systems.

After testing the GA approach, we built an ATS based in Grammatical Evolution (GE). Due to this methodology the presented system can generate more complex rules even building its own technical indicators, therefore it exhibit better performances. We introduced a novel multi-objective optimisation based on the Non-dominated Sorting Genetic Algorith (NSG-II) and GE. The multi-objective approach demonstrated high returns (average return of 20%) in very volatile periods, thus combining some of the best features of the fitness functions employed.

Once we evaluated all aspects discussed above, we obtained all the pieces to develop an ATS implemented by a double GE-layer methodology in a complete and innovative hybrid analysis. The system combines the main features of a macroeconomic, fundamental and technical analysis building set of rules by the analysis of historical periods, and finally providing us solutions formed by a portfolio of promising companies and a series of profitable investment operations. The ATS was tested in 2013 achieving average returns of approximately a 30% and only one portfolio of companies providing negative returns.

Furthermore, despite the requirements for intra-day investment in our ATS, we conclude that trading systems are able to respond in the available time frames through the parallelisation of the system. This thesis showed and compare two approaches combining GAs and parallel architectures. We used a grid computing architecture to paralelise the ATS at company level and a GPGPU framework to paralelise the system at individual level obtaining high speed-ups which reaches values up to 50x and 256x respectively.

Next we outline the main contributions presented in the thesis:

- providing results of divergence between the Random Walk and historical prices.
- presenting clusters of industrial sector with only the historical prices information.
- introducing a novel GA operator able to avoid the premature convergence.
- parallelising an ATS with two powerful architectures for intra-daily investments.

- testing two versions of the first multi-objective implementations with GE

- introducing a ATS which builds and evaluates its own technical indicators.

- presenting hybrid analysis unifying macroeconomics, fundamentals and technicals.

7.1 Future Research

This thesis has covered a wide range of topics and has presented valuables contributions in different fields. Therefore, the system improvements and research lines are countless. Next, we propose some significant points in the future research work which should follow this thesis.

- In the thesis we included several fitness functions and two multi-objective approaches, however there are other fitness functions which claim to improve the accuracy of the evaluations of a investment strategy. We consider an important matter to perform experiments to rank the utilities of the fitness functions. Therefore, we should seek to the best fitness function or the best combinations of them.

- As some results of the last chapter pointed, the fitness functions could be correlated with different strategies, sectors or companies. Therefore, we could develop an interesting research if we include the possibility into the grammar to choose the fitness functions depending of different factors.

- We do not perform any money management, techniques to control the profit and losses, as the named "stop loss", that can be critical to increase profits. The strategies in the professional world usually implement such techniques, which increase the returns exponentially, therefore we consider this feature one of the most important points to be developed.

- We used six main technical indicators allowing the creation of new versions, however it would be interesting to expand the type of technical indicators used. For example would be interesting to use indicators using data of opening and closing prices as Commodity Channel Index.

- Once we decide to invest in a particular company, we use a set of rules to manage the investments procedures. The system could be improved by performing a co-evolution methodology to manage the entry and exits for an individual investment. Thus, we could use a two set of rules with the objective of optimising the entries and exits.

- We have developed a ATS providing multiple features and the ability of building very flexible solutions. That fact causes a enormous space of solutions. We could perform a analysis of the convergence of the algorithm to analyse its performance in this regard

As we noted previously, there are countless improvements that we could apply to the system, however if we want to convert the ATS implemented into an application to make investments into the professional world, we should consider the next points:

- We should perform a study of the collateral parameters working in our system, as the percent of crossover or mutation, length of the chromosome, individuals, generations, number of wrappers or values of the buy and sell thresholds.
- We tested several versions of ATSs into different indexes and countries, nevertheless the system would offer more confidence to investors with bigger battery of tests.
- Due to the complexity task to obtain technical and fundamental data, the system could be more accessible if it is connected directly to a stable platform providing a database where we can find all the necessary data.
- With the aim of investing at real time, the system should implement a framework able to process flows of real-time data.
- Becoming a real investment tool requires a graphic interface which provides accessibility and usability for the ATS.

7.2 Related Publications and merits

As a result of the work we have developed in this dissertation we have contributed with twelve publications of our work in several journals, conferences and books. Furthermore we want to emphasise that this thesis won the **first award** to the best PhD work in MAEB congress at the end of 2013. Next we present the related twelve publications and the contributions related with the present thesis:

Publications in international journals

1st Publication

Contreras I., Jiang., Y. Hidalgo J., Nuñez-letamendia L. “Using a GPU-CPU architecture to speed up a GA based in real-time system for trading the Stock Market ” in Soft Computing, February 2012, Volume 16, Issue 2, pp 203-215.

Contributions : In this paper we expose the paralelisation of an ATS through an architecture based in GPGPU using CUDA. The results demonstrate that it is possible to parallelise calculations and that this procedure improves the scalability of the system and possible application to trading systems in real time obtaining accelerations up to x264.

2nd Publication

Arnaldo, I.; Contreras, I.; Millán-Ruiz, D.; Hidalgo, J.I.; Krasnogor, N. “Matching Island Topologies to Problem Structure in Parallel Evolutionary Algorithms”. International Journal of Soft Computing: Special Issue on Bio-inspired Algorithms with Structured. July 2013, Volume 17, Issue 7, pp 1209-1225.

This publication is accepted in September 2012 , it is a paper member of a Special Issue which joined only six papers of a total of 43 candidates (14% acceptance rate).

Contributions: In this paper we expose the correlation between problems and parallel architectures islands. This correlation is studied in theoretical and real scenarios using NK-landscape problems, the B-graphs and multi-skill centre problem. This article provides a topology of islands to our custom problem and optimal performance. The results show that the choice of architecture is vital for optimum resolution of the problem.

3rd Publication

Contreras, I.; Arnaldo, I.; Hidalgo, J.; Krasnogor, N. “Blind Optimisation Problem Instance Classification via enhanced Universal Similarity Metric” Memetic Computing Springer Verlag. Accepted for publication (In Press)

Contributions: In this paper we present a novel methodology based on Kolmogorov complexity improving the previous approaches. This methodology is based on compressors and it is used to study the feasibility of a trading system based on the analysis of time series of historical prices. Thus, we use the methodology to analyse series of real and modelled prices.

4th Publication

Contreras, Nuñez-Letamendía L.; Hidalgo, J.. "Time series of Prices and Randomness: Undercover Patterns". International Journal of Mathematical Finance. (Sent)

Contributions: In this paper we present a study on the price randomness and on the patterns shared by real companies. Further we study the time series of historical prices of industrial sectors providing arguments for the inclusion of macroeconomic analysis in the trading system.

Publications in books or conferences

5th Publication

Arnaldo I.; Contreras I., Hidalgo J. Krasnogor N. "Relación entre la Estructura de un Problema y la Topología de Islas en Algoritmos Evolutivos Paralelos". 4th IT congress in Spain: MAEB 2013 - CEDI, 17-20 September 2013 - Madrid, Spain.

Contributions: Using different statistical experiments the article exposes the correlation between problems and parallel architectures islands. This article provides a first approach to the islands optimal topology for solving our trading system.

6th Publication

Contreras I., Hidalgo J., Nuñez-Letamendia L.; Diego J. "Sistemas de trading en la recesión económica: gramáticas evolutivas, relación fitness-objetivos, y sectores empresariales". 4th IT congress in Spain: MAEB 2013 - CEDI, 17-20 September 2013 - Madrid, Spain.

Contributions: This paper describes several versions of an ATS based in GE which are guided by different fitness functions. The paper includes implementation details of a novel multi-objective version GE-based. Furthermore we develop a brief analysis of the obtained returns if we divide the data-base of companies by industrial activities.

7th Publication

Contreras I., Hidalgo J., Nuñez-letamendia L. "Combining Technical Analysis and Grammatical Evolution in a Trading System". Proceedings of the 6th European Event on Evolutionary and Natural Computation in Finance and Economics : EVOFIN, 11-13 April, 2013- Málaga, Spain.

Contribution: Analysis of an ATS in the hostile environment of the economic recession. This paper shows the results of a updated version of an ATS based in GA into a ATS based in the flexible Grammar Evolution.

8th Publication

Contreras I., Hidalgo J., Nuñez-letamendia L. Jiang., Y. Parallel Architectures and Bioinspired Algorithms. Chapter: "Parallel Architectures for Improving the Performance of a GA based trading System" in Springer, February 2012, Series Volume 415, pp 189-218

Contributions: This article presents comparisons between two parallelisation architectures, the grid computing and GPGPU computing. The results show how the grid provides a parallelisation of an ATS optimal for less than 2000 individuals, however from this number, the implementation working on the gpu obtains higher speed-ups.

9th Publication

Contreras I., Jiang. Y., Hidalgo J., Nuñez-letamendia L. "Arquitectura GPU-CPU para la aceleración de un sistema de inversión bursatil en tiempo real. Congress on Numerical Methods in Engineering". Coimbra, 14 a 17 de June 2011.

Contributions: This article describes the implementation and results of the parallelisation of an ATS on a GPGPU architecture. The results shows a speed-up which allows a real-time reply of the ATS.

10*th* *Publication*

Contreras I., Hidalgo J., Nuñez-letamendia L. “A GA combining Technical and Fundamental Analysis for trading the Stock Market”. Proceedings of the 5th European Event on Evolutionary and Natural Computation in Finance and Economics : EVOFIN, 27-29 April 2011 - Torino, Italy.

Contributions: This paper shows a trading system based on GAs and guided by a hybrid analysis using fundamentals and technicals. Furthermore, we introduce innovative operators, as the cross-dominant and the filling-operator, preserving the diversity of the population and incrementing the final fitness values.

11*th* *Publication*

Martín D.; Villanueva-Oller J.; Contreras I. Hidalgo J. “Supercomputadores virtuales y computación distribuida con BOINC”.Annals in computer science 2010 : CES Felipe Segundo

Contributions: In this chapter we present an extensive study of our first paper (bellow) of the implementation details and experimental results, measuring the run-times and financial returns of the parallelisation of an ATS in a grid architecture.

12*th* *Publication*

Villanueva-Oller J.; Martín D.; Hidalgo J., Contreras I.. “El proyecto Falúa: computación distribuida mediante BOINC en el Campus de Aranjuez de la UCM”. 3rd IT congress of Spain : CEDI, 7-10 September 2010 - Valencia, Spain.

Contributions: In this paper we present implementation details and the runtime results of an parallelisation of an ATS in a platform of volunteer computing through a corporative grid and some well-known volunteers.

7.3 Research Projects and Grants

This work has been partially supported by the following projects and grants:

- Project EIFIN : Supercomputation and finances project funded by the IE Business School.
- Project: AMBU: Arquitectura de servicios de supercomputación en la nube, funded by the Spanish Government, Ministerio de Industria, Turismo y Comercio, Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica 2008-2011. Avanza Competitividad I+D+I: TSI-020100-2010-962
- Project: IYELMO: Plataforma de servicios en la nube para operaciones en mercados financieros, funded by Spanish Government, Ministerio de Ciencia e Innovación. NNPACTO-IPT-2011-1198-430000
- Mobility Grant Orden ECD /3628/2011, de 26 de diciembre: A five-month stay in the Nottingham University has been funded with a mobility grant from the Spanish Government, Dirección General de Política Universitaria, Ministerio de Educación, Cultura y Deporte.
- Genil award 2013 : A one-month stay in the university of Granada with a mobility grant related to the Genil award in the MAEB congress.

Chapter 8

Resumen en Español

8.1 Introducción: Mercados y metodologías

Los mercados de valores son sistemas no lineales determinados por infinidad de variables en las que un pequeño cambio puede transformar drásticamente los movimientos de un activo [37]. Además, las operaciones de los inversores en el mercado de valores dependen de las emociones de miles de otros inversores, que a su vez dependen de su edad, situación económica o social, nervios, hormonas, conocimientos, antecedentes, etc. Este gran número de factores han llevado, con frecuencia, a sugerir los mercados de valores como sistemas impredecibles o incluso modelos aleatorios. La búsqueda de modelos de predicción (reglas de inversión para el mercado basadas en el análisis técnico, fundamental o macroeconómico), junto con el intento de comprender los modelos de comportamiento en la formación de precios de los mercados (por ejemplo, los caminos aleatorios o la hipótesis del mercado adaptativo) han establecido una de los ramas más importantes en la historia de la investigación relacionada con las finanzas .

Los sistemas de comercio automatizados (SACs) son sistemas predictivos basados en reglas en reglas que usan información del mercado, de empresas o macroeconómica incrustadas en algoritmos que buscan la mejor combiación de reglas para guiar las operaciones en bolsa en un intento de obtener la máxima rentabilidad posible para un período específico. Consideramos que un SAC es una gran ventaja para el estudio de las posibles inversiones financieras debido a la complejidad del mercado. Las operaciones de los mercados están influenciadas por una gran cantidad de factores de diferentes fuentes. En primer lugar, estas se ven afectadas por factores asociados a políticas gubernamentales, factores naturales, comercios internacionales, sentimientos de mercado, factores políticos, etc. Por tanto, es muy complejo seguir correctamente un flujo de información y posteriormente actuar según las consecuencias que esta información implica. En segundo lugar, los inversores se ven afectados por las llamadas finanzas del comportamiento. Brokers o inversores en general pueden verse afectados por emociones humanas, por lo que su comportamiento en el mercado de valores puede no ser objetivo. La alta presión inducida por el manejo de grandes volúmenes de dinero es la razón principal capaz de provocar comportamiento como aversión a la pérdida, exceso de confianza, reacciones exageradas, efectos manada y otros similares.

Podemos resumir que en los últimos años el interés en el SAC se ha extendido debido a los siguientes factores:

- (i) La cantidad explosiva de información disponible de empresas y mercados .
- (ii) El avance de la informática, sobre todo respecto a la capacidad de computación de bajo coste y la creación de algoritmos avanzados.
- (iii) La expansión del proceso de globalización .
- (iv) El intento de evitar aspectos psicológicos que influyen en los procesos de inversión.

Encontrar series óptimas de decisiones de inversión implica una inspección de complejos espacios de búsqueda. La complejidad del espacio de búsqueda depende de la cantidad de los parámetros analizados. Según aumenta el número de combinaciones, el espacio de búsqueda crece de manera exponencial. Los SAC pueden ser desarrollados desde un simple conjunto de ramificaciones *if-then* a modelos más sofisticados que utilizan métodos como la inteligencia artificial, la teoría del caos, fractales, algoritmos evolutivos,e etc.

El entorno adverso descrito sugieren a las meta- heurísticas como uno de los mejores métodos para encontrar buenas soluciones en un cortos período de tiempo. La aplicación de las meta-heurísticas en los sistemas de comercio ha tenido un rápido crecimiento, tanto en el mundo científico como en el profesional. La literatura sobre las meta-heurísticas es extensa y en esta tesis nos centraremos en una de las ramas más conocidas y de mayor éxito dentro de este tipo de métodos: las meta-heurísticas evolutivas comúnmente conocidos como algoritmos evolutivos. Los algoritmos evolutivos son un conjunto de metodologías de búsqueda y optimización inspiradas y basadas en los principios y teorías del mundo de la biología. Estos procedimientos se caracterizan por emular comportamientos evolutivos de la naturaleza que se basan en la supervivencia de la mejor solución (individuo) posible entre un conjunto de otras soluciones (población). En la literatura se pueden encontrar gran número de estudios previos que documentan el uso de algoritmos evolutivos para diseñar y optimizar SACs (como por ejemplo [3], [83], [54], [11],[107], citeLohpetch2009, [72], [32], [12], [33], [1], [2] y [99]). Además, tanto las grandes empresas de inversión y como las pequeñas han comenzado a utilizar los algoritmos evolutivos para construir SAC. En esta tesis centramos nuestro trabajo en dos conocidos algoritmos evolutivos, los Algoritmos Genéticos (GA) [50] y la Evolucion Gramatical (GE) [24]. Los GAs son las metodologías más populares y probadas entre los algoritmos evolutivos, mientras que las GEs son relativamente una nueva metodología considerada más compleja y flexible. Ambos algoritmos son métodos de optimización que utilizan operadores inspirados en los principios darwinianos de la evolución, como el cruce, mutación o selección para evolucionar conjuntos de soluciones con el objetivo de encontrar una buena solución en un periodos cortos de tiempo.

8.2 Predictibilidad de Mercados: Patrones Ocultos en Series de Precios Historicos

La teoría de los paseos aleatorios afirma que la sucesión de cambios de precio de un valor de la bolsa no es más predecible que el valor de un número aleatorio. Un paseo aleatorio es una sucesión de cambios independientes en el precio y con la misma distribución que una variable aleatoria, por tanto los datos del pasado no pueden constituir una fuente de información para predecir acciones futuras. Desde la presentación del primer trabajo por Louis Bachelier [5] ("The Theory of Speculation") sugiriendo este comportamiento en los mercados financieros, muchos autores han desarrollado y apoyado esta teoría [27]; [56]; [94]; [73]; [26], [96], [39] y [75]. Sin embargo, en los años 80 algunas investigaciones presentaban anomalías en la aceptación de los paseos aleatorios ([20], [19], [109]) y probaron que las distribuciones de los rendimientos mostraban colas inusualmente largas y asimétricas. Esta época presenció el surgimiento de un paradigma que ofrece una manera alternativa de explicar el comportamiento de los precios en la bolsa, las Finanzas del Comportamiento. Más recientemente Lo publica uno de los libros más conocidos que discuten la teoría del paseo aleatorio "A Non-Random Walk Down Wall Street" [68]. Los autores señalan el hecho de que la existencia de caídas y crecimientos de largos periodos son una clara existencia de que el mercado no es aleatorio ya que estas tendencias son muy frecuentes y largas para ser explicadas con los paseos aleatorios. Shiller [106] también subraya que la volatilidad es demasiado alta para apoyar la EMH.

8.2.1 Metodología

En esta sección nosotros proponemos el uso de una novedosa metodología para probar la existencia de patrones en series temporales de precios históricos pertenecientes a compañias listadas en la mercados bursatiles. La metodología se basa en la clasificación ciega mediante clusterings ([112]) de series temporales usando distancias dadas por compresores de software. Clustering es una de las técnicas de clasificación no supervisadas más importantes. Otros trabajos ya han usado esta técnica con series temporales de precios históricos, como por ejemplo para examinar precios y buscar formas que indiquen el rendimiento futuro del valor [18] [79]. Existen muchos tipos de algoritmos de Clustering [53], para este trabajo hemos empleado el Clustering jerárquico [55] usando el metodo de Complete Linkage Method [29].

La novedosa metodología que proponemos [25] utiliza una versión alternativa de la NCD ó Distancia de Compresión Normalizada (basada en la complejidad de Kolgomorov) para medir la distancia entre dos series temporales de precios históricos. La NCD ha demostrado ser una buena media de clasificación en numerosos trabajos previos [115], [9], [90], [110], [61] y ha sido utilizada en variadas areas, tal y como en glasificacion de genomas, piezas de música, plagio de software, registro de imagen, filogenia, comparación de la estructura de proteínas, genotipificación, subclasificación de tumores, detección de virus, etc. ([21],[61],[97], [45], [6]). Nuestra propuesta es una modificación a esta medida que es más precisa gracias al manejo de diccionarios de compresión. La propuesta distancia, la cual hemos denominado mNCD (modified Normalised Compression Distance) es la siguiente:

$$mNCD(x,y) = \frac{Max\{C(x|D_y), C(y|D_x)\}}{Max\{C(x|D_x), C(y|D_y)\}} \tag{8.1}$$

Donde C representa una forma medible para aproximar la complejidad de Kolmogorov usando un programa compresor, C(i) es el tamaño comprimido de i y $C(i|D_j)$ es el tamaño comprimido de i utilizando el diccionario de compresión de j .

Para medir correctamente estas distancias hemos simplificado la información de las series temporales (números reales) discretizando las series temporales para poder buscar patrones en los siguientes dos pasos:

- Primero hemos transformado las serie de precios por series de rendimientos.

$$rendimiento = \frac{Precio_{t+1}}{Precio_t} \tag{8.2}$$

- Y en segundo lugar hemos codificado los rendimientos por rangos.

$$Day(rendimiento) \begin{cases} rendimiento < 0.8 = "A" \\ 0.8 < rendimiento < 0.9 = "B" \\ 0.9 < rendimiento < 0.95 = "C" \\ 0.95 < rendimiento < 1.0 = "D" \\ 1.0 < rendimiento < 1.05 = "E" \\ 1.05 < rendimiento < 1.1 = "F" \\ 1.1 < rendimiento < 1.2 = "G" \\ rendimiento > 1.2 = "H" \end{cases} \tag{8.3}$$

De la misma forma que las canciones de los mismos estilo musical comparten patrones, o software maliciosos y virus comparten características comunes, esperamos que las series temporales de precios compartan los patrones que son invisibles a un análisis típico. Nuestro enfoque se basa en la suposición de que los patrones formados por series de precios aleatorias serán más similares entre sí que a las series de precios reales.

8.2.2 Resultados Experimentales: Precios Modelados Aleatoriamente y Precios de Mercado

El objetivo de esta sección es demostrar que los mercados bursátiles no siguen un comportamiento determinado por paseos aleatorios, y por lo tanto son de alguna forma predecibles. El primer paso en nuestro conjunto de experimentos consiste en en comprobar si los precios modelados artificialmente siguiendo paseos aleatorios son distinguibles de los precios reales de mercado. Para ello generamos series de precios basandono en el Movimiento Geométrico Browniano mas conocido por su nombre en ingles Geometric Brownian Motion (GBM). GBM es uno de los procesos estocásticos con incrementos independientes más usados y es definido mediante la siguiente ecuación:

$$\begin{cases} dS_t = S_t\mu dt + S_t\sigma dW_t \\ dW_t = \epsilon\sqrt{dt} \end{cases} \tag{8.4}$$

Donde St is el precio en un determinado momento t, dt es el intervalo de tiempo, μ es el incremento, σ es la volatilidad, Wt es un proceso de Weiner y ϵ es una distribución normal. Finalmente si transformamos la equacion 8.4 mediante una forma finita obtenemos la Equación 8.5:

$$\Delta S_{t+\Delta t} = S_t\mu\Delta t + S_t\sigma\epsilon\sqrt{\Delta t} \tag{8.5}$$

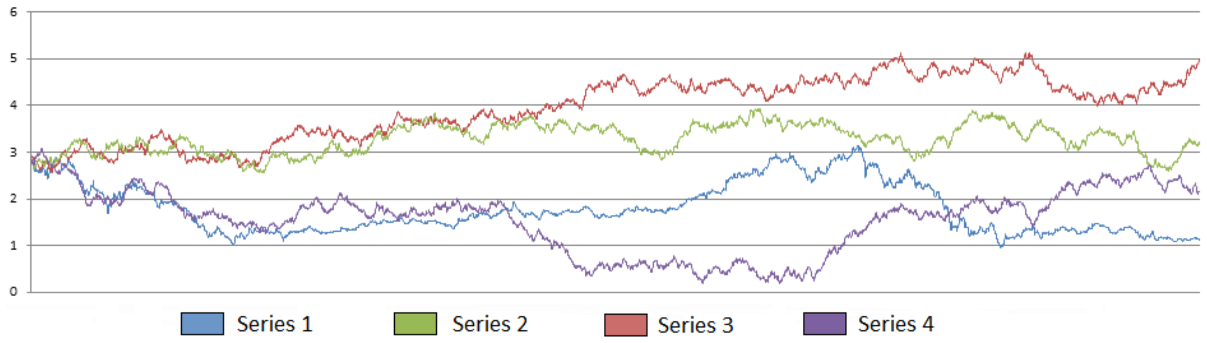


Figure 8.1: En azul serie de precios históricos diarios de A2A (2001-2010). El resto de las series modeladas con GBM

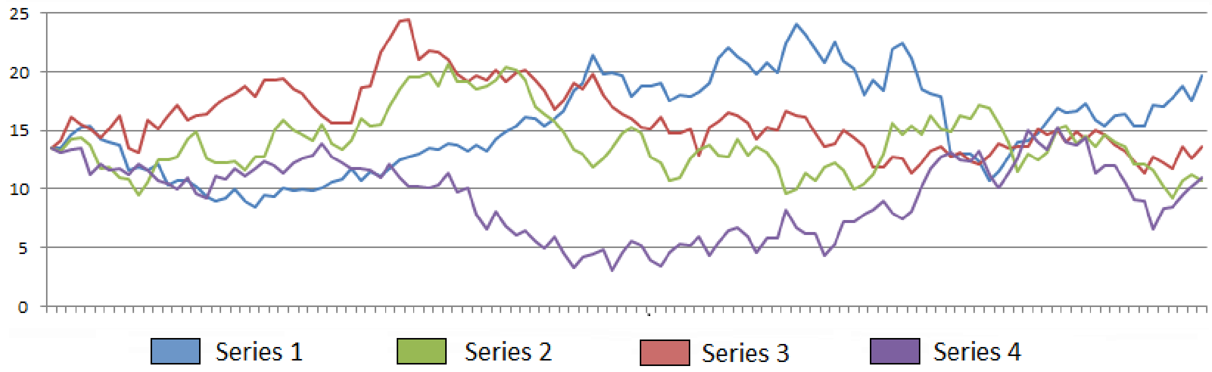


Figure 8.2: En azul serie de precios históricos mensuales de Repsol (2001-2010). El resto de las series modeladas con GBM

La figuras 8.1 y 8.2 reflejan series temporales diarias y mensuales (respectivamente) de 4 empresas durante el periodo de 2001 a 2010. El eje vertical representa los precios y el eje horizontal el periodo. De las cuatro series mostradas en cada gráfica solo una (en color azul) esta correlacionada a una serie temporal de precios históricos (A2A S.P.A y Repsol). La figuras muestran la complejidad de diferenciar series modeladas artificialmente con aquellas que no lo son, mostrando comportamientos indistinguibles tanto en periodicidad diaria como mensual.

En los siguientes experimentos se usan grafos no dirigidos (arboles binarios) construidos mediante el método de fuerza dirigida [43]. Cada nodo hoja está vinculado con una serie temporal específica, los nodos restantes son nodos intermedios que representan las distancias entre ramas. Por tanto, cuanto mas pequeña sea la distancia, mayor similitud entre las hojas.

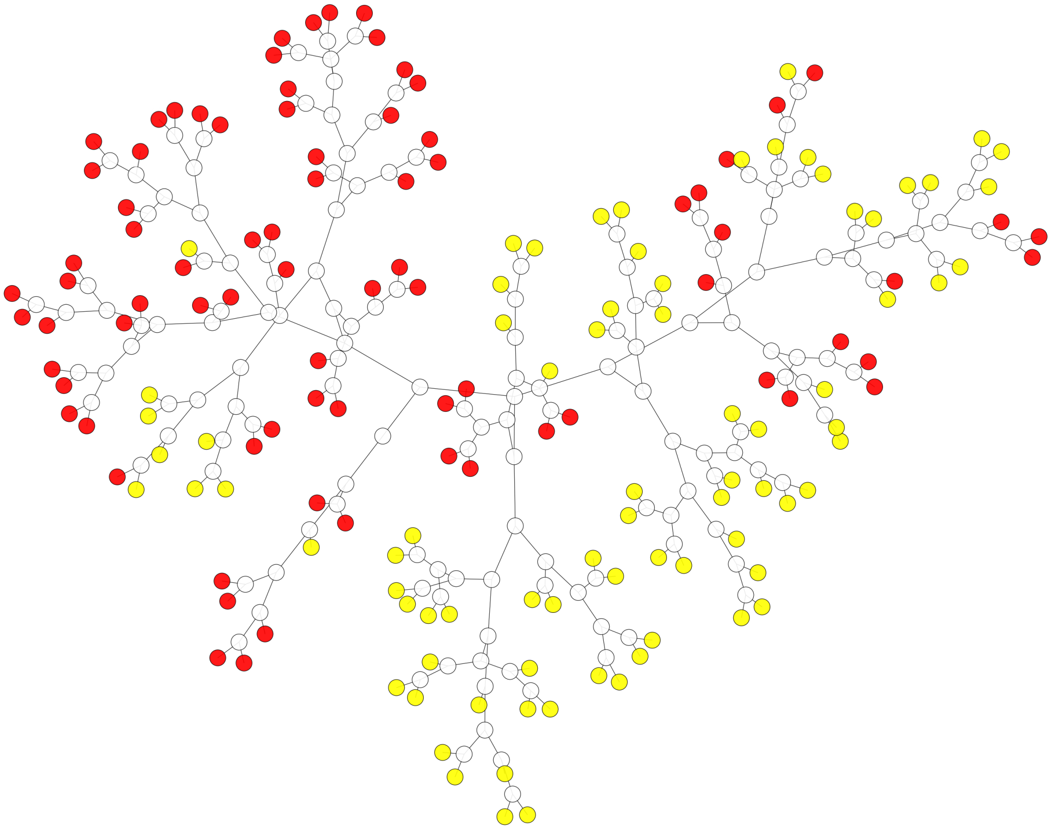


Figure 8.3: Clustering de series mensuales de 80 compañias listadas en Europa entre el 2001 y 1012 y 80 series modeladas mediante la GBM

La Figura 8.3 presenta la clusterización de series mensuales modeladas y reales. Los nodos en color rojo representan las series temporales modeladas y los nodos de color amarillo las series temporales reales. El grafo muestra claramente dos partes bien diferenciadas aunque con parte de las series clasificadas erróneamente. Dividiendo el gráfico en dos secciones, en la sección a la izquierda obtenemos un 13,43% (9 nodos amarillos) de error de clasificación y un 86,57% (58 nodos rojos) de éxito. En la sección que queda a la derecha obtendríamos unos porcentajes de un 23,65% (22 nodos rojos) de error de clasificación y un 76,35% (71 nodos amarillos) de éxito. El número de observaciones en este periodo no es alto al tratarse de series mensuales (120), por lo tanto, el número de observaciones puede estar correlacionado con los errores de clasificación. En la Figura 8.4 proponemos otro experimento similar con la diferencia que cambiamos la periodicidad de los datos, siendo

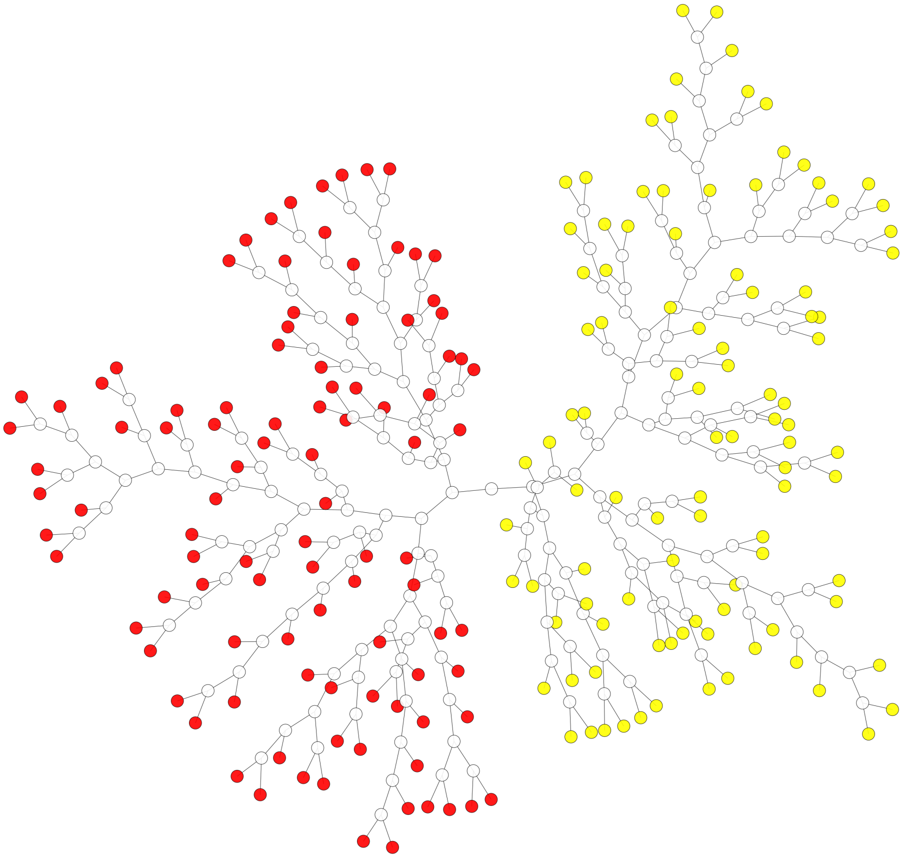


Figure 8.4: Clustering de series diarias de 80 compañias listadas en Europa entre el 2001 y 1012 y 80 series modeladas mediante la GBM

en este caso diarios (2500 observaciones aproximadamente). Este experimento presenta dos secciones totalmente diferenciadas de los dos tipos de nodos, y por lo tanto clasificando exitosamente las series de precios de manera totalmente ciega. Los resultados demuestran unos patrones diferentes entre la formación de precios en los mercados de valores y la formación de precios siguiendo paseos aleatorios.

8.3 Patrones ocultos entre industrias

En linea con los argumentos expuestos en por Moskowitz and Grinblatt en [81] en esta sección realizamos una serie de experimentos para comprobar si las empresas desarrollando diferentes actividades industriales estan enmarcadas en diferentes tendencias. Aunque estudios anteriores han mostrado poco impacto de las industrias en las tendencias de mercado, es posible encontrar otros trabajo relacionados con nuestra linea argumental [95], [91], [66], [84] ó [48].

Con la misma metodología utilizada en experimentos anteriores (Section 8.2.1) buscamos patrones comunes entre empresas pertenecientes al mimo sector industrial. Estas pruebas nos aportaran pruebas directas sobre la previsibilidad de información dada con valores fundamentales. Y de manera más general, nuestro marco empírico proporciona una nueva medida que se podrá ser utilizada para ampliar el trabajo existente sobre la predección de rendimientos. Patrones similares daran lugar a que estrategias similares o idénticas puedan obtener altos rendimientos en un mismo sector industrial.

Hemos usado una base de datos de empresas listadas en Europa en el período 2001-2013 y con al menos 5 últimos años de datos históricos en la bolsa. Las empresas han sido clasificadas en base a los diferentes sectores industriales de acuerdo con la NACE Rev. 2 [1] Queremos notar que la división propuesta es bastante general, y por tanto no se buscan clasificaciones tan precisas como en los experimentos anteriores.

8.3.1 Resultados Experimentales: Patrones entre empresas desarrollando actividades similares

La figura 8.5 muestra Clusterings de empresas pertenecientes a industrias con ocupaciones no relacionadas. La figura esta compuesta por tres grafos (A, B y C). El grafo A de industrias de *extracción y refinación de petróleo* y *manufacturación de productos alimenticios, bebidas y tabaco* (verde y azul), el B *telecomunicaciones e informática* y *manufacturación de costura y textil* (gris y azul) y finalmente el C *suministro y gestión de gas, electricidad, agua y residuos* y *manufacturación de maquinaria, equipos, vehículos y medios de transporte* (rosa y negro).

[1] La NACE es la nomenclatura de actividades económicas en la Unión Europea; Esta clasificación se basa en las unidades estadísticas que corresponden a una actividad económica específica (o grupo de actividades similares).

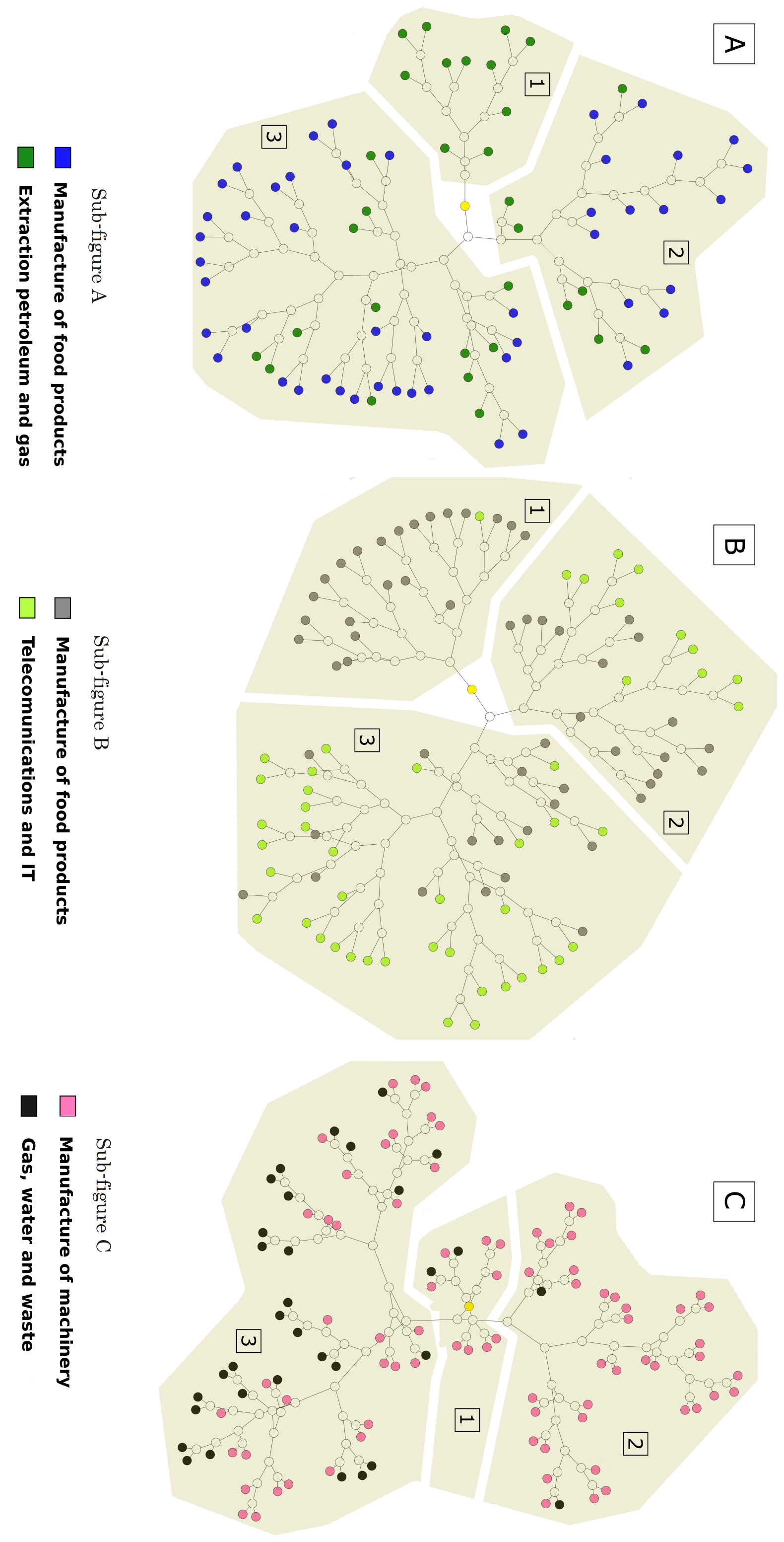

A
1
2
3
Sub-figure A
■ Manufacture of food products
■ Extraction petroleum and gas
B
1
2
3
Sub-figure B
■ Manufacture of food products
■ Telecomunications and IT
C
1
2
3
Sub-figure C
■ Manufacture of machinery
■ Gas, water and waste

Figure 8.5: Agrupaciones entre empresas pertenecientes a diferentes industrias

Los nodos blancos son nodos de ramificación intermedios y el nodo raíz de cada árbol es de color amarillo. En estos tres grafos se puede observar cierto grado de agrupamiento, y en términos generales parece que las empresas que realizan diferentes actividades económicas generan series de precios que no guardan la misma similitud entre ellos que las empresas que actúan en el mismo sector económico. Con el objetivo de sondear el éxito de nuestra agrupación, dividimos el árbol en sub-grupos. Partimos los grafos a partir de la raíz hasta obtener sub-grupos con un número de empresas (hojas) inferior que el número de empresas contenidas en el sector más amplio. La Tabla 8.1 muestra un resumen de los resultados obtenidos.

Table 8.1: Porcentajes de éxito de clasificación del Clustering de la Figura 8.5

Grafo		**A**		**B**		**C**	
Sector (Color)		**Verde**	**Azul**	**Gris**	**Verde**	**Rosa**	**Negro**
Empresas Tot.		48	31	54	48	79	33
Sub-Cluster 1	Empresas	11	0	22	1	9	2
	Porcentajes	**100%**	**0%**	**96%**	**4%**	**82%**	**18%**
Sub-Cluster 2	Empresas	7	15	16	11	37	2
	Porcentajes	**31%**	**69%**	**60%**	**40%**	**95%**	**5%**
Sub-Cluster 3	Empresas	13	33	16	36	33	29
	Porcentajes	**29%**	**71%**	**31%**	**69%**	**53%**	**47%**

Podemos afirmar que, incluso con el bajo detalle y la extrema generalización de la clasificación NACE, el sistema puede agrupar aproximadamente las empresas que desarrollan actividades económicas similares tan sólo utilizando las series temporales de precios históricos. Por otra parte, queremos señalar que y que se podía atribuir a un mismo sector la capacidad de generar sub-actividades, es decir, la misma industria se puede dividir en grupos más pequeños que se comportan de diferentes maneras entre sí. Los resultados muestran agrupaciones indicando tendencias de mercado compartidas y, por lo tanto, señalando que la inclusión de un análisis industrial puede reportar ventajas en la toma de decisiones de inversión.

8.4 Operando en el Mercado de Valores: Algoritmos Genéticos y Análisis Financiero Híbrido

Comprobada la factibilidad de un sistema de predicción basado en series temporales y demostrada la existencia de tendencias macroeconómicas en las diferentes industrias, proponemos el desarrollo del sistema completo a través de diferentes etapas. Iterativamente y mediante varias aproximaciones, testearemos y analizaremos las piezas que compondrán el sistema final. Las primeras fases describen un sistema de comercio automatizado basado en el estudio de seriies temporales de precios y volúmenes por parte de 4 indicadores pertenecientes al análisis técnico y del estudio de valores fundamentales mediante 4 ratios derivados del análisis fundamental de empresas. El sistema utiliza un motor de optimización guiado por una versión modificada de un algoritmo genético. Esta versión no permite presentar una metodología que incluye un novedoso operador llamado GAwFO. En esta sección describimos y testeamos (comparándolo con un Algoritmo Genetico simple (AGs)) las características básicas que proporcionan mecanismos para evitar una convergencia prematura del algoritmo y mejorar los resultados de rendimiento finales. A continuación exponemos el flujo básico por ciclo de este novedoso algoritmo:

- Inicialmente disponemos de una población de n individuos (generados aleatoriamente si es la primera generación).
- Evaluamos toda la población y seleccionamos el individuo con mayor valor para que no pueda ser modificado (elitismo).
- Seleccionamos mediante torneo $S = (n/2) - 1$ individuos que participan en la creación de una nueva población.
- Con el conjunto de los s individuos, el algoritmo aplica una versión del operador de cruce uniforme con una probabilidad (p_c), asi generando m individuos nuevos, debido a que $p_c < 1$, en la mayoria de las generaciones obtendremos un $m < (n/2)$. El operador de cruce tiene dos características principales:
  - En primer lugar, los descendientes tienen una probabilidad mayor de ser más similares al padre que más aptitud tenga, es decir los nuevos individuos podrán tener más genes del padre que funcione mejor.

  - En segundo lugar, los cruces que den lugar a individuos ya existentes no se permiten. De manera que cuanto más convergencía se produce en la población, menos individuos generados mediante el operador de cruce.

- Con el objetivo de conservar el tamaño de la población n, el algoritmo genera $k = n - s - m$ individuos adicionales. El algoritmo genera k individuos aleatorios en cada generación. Teniendo en cuenta que k no es constante y depende del numero de cruces efectivos realizados, así cuanto más converge el algoritmo más individuos aleatorios se generan.

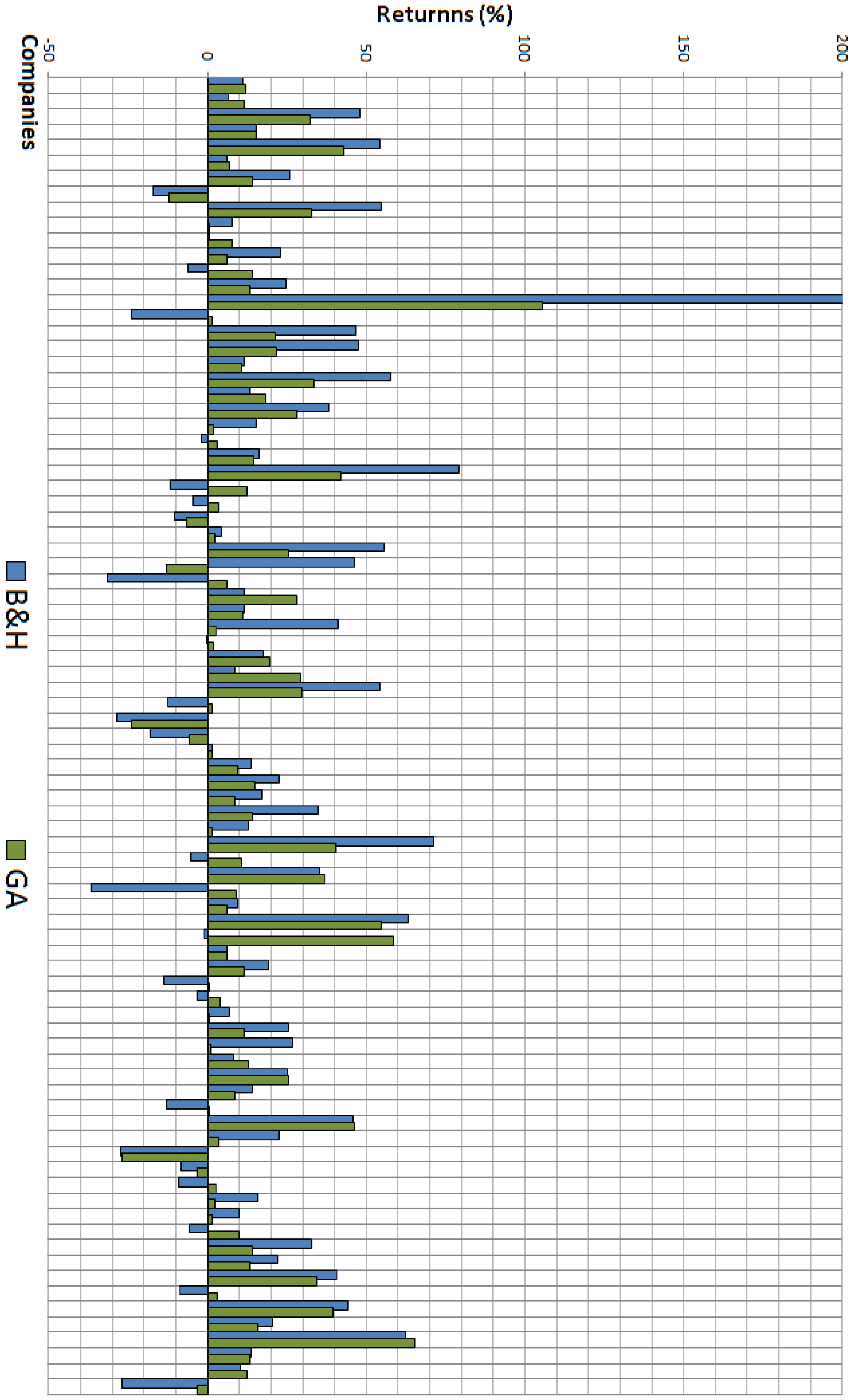


Figure 8.6: Rendimientos medios del SAC y B&H para 85 empresas del S&P 500. B&H consigue un 17.47% con 24/61 operaciones negativas/positivas y el SAC un 14.31% con 8/77 operaciones negativas/positivas

- En esta nueva población de $n = 1 + s + m + k$ individuos se aplica la clásica mutación con probabilidad p_m a los nuevos individuos generados m.
- Con la nueva población completamente generada el algoritmo regresa al segundo punto, repitiendo así el proceso hasta que el número máximo de generaciones sea alcanzado.

La función de aptitud que evalúa los individuos viene determinada por el rendimiento acumulado (AR):

$$AR_f = \prod_{f}^{i=1}(1 + DR_i) \tag{8.6}$$

Donde DR_i es el rendimiento del dia_i mediante la siguiente función:

$$DR_i = \begin{cases} \frac{P_i - P_{i-1}}{P_{i-1}} & \text{Si el SAC produce una señal a largo} \\ \frac{-(P_i - P_{i-1})}{P_{i-1}} & \text{Si el SAC produce una señal de venta corta} \\ 0 & \text{Si el SAC no produce una señal} \end{cases} \tag{8.7}$$

And where P_i is the close price of the day_i.

La figura 8.6 presenta los resultados de una serie de ejecuciones del SAC en una muestra de 85 empresas listadas en el S&P500 y las compara con la estrategia de B&H. El SAC es ajustado entre 1994-2003 y aplicado en 2004. Los resultados alcanzan altos rendimientos y minimizan las pérdidas de B&H. Sin embargo, nuestro sistema de comercio rara vez supera los altos rendimientos proporcionados por el propio mercado, en gran medida gracias al periodo en alza donde se ha ejecutado el sistema. Las ganancias medias del SAC alcanzan un 14,31%, con 8 operaciones negativas y 77 positivas, mientras la estrategia de B%H alcanza un 17,37% de ganancias con 24 operaciones negativas y 61 positivas.

Como uno de los principales objetivos del GAwFO es preservar la diversidad de la población y mejorar la convergencia del algoritmo realizamos el experimento siguiente con respecto a la convergencia de los algoritmos presentados. La Figura **??** presenta los resultados experimentales de una serie de ejecuciones del sGA y el GAwFO para 100/500 generaciones y un número incremental de los individuos en el eje X. El eje Y representa el rendimiento medio alcanzado por el algoritmo en el periodo de entrenamiento para una compañia (GVKEY: ID1356 en la base de datos del CRSP). GAwFO obtiene mejores

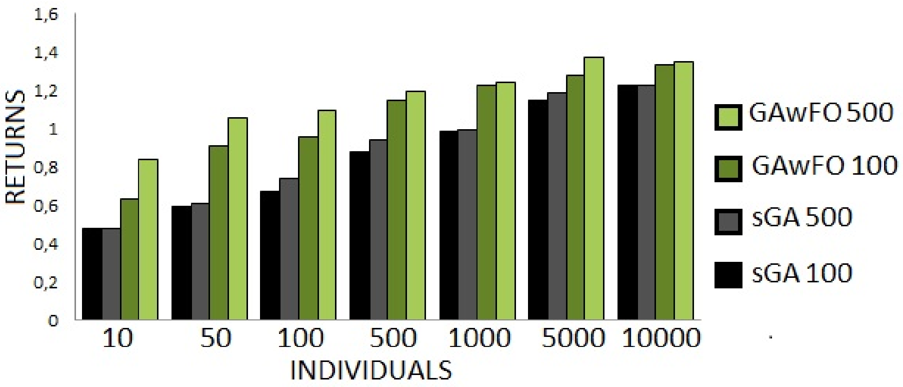


Figure 8.7: Rendimientos (eje Y) para el sGA y el GAwFO para 100/500 generaciones y un numero en incremento de individuos (eje X)

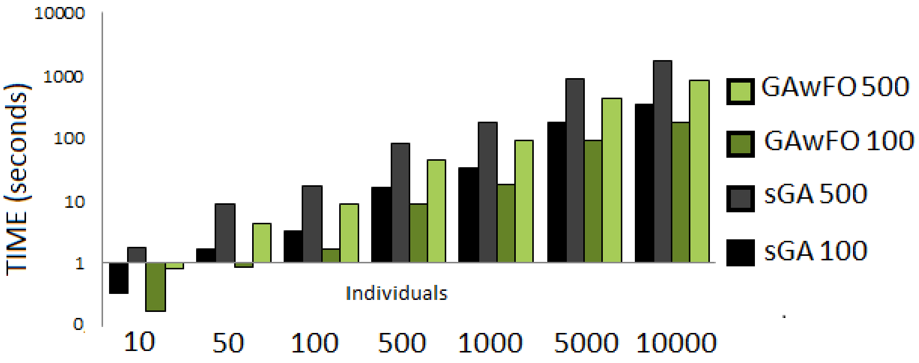


Figure 8.8: Tiempos de ejecución (eje Y) para el sGA y el GAwFO para 100/500 generaciones y un numero en incremento de individuos (eje X)

resultados que el sGA para todas las configuraciones. También podemos observar una mejor convergencia del GAwFO ya que el valor final de aptitud aumenta entre las generaciones 100 y 500 generaciones, lo que indica una preservación de la diversidad a lo largo de las generaciones. Además hemos medido los tiempos en estos experimentos dando lugar a la Figura 8.8, la cual presenta ejecuciones de 100/500 generaciones con un número incremental de los individuos en el eje X, y los tiempos de ejecución en el eje Y. La figura muestra que aparte de los beneficiosos resultados en cuanto a rendimiento financiero, el algoritmo también ofrece ventajas en cuanto a tiempo computacional se refiere. Este hecho es debido a la reducción en el número de evaluaciones de aptitud, debido a que estas se reducen a $n/2$

logrando así una aceleración aproximada de x2 al comparar GA y GAwFO.

Utilizando este mismo SAC proponemos técnicas de optimización novedosas en relación a uno de los problemas más característicos de estos sistemas, el tiempo de ejecución. A continuación presentamos la paralelización del sistema de comercio automatizado mediante dos técnicas de computación paralela, computación distribuida y procesamiento gráfico.

Ambas arquitecturas presentan aceleraciones elevadas, alcanzando los 50x y 256x respectivamente. Estápas posteriores presentan un cambio de metodologia de optimización, algoritmos genéticos por evolución gramatical, que nos permite comparar ambas estrategias, e implementar características más avanzadas como reglas más complejas o la auto-generación de nuevos indicadores técnicos. Testearemos con datos financieros recientes varios sistemas de comercio basados en diferentes funciones de evaluación, incluyendo una innovadora versión multi-objetivo, que nos permitirán analizar las ventajas de cada función de evaluación. Finalmente, describimos y testeamos la metodología del sistema de comercio automatizado basado en una doble capa de gramáticas evolutivas y que combina un análisis técnico, fundamental y macroeconómico en un análisis top-down híbrido. Los resultados obtenidos muestran altos rendimientos y bajo riesgo en pérdidas respecto otras aproximaciones.

8.5 Paralelización mediante un Grid de computación

En esta sección proponemos una implementación bajo la plataforma Boinc (Berkeley Open Infractucture for Network Computing). Boinc es un sistema de software libre que se desarrolló con la principal intención de tener una capacidad masiva de computación gracias a la interconexión de ordenadores a través de Internet, ya sea de manera local o global. Los proyectos vinculados con Boinc tienen una alta exigencia de computación, como podemos ver en SETI (Search for ExtraTerrestrial Intelligence)(http://setiathome.ssl.berkeley.edu/), donde según las estadísticas oficiales del proyecto (http://boincstats.com/) hay mas de tres millones de computadoras personales participando en el programa y que en conjunto tienen una potencia de procesamiento de 654.7 TeraFLOPS in 2014 (la capacidad de calculo total de Boinc alcanza 7 702 teraFLOPS). El coste total del proyecto cuatro años después de su fundación no excedió los 500000 dólares. En comparación y con similar potencia de calculo, solo el coste inicial de la computadora más potente de 2011 (proyecto RIKEN http://www.top500.org/lists/2013/11/) rebasó los 7.5 millones de dólares.

La enorme potencia de cálculo de Boinc, sin duda alguna, recae en los usuarios voluntarios (también llamados cilentes). La arquitectura general de Boinc es un modelo cliente-servidor. En un flujo normal de trabajo se comienza con el envío de unidades de trabajo desde el servidor a los equipos clientes. Más tarde, cuando los clientes han completado el trabajo, estos informan del trabajo al servidor, y finalmente, todos los resultados se procesan y se recomponen en el servidor. Boinc fue creado para aprovechar los ciclos perdidos de una unidad de procesamiento, es decir, aquellos ciclos que permanecen libres de tareas en la ejecución del procesador. Boinc es soportado por varios sistemas operativos (Unix, Windows o Mac) y por varias unidades de ejecución, es decir, maneras de ejecutar los programas, en la CPU o GPU. De esta manera, se permite interconectar un conjunto de ordenadores voluntarios. Las personas interesadas en ayudar a la ciencia y colaborar con esta, se pueden unir a estos proyectos sin esfuerzo, sin ralentizaciones y sin coste alguno para el propietario del ordenador. Además otros muchos proyectos dedican en específico ordenadores para la ejecución de Boinc, lo que provoca un aumento de rendimiento para los proyectos.

8.5.1 Modificaciones en el código y tareas de paralelización

Con el objetivo final de ejecutar el SAC en esta plataforma de paralelización se han realizado las siguientes tareas:

- Modular el programa original en sub-módulos, de forma que el ajuste del SAC se independiza dependiendo del periodo de ajuste y de la empresa a estudiar.

- Debido a la enorme cantidad de datos a estudiar, debemos desarrollar scripts que pre-procesen y post-procesen los datos de entrada y salida (resultados).

- Construir los ejecutables corelacionados a el SAC que serán enviados a los clientes Boinc (uno por sistema operativo).

- Crear un script para activar las librerías MCR (Matlab Component Runtime) antes de ejecutar el SAC en el cliente.

- Implementar un nuevo script que cargue los datos, ejecute el programa y almacene los resultados.

- Modificar el código fuente del software intermediario Wrapper para ajustarse a nuestros propósitos.

- Configurar la plataforma Boinc (servidor) mediante los siguientes pasos:

  - Crear las estructuras de proyecto y aplicaciones de Boinc
  - Establecer las preferencias de comunicación y metodología de trabajo del servidor y cliente (archivos *project.xml, config.xml and job.xml*).
  - Desarrollar una interfaz web para manejar y promocionar el proyecto.
  - Implementar un script responsable de generar todas las tareas.
  - Implementar scripts (entrada y salida) que generan plantillas para quelo clientes de Boinc reconozcan los datos de entradas y resultados de las tareas.

8.5.2 Resultados Experimentales: Tiempos de Ejecución en el Grid

Los resultados experimentales ha sido obtenidos en un grid perteneciente al CES Felipe II *(Universidad de Informática de Aranjuez, Madrid, España)*, donde se utilizó un servidor independiente, varios laboratorios de ordenadores y algunos voluntarios formando un grid con una capacidad total de computación de 225 gigaFLOPS y 433 gigaMIPS (http//:falua.cesfelipesegundo.com).

La figura 8.9 muestra los datos obtenidos de un conjunto de ejecuciones donde el número de individuos crece de forma iterativa. Esta serie de ejecuciones se han llevado a cabo con

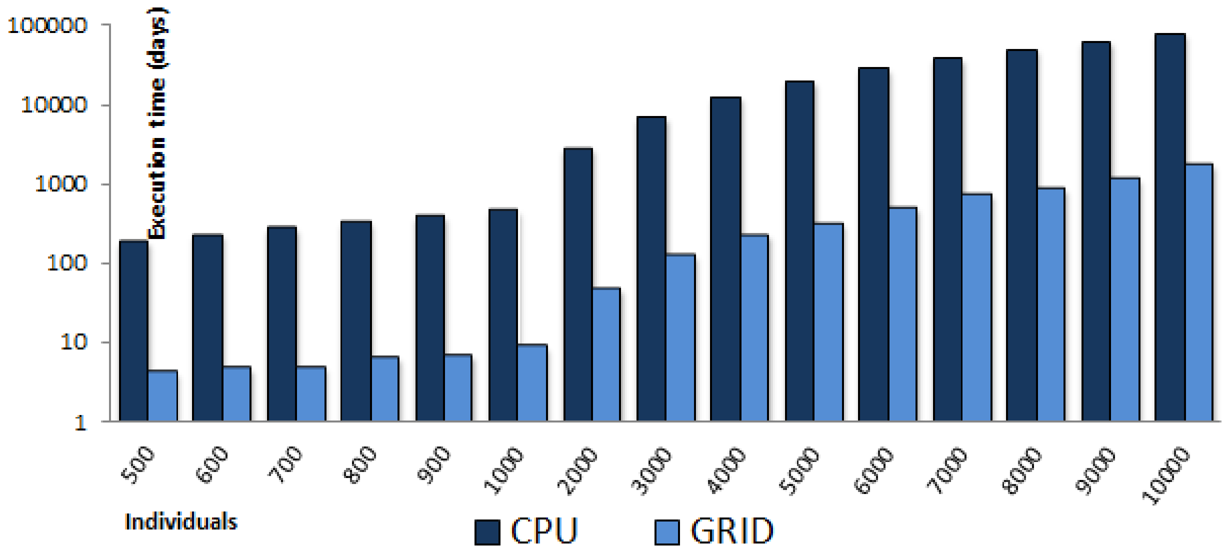


Figure 8.9: Tiempos de ejecucion en el grid (eje Y) para 500 generaciones y un numero de individuos creciente (eje X)

500 generaciones. El número de individuos está representado en el eje de coordenadas y el eje de abscisas muestra el tiempo de ejecución para el programa representado en una escala logarítmica de base 10. Desde el principio, la aplicación implementada en el grid proporciona tiempos de ejecución beneficiosos. La figura muestra tiempos muy ineficientes para la CPU independiente, como por ejemplo en las primeras ejecuciones, donde los tiempos de la CPU para 500 individuos son superiores a 100 días (184,9 días) y para el grid son aproximadamente de una semana (4,11 días). Cientos de días de ejecución es un tiempo demasiado elevado, más aún si pensamos que la ejecución sólo depende de una máquina con una ejecución sin puntos de control. Las aceleraciones alcanzadas son de hasta de x50, manteniéndose altas durante todos los experimentos.

8.6 Paralelización mediante tarjetas gráficas

Esta sección presenta una nueva paralelización del SAC basado en AG mediante el uso de tarjetas gráficas, conocida como GPGPU (General-Purpose Computing on Graphics Processing Units). En este capítulo se implementa un enfoque centrado en una paralelización interna del programa, proporcionándonos una ventaja en tiempos de cálculo y permitiéndonos aumentar el numero de individuos sin graves perdidas de rendimiento

Los hilos de ejecución múltiples en los procesadores de propósito general es una técnica común que se utiliza para aprovechar al máximo los recursos disponibles donde el procesador en colaboración con el sistema operativo procesa instrucciones de dos o más hilos simultáneamente. El paralelismo y la ejecución en hilos es esencial en el diseño de las GPUs actuales ya que realizan tareas altamente paralelas, como procesar cada vértice o fragmento de un gráfico, lo cual implica repetir constantemente las mismas operaciones con datos diferentes. Por este motivo, en lugar de dedicar gran cantidad de transistores para la memoria caché y para el flujo de control como la CPU, la GPU utiliza la mayor parte del área del chip para las unidades aritmético-lógicas (ALUS) que permiten procesar los accesos a memoria y cambios de hilo mucho más rápido. A pesar de la alta potencia de cálculo, las GPUs tienen límites relacionados con el ancho de banda de memoria debido al alto consumo en el intercambio de información. Los lenguajes de programación de GPU solían ser un lenguajes de bajo nivel que utilizaban APIs para transformar datos de imágenes, o paran transformar cualquier tipo de metodología de procesamiento de imágenes. CUDA nació como un lenguaje de propósito general para el desarrollo de software paralelo. CUDA representó un nuevo paso en el desarrollo de aplicaciones basadas en GPUs permitiendo a los programadores utilizar una versión de *C* en lugar de una API de gráficos. A pesar de las facilidades que se ofrecen a los programadores de CUDA, es un entorno muy complejo para las personas que rara vez desarrollan software. En la literatura podemos encontrar varias aproximaciones paralelizando AG mediante GPGPUs ([74]; [62]; [88]; [8]; [82] ; [116]), la mayor parte de estos trabajos se basan en la implementación mediante CUDA (`http://www.nvidia.com/cuda.html`) proporcionando información detallada sobre cómo configurar parámetros para obtener una implementación eficiente. Estos trabajos muestran que realizar una implementación requiere un buen nivel de conocimiento tanto de de arquitectura de computadores como de software de programación. Nuestra propuesta, el software *Jacket*, pretende ofrecer una herramienta adaptable para los inversores sin conocimientos especiales sobre la arquitectura de computadores, aunque familiarizados con las herramientas de *Matlab*. *Jacket* es una solución de software de cálculo numérico desarrollado por *AccelerEyes* La elección de este software está motivada por su capacidad de manipular matrices fácilmente en la GPU y por el conjunto de funciones GPU que ofrece que facilitan la programación en comparación con cualquier otro lenguaje de programación. Un buen ejemplo es la generación de números aleatorios, mientras que en la mayoría de los idiomas de la GPU no son inmediatos y pueden generar problemas ([63]), en *Jacket* es tan simple como el uso de una sola función.

8.6.1 Tareas de paralelización

Todo programa diseñado para CPU debe ser modificado para poder ser ejecutado en una GPU debido a las limitaciones existentes en la programación en GPU y las diferencias entre ambas unidades de proceso. El código normalmente requiere cambios drásticos, sin embargo Jacket muestra aqui su punto fuerte haciendo mucho mas accesible esta tarea. Los principales cambios que hemos realizado se enumeran a continuación:

- Pre-localizamos los datos en la memoria de la tarjeta gráfica con funciones tales como *gzeros* o *grand* (esta última función simplifica y resuelve uno de los problemas más conocidos en la progrmación de de GPGPUs, los números aleatorios).
- Cuando tiene que transferir los datos leídos por la CPU a la GPU o viceversa se realiza un intercambio de los tipos de datos.
- El resto de los cambios están relacionados con las incompatibilidades que *Jacket* presenta al transformar código de CPU. Los cambios más significativos se encuentran en las partes del código que se incluyen dentro de un bucle *gfor* (bucle idéntico a un *for* estandar pero donde las iteraciones del bucle se ejecutan paralelamente en la GPU.
  - La generación de numeros aleatorios mediante la función *grand* comando no es compatible por lo que el código fuente ha asumido ligeras variaciones en la estructura.
  - Los acumuladores deben ser cambiados por otras estructuras debido a su incompatibilidad con su ejecución en paralelo.
  - Las ramificaciones no están permitidas (el comando *if*) por tanto hemos implementado alternativas a este.
  - Modificaciones en segmentos del código que contienen funciones como *break*, *return*, que no son compatibles y deben ser reemplazados por sus códigos equivalentes.

8.6.2 Resultados experimentales: Tiempos de Ejecución con GPGPU

Los resultados experimentales presentados en esta sección se basan en una serie de pruebas ejecutadas en tanto la CPU y la GPU. Estas pruebas consisten en una serie de mediciones de tiempo de computación en ambas unidades de procesamiento. Debido a la naturaleza estocástica de AG, todas las pruebas experimentales han sido ejecutadas 30 veces. Lo experimentos se hn ejecutado sobre tres CPUs diferentes (P4, i7860 y SU4100) y dos GPUs (460GTX y 570GTX) que forman un variado espectro en cuanto a rendimiento se refiere. El P4 es la CPU más antigua y sólo tiene capacidad para ejecutar un hilo, el procesador i7 -860 es un moderno procesador con capacidad para ejecutar hasta 8 hilos simultáneamente y, por último la CPU SU4100 es una arquitectura intermedia con capacidad para procesar dos hilos diferentes al mismo tiempo. Las GPUs 460GTX y 570GTX son dispositivos modernos de una gama utilizada para el entretenimiento y con precios de entre 100 y 200 euros, respectivamente.

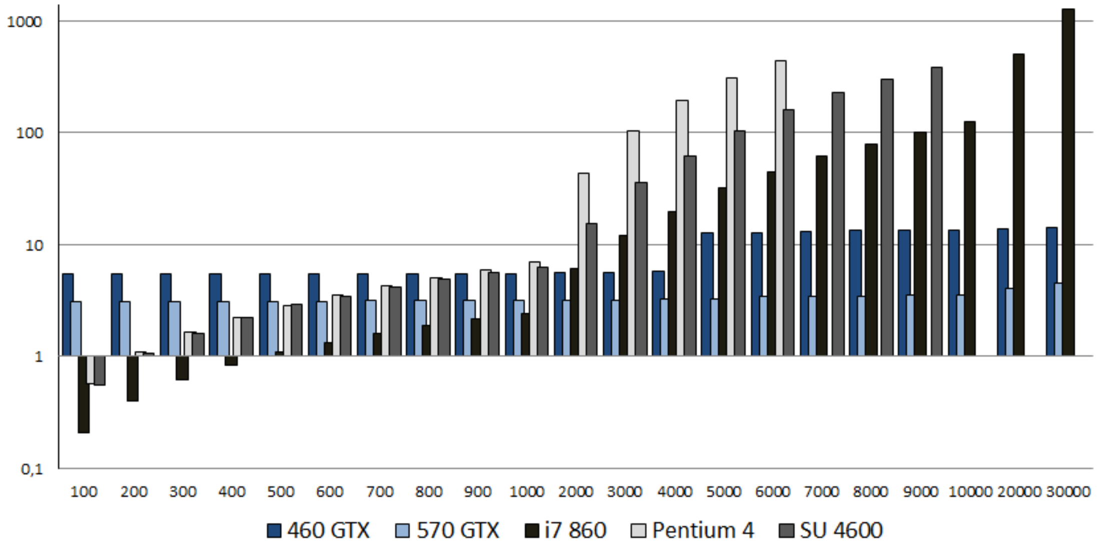


Figure 8.10: Tiempos de ejecución (eje-Y) para 500 generaciones y diferentes número de individuos (eje-X)

La Figura 8.10 muestra los tiempos de ejecución del SAC obtenidos (eje-Y) en una serie de experimentos donde los individuos se incrementan iterativamente. Los diferentes colores indican las unidades de proceso que ejecutan el SAC, un total de tres CPUs y dos GPUs. El P-4, SU-4100 y i7-860 no son capaces de terminar su ejecución en 7000, 10.000 y 40.000 individuos respectivamente. El tiempo de ejecución está intrínsecamente relacionado con el número de individuos y el número de generaciones de una ejecución particular, por lo tanto, el análisis de la figura 8.10 muestra los puntos de inflexión en los que las GPUs comienzan a obtener mejores resultados que las CPUs evaluadas. La figura muestra que las GPUs son más lentas que las CPUs en las primeras iteraciones ya que las mejoras proporcionadas por la paralelización no son suficientes para superar la velocidad a la que la CPU ejecuta los datos, según aumenta el número de individuos, se reducen las brechas de tiempo entre CPUs y GPUs. Los resultados máximos (GTX570) de aceleración que nos proporcionan los experimentos anteriores son:

- CPU Pentium 4: Aceleración de 128x en 6000 individuos.
- CPU SU 4100: Aceleración cerca de los 100x con 9000 individuos.
- CPU i7 860: Aceleración aproximada de 256x con 30000 individuos

8.7 Operando en el Mercado de Valores: Gamáticas Evolutivas y Funciones de Aptitud

En esta sección se presenta un cambio de metodología de optimización, algoritmos genéticos por evolución gramatical, que nos permite comparar ambas estrategias y posteriormente implementar características más avanzadas como reglas más complejas o la auto-generación de nuevos indicadores técnicos. Esta sección sienta las bases para el desarrollo de el SAC híbrido completo en la Sección 8.8 capaz de realizar análisis de las empresas más rentables en amplias las regiones del mundo.

El SAC basado en GE está gestionado por seis indicadores técnicos que nos proporcionan señales de compra, venta o neutrales. Los indicadores se han seleccionado por su utilidad en el mundo profesional y académico de las finanzas(MA crossover, MACD, VPCI, Support & Resistances, RSI divergence & convergence, RSI overbought & oversold [22]). Sin embargo, gracias a la flexibilidad de las gramáticas, no estamos limitados a seis indicadores clásicos. La GE además de optimizar los parámetros relacionados con cada indicador, además permite la construcción de nuevas versiones de los indicadores inicialmente implementados. Hay un infinidad de indicadores técnicos y hoy en día nuevos indicadores técnicos son creados. La mayoría de ellos son modificaciones de otros indicadores, o combinaciones de los mismos. Además, no hay una fórmula perfecta para seleccionar y usarlos. Por lo tanto, hemos implementado cuatro medias móviles diferentes: la simple, la ponderada, la exponencial [22] y la Hull [52], para permitir que nuestro sistema de comercio combine cualquiera de los indicadores utilizados y genere nuevas versiones de estos usando diferentes versiones de medias móviles.

En línea a la anterior característica, se ha implementado la posibilidad de aplicar un conjunto diferente de indicadores para el mismo periodo. La gramática permite seleccionar estrategias de ramificación, es decir, elegir diferentes estrategias en función del comportamiento de las inversiones realizadas. Así, cuando una estrategia no funciona correctamente en un período determinado, el SAC cambia su estrategia de inversión. Por tanto, el SAC puede ajustarse a cambios repentinos de los mercados. Además esta característica permite al SAC optimizar los períodos de entrenamiento del sistema. Suponiendo X el periodo fijo de entrenamiento del sistema (10 años en todos nuestros experimentos) e Y el nuevo periodo de entrenamiento del sistema, cuando el SAC esta entrenándose y logra un alto rendimiento en Y, el sistema tiende a converger a estrategias no rentables para X, por lo tanto, el sistema se optimiza en base a Y, dejando de tener en cuenta el período de optimización fijo.

Por último, nos centrarnos sobre la función de aptitud de nuestro SAC. En la literatura encontramos autores que claman que la falta de rendimiento de el AR para la evaluación de inversiones [86]. Esto se debe a que el AR evita un factor determinante en la evaluación de las inversiones, el riesgo. Aunque el propio sistema de comercio utiliza indicadores técnicos que de algún modo ayudan a medir el riesgo (por ejemplo Support & Resistances), los períodos de alta volatilidad exigen una mejor evaluación de los riesgos. Por tanto introducimos dos nuevas funciones en el SAC: El Indice de Sharpe (SI) [105] y el indice de correlación entre la inversión perfecta y la evaluada, en ingles *Coefficient between the Equity Curve of a strategy and the Perfect Profit* (CECCP) [86].

$$IS = \frac{AR - FR}{\sigma^{(R_i - R_f)/|(R_i - R_f)|}} \tag{8.8}$$

$$CECPP_i = \sum_{i=1}^{n} \frac{(X_i - \overline{X})(Y_i - \overline{Y})}{((n-1)(\sigma_X)(\sigma_Y))} \tag{8.9}$$

Donde:

- AR es el rendimiento acumulado
- FR rendimiento libre de riesgo
- σ representa la desviación estándar
- X es la inversión perfecta
- Y es la inversión evaluada

8.7.1 Resultados Experimentales: Gramáticas Evolutivas y Funciones de Aptitud

Con el objetivo de ofrecer experimetos más acordes a la situación actual y aprovechando el SAC de la Sección , actualizamos nuestras bases de datos, con datos recientes (2001-2013) para verificar su comportamiento en un entorno adverso como la actual recesión económica.

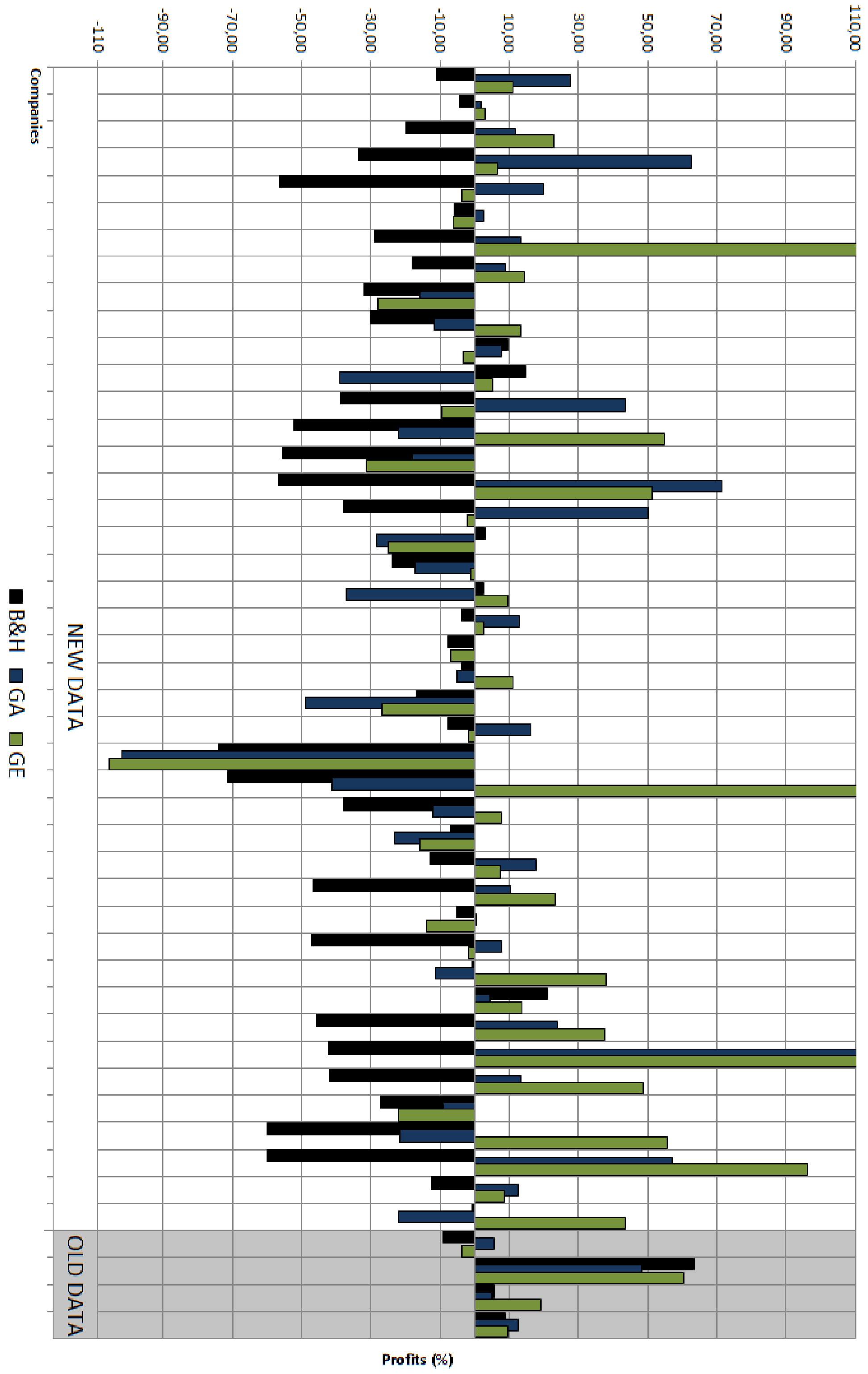


Figure 8.11: Rendimientos de 43 empresas españolas durante el 2012 (recent data) and 4 compañías del S&P500 en 2004 (old data)

En el primer experimento de esta sección se analizan los resultados proporcionados por dos SAC en un período de recesión económica y en uno de los países más afectados por esta: España. Para comparar justamente los dos SAC solo hemos utilizado la función de aptitud AR función 8.7 y ambos sistemas se ajustan para utilizar el mismo número de indicadores. La Figura 8.11 muestra los resultados de inversión en 43 empresas españolas. El eje vertical representa la rentabilidades obtenidas y el eje horizontal muestra las diferentes empresas y estrategias testeadas. La rentabilidad media obtenida para la estrategia de B&H es de -23,63%, un 5,89% para SAC basado en GA y, finalmente, un 21,08% para SAC basado en GE, además, hemos que la media de operaciones con rendimientos positivos son 8, 28 y 29 respectivamente.

Tras enfrentar las metodologías en uno de los países más afectados por la recesión económica centramos nuestros experimentos en el SAC basado en GE. Cambiamos nuestra muestra de empresas, con el objetivo de obtener una muestra más diversa, por 36 empresas pertenecientes a Alemania, Reino Unido, España y Francia. La Figura 8.12 muestra rendimientos de una serie de inversiones en 36 empresas en 2012, realizando el proceso de optimización de las reglas de inversión en los 10 años anteriores. El eje Y muestra los rendimientos obtenidos por los SAC. El eje X representa las diferentes funciones de aptitud y las empresas que han contribuido al experimento. Cada empresa cuenta con 3 barras, cada una relacionada con una función aptitud: AR, SI y CECPP. Los rendimientos medios proporcionados por el sistema son de 10.94% para el SI, 40.79% para el AR y 20.32% para el CECPP. El SI a pesar de conseguir los resultados más bajos es la estrategia con el menor número de operaciones con rendimientos negativos, un total de 12 inversiones no rentables que alcanzan un total de -26.68% (suma de las pérdidas). El AR obtiene 16 operaciones negativas con una rentabilidad total de -274%. Finalmente el CECPP muestra el peor comportamiento debido a que los rendimientos negativos alcanzan el -332 % con 18 inversiones fallidas, y como hemos señalado el rendimiento promedio no es muy alto superando ligeramente el 20%.

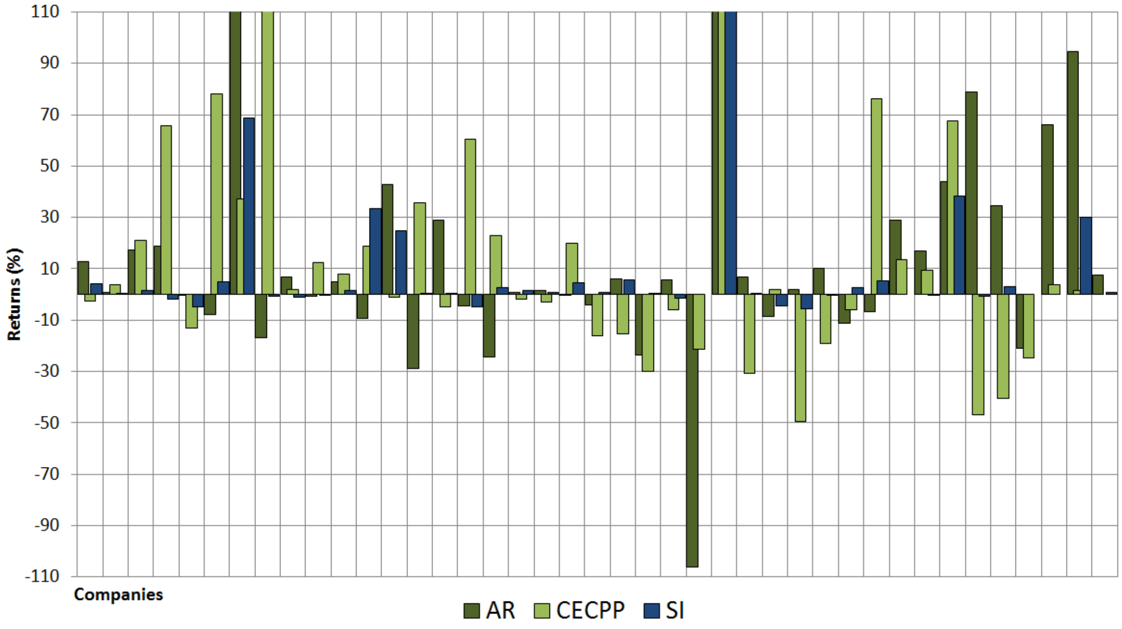


Figure 8.12: Rendimientos de 36 compañías europeas en 2012 de diferentes funciones de aptitud del SAC basado en GE

La conclusión más simple es, en primer lugar, no utilizar el CECPP. En segundo lugar, utilizar el RA si diversificamos las inversiones en un gran conjunto de empresas o disponemos de conocimientos expertos sobre la cartera analizada, y por último, utilizar SI cuando la cartera este compuesta por pocas empresas o se selecciona sin ningún criterio. Sin embargo, un estudio detallado de la figura muestra que, si bien en términos generales el IS y RA proporcionan una mayor confianza, el CECPP logra buenos resultados en muchos casos en que las demás funciones muestran resultados negativos. De esta forma el CECPP puede ofrecer algunas características capaces de obtener rendimientos donde el SI y el RA no son capaces de hacerlo. Esta observación nos anima a implementar un sistema que trata de captar las ventajas de las diversas funciones aptitud con el fin de mejorar los rendimientos finales.

8.7.2 Operando en el Mercado de Valores: Gamáticas Evolutivas y algoritmos multi-objetivo

Los SAC pueden evaluar gran diversidad de factores y objetivos con el fin de optimizar series de inversiones. Debido a la complejidad del problema, diferentes entornos financieros pueden requerir diferentes factores y objetivos. El SAC basado en GE evalúa multitud de posibles combinaciones de estos factores, sin embargo, nos guiamos por una función de aptitud única. Un enfoque combinado podría proporcionar soluciones que aporten mejores rendimientos. Por tanto, incluimos en el SAC una nueva funcionalidad capaz de proporcionar un compromiso óptimo entre los diferentes objetivos, es decir una algoritmo de optimización multi-objetivo. Otros trabajos han utilizado la optimizaciónes multi-objetivo para inversiones. Por ejemplo, en [13] los autores claman altos beneficios con una versión multi-objetivo de un SAC basado en el AR y SI. Más recientemente, Bodas en [11] y [107] describe un GA multiobjetivo donde se optimizan los parámetros de varios indicadores técnicos mientras recibe un flujo de datos continuo.

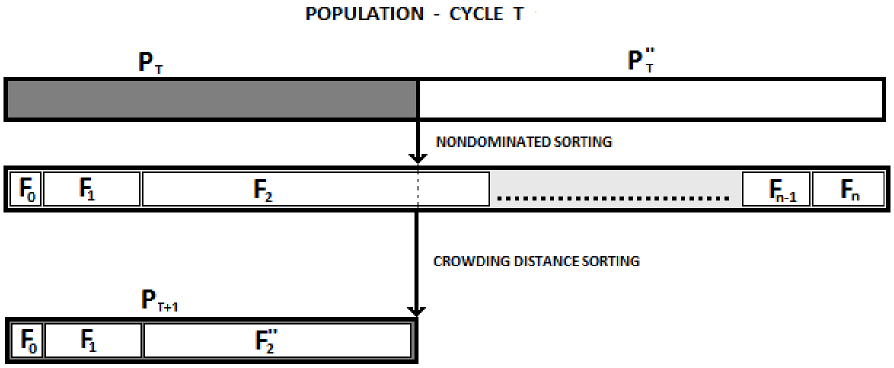


Figure 8.13: Flujo de el algoritmo multiobjetivo NSGAII.

En esta sección implementamos una versión basada en GE del algoritmo multi-objetivo conocido como *Non-dominated Sorting Genetic Algorithm* [31] (NSGA -II) . Hemos elegido NSGA -II por ser un algoritmo que ha mostrado su eficacia en multiples trabajos y que funciona con cualquier número de objetivos. Por tanto, evaluamos y ordenamos por dominancia la población obteniendo un conjunto de soluciones no dominadas como mejores individuos. El flujo del NSGA -II se muestra en la Figura 8.13. En primer lugar, se genera una población combinada de padres P_t y descendencia P_t''. Seguidamente, la población se clasifica en frentes no dominados F_n (F_0 contiene las mejores soluciones). A continuación, se trasladan por orden los individuos de los frentes a la nueva P_{t+1} hasta que quedan llenos. Si una frente queda a medias, clasificamos según su distancia de *crowding* [31] (en orden descendente) eligiendo el número necesario de individuos para llenar la nueva población. A la población generada P_{t+1} se le aplican los operadores de selección, cruce y mutación para crear una nueva población. Una vez el algoritmo llega a este punto el proceso se repite hasta alcanzar el número de generaciones máximo, donde escogemos el frente F_0 como solución.

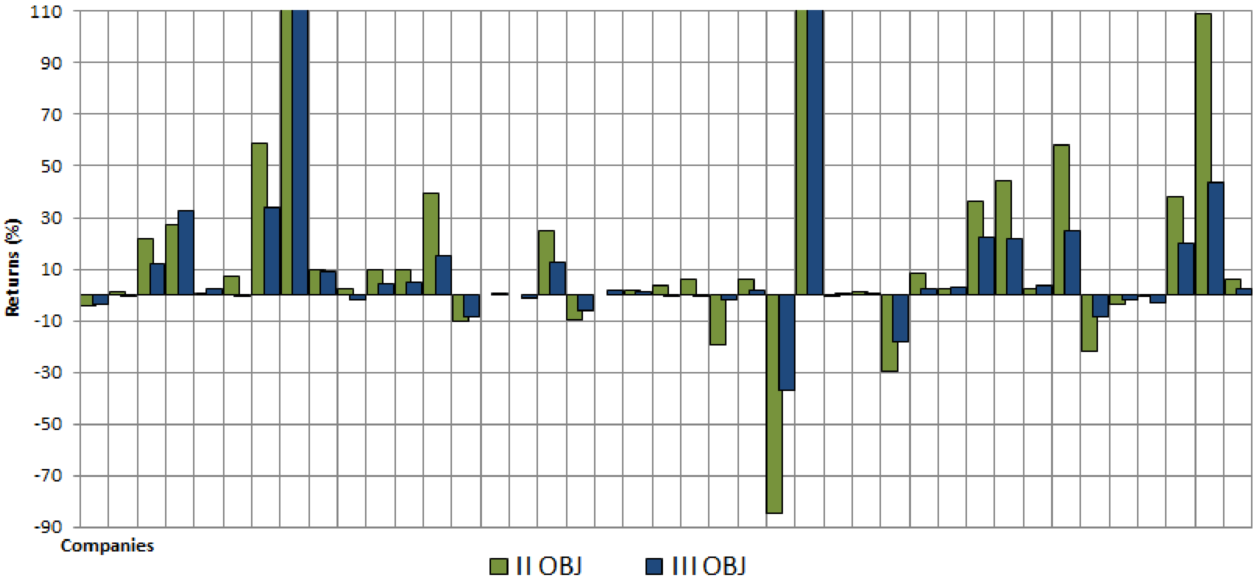


Figure 8.14: Rendimientos para el SAC multi-objetivo en 36 compañías europeas.

Por lo que respecta a nuestro conocimiento, el sistema basado en GE y NSGA II es el primer sistema que utiliza un enfoque multi-objetivo y una metodología de GE conjuntamente. La Figura 8.14 presenta los rendimiento de dos sistemas multi-objetivo (eje Y) en las 36 empresas de experimentos anteriores (eje X). Por un lado se utiliza un enfoque multi-objetivo basado en todas las funciones de aptitud utilizadas hasta ahora (3 objetivos) y, por otra parte, seleccionamos dos de las funciones de aptitud para construir otro enfoque con 2 objetivos, en este caso seleccionamos el AR y CECPP por ser las funciones con resultados más variados del experimento anterior y con el fin de tomar ventaja de las principales características de ambas funciones. El enfoque de 3 objetivos alcanza una rentabilidad media del 11,07% y la versión de 2 objetivos del 23,31%. La primera está ligeramente por encima del SI (función de aptitud independiente), pero con 15 operaciones negativas con un total de un -92%, por tanto, sin mejorar ninguna de las funciones independientes. Sin embargo, la versión de 2 objetivos alcanza la media más baja de operaciónes de perdida, con un total de 10 y alcanzado la segunda posición en cuanto a los rendimientos medios.

8.8 Operando en el Mercado: Análisis Macroeconómico, Fundamental y Técnico

En esta sección se incluye la contribución final de la tesis mediante el desarrollo de un SAC capaz de seleccionar empresas de diferentes países y mercados financieros. Existen diferentes formas de analizar los mercados, el análisis fundamental de empresas, el análisis macroeconómico y el análisis técnico. Inversores profesionales a menudo usan técnicas mixtas para invertir en las bolsas, sin embargo esta característica no se ha transmitido a los sistemas automáticos de inversión en bolsa.

En la Seción 8.3 analizamos y detectamos patrones en el comportamiento de los precios de varios sectores industriales, señalando que las compañías que componen lo sectores industriales se mueven en tendencias similares. Estas tendencias pueden se explotadas con el objetivo de escoger las carteras de empresas más beneficiosas. Nosotros incluimos este análisis en busca de aprovechar estas tendencias y combinar las mejores características de cada análisis. Hoy en día la mayoría de los SAC que se encuentran en la literatura se basan en indicadores técnicos, aunque en menor medida se pueden encontrar SAC basados âĂŃâĂŃen reglas fundamentales. En los últimos años se encuentran incluso diversos SAC usando un análisis híbrido que combina ratios fundamentales con los indicadores técnicos. En esta sección presentamos un novedoso enfoque basado en el análisis fundamental y técnico de la empresa y un análisis macroeconómico (incluida la industria) de entorno de esta. Nuestra propuesta aplica un análisis financiero top-down, es decir tomando decisiones de lo general a lo particular. El análisis híbrido comprenden cuatro puntos. En primer lugar un análisis económico para seleccionar los países europeos mejor valorados para invertir. En segundo lugar un análisis industrial para seleccionar industrias específicas y formar una cartera de empresas. En tercer lugar un análisis fundamental de la empresas que hemos seleccionado anteriormente para poder filtrar las empresas mejor valuadas. Y en último lugar, un análisis técnico para optimizar el momento y tipo de las decisiones de inversión.

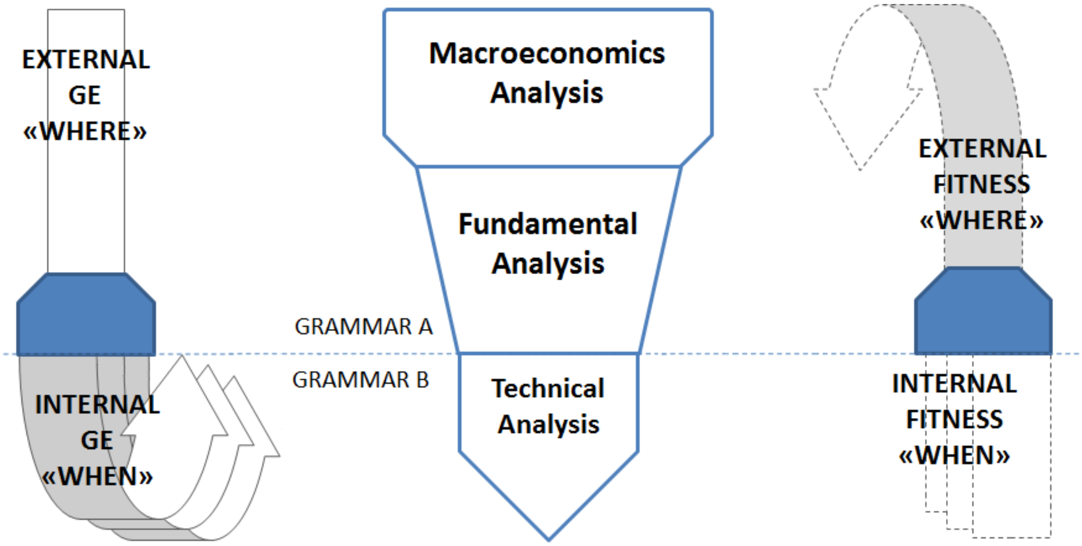


Figure 8.15: Metodología basada en una capadoble de GE

El SAC se basa en una metodología de una doble capa de GE. Como la Figura 8.15 representa, la metodología de doble capa de GE consiste en una GE (GE interna) trabajando como función de aptitud de otra GE (GE externa). Por tanto, la GE interna (el SAC explicado y probado en las secciones anteriores) proporciona un valor para la cartera (conjunto de empresas) evaluando cada una de las empresas que lo componen. De esta forma, la GE externa evoluciona los individuos buscando una cartera con las más empresas prometedoras, proporcionada por la gramática externa y evaluada por la GE interna.

8.8.1 Resultado Experimentales: Análisis Macroeconómico, Fundamental y Técnico

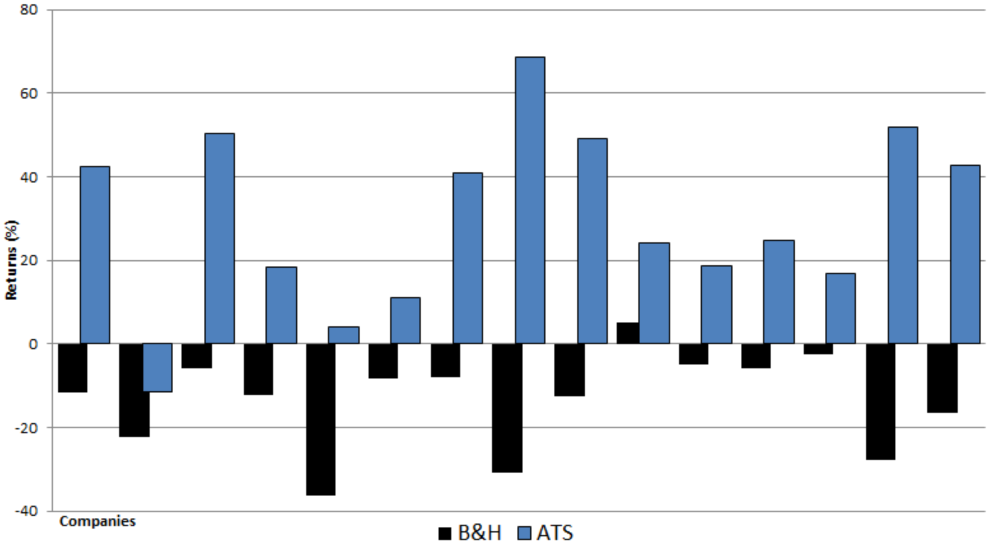


Figure 8.16: Rendimientos de 15 ejecuciones en 2013 por el SAC mediante un análisis fundamental, técnico y macroeconómico. Rendimientos medios de aproximadamente un 30%

La Figura 8.16 muestra los rendimientos en 2013 de 15 ejecuciones del SAC descrito en esta sección. El SAC ha construido las reglas de inversión guiadas por el SI con datos entre 2003 y 2012 (tanto las reglas de construcción de la cartera de empresas como las reglas para las decisiones de inversión). Los rendimients son comparados para las estrategias de B&H del mismo período. Por un lado la Figura 8.16 presenta altos rendimientos, alcanzando una rentabilidad media del 30.14% y una sola cartera con rentabilidades negativas, por otro lado, las mismas carteras en la estrategia B&H obtienen una rentabilidad media del -13,35% con 13 carteras de rentabilidades negativas.

8.9 Conclusiones

En esta tesis, se ha profundizado en los modelos de comportamiento modelando series temporales de los precios históricos siguiendo paseos aleatorios por la GBM que nos muestran cómo las series son indistinguibles a simple vista. Sin embargo, gracias una novedosa metodología Clustering, encontramos patrones entre conjuntos de series de precios modelados e históricos que nos permiten diferenciar completamente el origen de las series temporales. Por lo tanto, presentamos resultados de divergencia entre los precios siguiendo caminos aleatorios y los precios de los mercados de valores. Esta conclusión apoya el desarrollo de un sistema de comercio automatizado (SAC) capaz de analizar gran cantidad de precios históricos y usarlos como fuente de información para predecir futuros movimientos del mercado.

El método de Clustering presentado nos ha permitido buscar patrones entre los sectores industriales que componen los mercados de valores. Se han realizado varios experimentos que presentan similitudes entre conjuntos de series temporales de precios históricos pertenecientes a empresas que desarrollan actividades similares. Por tanto, esta tesis señala que los sector industriales de los mercados de valores están relacionados en alguna manera y se ven afectados de manera similar por el entorno económico. La idea de las tendencias similares entre empresas del mismo sector industrial hace más atractivo incluir un análisis macroeconómico con el fin de tomar ventaja de las tendecias de mercado.

Esta tesis ha introducido los SAC mediante el desarrollo de un SAC basado en Algoritmos Geneticos (GAs) implementando una innovadora versión de la metodología. Los resultados de los experimentos mostraron varias ventajas, primero evitando la convergencia prematura del algoritmo y, segundo, mejorando los resultados finales. El SAC fue testeado en 2004, donde logró un alto rendimiento y reducio el riesgo de pérdidas demostrando como las metodologías evolutivas se integran perfectamente con los sistemas de inversión.

Después de probar el enfoque basado en GA, se implemento un sistema de comercio automatizado basado en Evolución Gramatical (GE). Gracias a este cambio de metodología, el sistema presentado es capaz de generar reglas más complejas e incluso generar sus propios indicadores técnicos, consiguiendo así mejores rendimientos. Ademas este SAC nos ha servido para presentar una novedosa optimización multi-objetivo basada en GE y el algoritmo NSGA II que ha demostrado ser capaz de combinar algunas de las caraterísticas de las funciones de apttud empleadas.

Una vez que se evaluados todos los puntos mencionados en parrafos anteriores, se obtuvieron todas las piezas para desarrollar un SAC guiado por una metodología de una doble capa de GE y basada en un análisis completo del entrono financiero. El SAC combina novedosamente las características principales de un analisis macroeconómico, fundamental y técnico construyendo conjuntos de reglas mediante el análisis de períodos históricos. Estos conjuntos de reglas nos proporcionan soluciones formadas por una cartera de las empresas más prometedoras y un conjunto de operaciones de inversión rentables. El SAC fue testeado en el año 2013 consiguiendo unos rendimientos medios de aproximadamente un 40% y sola una cartera de empresas dando resultados de perdidas.

Por otra parte, a pesar de la dificultad las inversión con sistemas SAC intradiarios, se ha concluido que los SAC son capaces de responder en ventanas de tiempo adecuadas a través de la paralelización del sistema. En esta tesis mostró y comparó dos enfoques que combinan GAs y arquitecturas paralelas. Utilizamos una arquitectura grid computing para paralelizar el SAC a nivel de empresa y un entorno GPGPU para paralelizar el SAC a nivel de individuo, obteniendo altas aceleraciones que alcanzan valores de hasta x50 y x256 respectivamente.

Bibliography

[1] K. Adamu and S. Phelps. Modelling Financial Time Series using Grammatical Evolution. In *Proceedings of the Workshop on Advances in Machine Learning for Computational Finance*, London, UK, 2009.

[2] Kamal Adamu and Steve Phelps. Coevolution of technical trading rules for high frequency trading. In S. I. Ao, Len Gelman, David WL Hukins, Andrew Hunter, and A. M. Korsunsky, editors, *Proceedings of the World Congress on Engineering 2010 Vol I, WCE '10, June 30 - July 2, 2010, London, U.K.*, Lecture Notes in Engineering and Computer Science, pages 96–101. International Association of Engineers, Newswood Limited, 2010.

[3] F. Allen and R. Karjalainen. Using genetic algorithms to find technical trading rules. *Journal of Financial Economics*, 51(2):245 – 271, 1999.

[4] G. Appel. *Technical Analysis: Power Tools for Active Investors*. Financial Times Prentice Hall books. Financial Times/Prentice Hall, 2005.

[5] Bachelier. L. J B. The theory of speculation, 1900.

[6] Michael Bailey, Jon Oberheide, Jon Andersen, Z. Morley Mao, Farnam Jahanian, and Jose Nazario. Automated classification and analysis of internet malware. In *Proceedings of the 10th international conference on Recent advances in intrusion detection*, RAID'07, pages 178–197, Berlin, Heidelberg, 2007. Springer-Verlag.

[7] T.G. Bali, O. Demirtas, and H. Tehranian. Aggregate earnings, firm-level earnings, and expected stock returns. *JFQA*, 43(3):657–684, 2008.

[8] W. Banzhaf, S. Harding, W. B. Langdon, and G. Wilson. Accelerating genetic programming through graphics processing units. In *Genetic Programming Theory and Practice VI*, pages 1–19. Springer, 2009.

[9] D. Barthel, J.D. Hirst, J. Blazewicz, E.K. Burke, and N. Krasnogor. Procksi: a decision support system for protein (structure) comparison, knowledge, similarity and information. *BMC Bioinformatics*, 8:416, 2007.

[10] S. Basu. The investment performance of common stocks in relation to their price-earnings ratios: A test of the efficient market hypothesis. *Journal of Finance*, 32:663–682., 1977.

[11] D. Bodas-Sagi, F. Soltero, J. Hidalgo, P. Fernández, and F. Fernandez. A technique for the optimization of the parameters of technical indicators with multi-objective evolutionary algorithms. In *IEEE Congress on Evolutionary Computation*, pages 1–8, 2012.

[12] Anthony Brabazon and Michael O'Neill. Evolving technical trading rules for spot foreign-exchange markets using grammatical evolution. *Computational Management Science*, 1(3):311–327, October 2004.

[13] Antonio C. Briza and Prospero C. Naval, Jr. Stock trading system based on the multi-objective particle swarm optimization of technical indicators on end-of-day market data. *Appl. Soft Comput.*, 11(1):1191–1201, January 2011.

[14] Campbell and Yogo. Efficient tests of stock return predictability. *Journal of Financial Economics*, 81:27–60, 2006.

[15] John Y. Campbell and Robert J. Shiller. Valuation rations and the long run stock market outlook. *Journal of Portfolio Management*, 1998.

[16] Erick Cantú-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[17] L.K.C. Chan, Y. Hamao, and R. Lakonishok. Journal of finance. *Fundamentals and stock returns in Japan*, December:1739–1764, 1991.

[18] Pei-Chann Chang, Chen-Hao Liu, and Chin-Yuan Fan. Data clustering and fuzzy neural network for sales forecasting: A case study in printed circuit board industry. *Knowledge-Based Systems*, 22(5):344 – 355, 2009.

[19] G Charest. Dividend information, stock returns and market efficiency-ii. *Journal of Financial Economics*, 6:297–330, 1978.

[20] G Charest. Split information, stock returns and market efficiency-i. *Journal of Financial Economics*, 6:265–296, 1978.

[21] Xin Chen, Brent Francia, Ming Li, Brian Mckinnon, and Amit Seker. Shared information and program plagiarism detection. *IEEE TRANS. INFORM. TH*, 50:1545–1551, 2004.

[22] Robert W. Colby. *The Encyclopedia Of Technical Market Indicators*. McGraw-Hill, 2002.

[23] J. Colmenar, J. Risco-Martin, D. Atienza, O. Garnica, J. Hidalgo, and J. Lanchares. Improving reliability of embedded systems through dynamic memory manager optimization using grammatical evolution. In *Genetic and Evolutionary Computation Conference (GECCO) 2010*, pages 1227–1234, Portland (OR), EE.UU., 07/2010 2010. Association for Computing Machinery, Inc. (ACM), Association for Computing Machinery, Inc. (ACM).

[24] J.J. Collins Conor Ryan and Michael O'Neill. Grammatical evolution: Evolving programs for an arbitrary language. In *Lecture Notes in Computer Science 1391, Proceedings of the First European Workshop on Genetic Programming*, pages 83–95. Springer-Verlag, 1998.

[25] I. Contreras, Arnaldo N., N. Krasnogor, and J. Hidalgo. Blind problem instance classification : a novel approach to the universal similarity metric. *Memetic Computing*, 2014.

[26] Paul H. Cootner. The random character of stock market prices. *The Mit Press. Cambridge*, 36:322, 1964.

[27] Alfred. Cowles. Can stock market forecasters forecast? *Econometrica*, 1:309–324, 1933.

[28] Charles Darwin. *On the origin of Species*. John Murray, 1859.

[29] P. Dawyndt, H. De Meyer, and B. De Baets. The complete linkage clustering algorithm revisited. *Soft Computing*, 9(5):385–392, May 2005.

[30] Alfonso Ortega de la Puente, Rafael Sánchez Alfonso, and Manuel Alfonseca Moreno. Automatic composition of music by means of grammatical evolution. *SIGAPL APL Quote Quad*, 32(4):148–155, June 2002.

[31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182 –197, apr 2002.

[32] Ian Dempsey, Michael O'Neill, and Anthony Brabazon. Live trading with grammatical evolution. In *GECCO 2004 Workshop Proceedings*, Seattle, Washington, USA, 26-30 June 2004.

[33] Ian Dempsey, Michael O'Neill, and Anthony Brabazon. Adaptive trading with grammatical evolution. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 9137–9142, Vancouver, 6-21 July 2006. IEEE Press.

[34] M.A.H. Dempster and C.M. Jones. A real-time adaptive trading system using genetic programming. *Quantitative Finance*, 1(4):397–413, 2001.

[35] Buff Dormeier. Connection and affinity; between price and volume. *Technical Analysis of Stocks and Commodities*, 2007.

[36] Hussein Dourra and Pepe Siy. Investment using technical analysis and fuzzy logic. *Fuzzy Sets and Systems*, 127(2):221 – 240, 2002.

[37] G.P. Dwyer and R.W. Hafer. *The Stock Market: Bubbles, Volatility, and Chaos*. Springer, 2010.

[38] E. Fama and French. The cross-section of expected stock returns. *Journal of Finance*, 47(2):427–465., 1992.

[39] E. F. Fama. The behavior of stock-market prices. *Journal of Business*, 38(1):34–105, 1965.

[40] E. F. Fama and K. R. French. Business conditions and expected returns on stocks and bonds. *Journal of Financial Economics*, 25:23–49, 1989.

[41] Robert Fischer. *The New Fibonacci Trader: Tools and Strategies for Trading Success*. Wiley Trading, 2006.

[42] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley, 1966.

[43] Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

[44] Edward Gately. *Neural Networks for Financial Forecasting*. John Wiley & Sons, Inc., New York, NY, USA, 1995.

[45] M. Gheorghescu. An automated virus classification system. In *Virus Bulletin Conference*, pages pp. 294–300, 2005.

[46] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[47] D.E. Goldberg and University of Alabama. Clearinghouse for Genetic Algorithms. *Optimal Initial Population Size for Binary-coded Genetic Algorithms.* TCGA report. Clearinghouse for Genetic Algorithms, Department of Engineering Mechanics, University of Alabama, 1985.

[48] Walter Torous Harrison Hong and Rossen Valkanov. Do industries lead stock markets? *Journal of Financial Economics*, (83):367–396, 2007.

[49] Randy L. Haupt and Sue Ellen Haupt. *Practical Genetic Algorithms.* John Wiley & Sons, Inc., New York, NY, USA, 1998.

[50] J. H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI, USA, 1975.

[51] Jonatan Hugosson, Erik Hemberg, Anthony Brabazon, and Michael O'Neill. Genotype representations in grammatical evolution. *Appl. Soft Comput.*, 10(1):36–43, January 2010.

[52] Alan Hull. *Active Investing.* 2010.

[53] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[54] Y. Jiang and L. Núñez. Efficient market hypothesis or adaptive market hypothesis? a test with the combination of technical and fundamental analysis. In *Proceedings of the 15th International Conference. Computing in Economics and Finance.* University of Technology, Sydney, Australia., July 2009.

[55] Stephen Johnson. Hierarchical clustering schemes. *Psychometrika*, 32:241–254, 1967. 10.1007/BF02289588.

[56] Maurice Kendall. The analysis of economic time series-part I: Prices. *Journal of the Royal Statistical Society*, 116(1):11–34, 1953.

[57] Lars Kerstner. *Quantitative Trading Strategies: Harnessing the Power of Quantitative Techniques to Create a Winning Trading Program.* McGraw-Hill Trader's Edge Series, 2003.

[58] I.Y. Kim and O.L. de Weck. Variable chromosome length genetic algorithm for progressive refinement in topology optimization. *Structural and Multidisciplinary Optimization*, 29(6):445–456, 2005.

[59] Andrey N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems in Information Transmission*, 1:1–7, 1965.

[60] John R. Koza, James P. Rice, and Jonathan Roughgarden. Evolution of food foraging strategies for the caribbean anolis lizard using genetic programming. *Adaptive Behavior*, 1(2):171–199, 1992.

[61] Natalio Krasnogor and David A. Pelta. Measuring the similarity of protein structures by means of the universal similarity metric. *Bioinformatics*, 20(7):1015–1021, 2004.

[62] F. Krüger, O. Maitre, S. Jiménez, A. Baumes, and P. Collet. Speedups between 70 and 120 for a generic local search (memetic) algorithm on a single gpgpu chip. In *EvoApplications (1)*, pages 501–511, 2010.

[63] W. B. Langdon. A fast high quality pseudo random number generator for nvidia cuda. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, GECCO'09, pages 2511–2514, New York, USA, 2009. ACM.

[64] Benjamin Lewin. *Genes VII.* 1999.

[65] Szu-Yin Lin, Chi-Hua Chen, and Chi-Chun Lo. Currency exchange rates prediction based on linear regression analysis using cloud computing. *International Journal of Grid and Distributed Computing*, 6(2), 2013.

[66] Oguzhan Ozbas Lior Menzly. Market segmentation and cross-predictability of returns. *The Journal of Finance*, 65(4):1555–1580, 2010.

[67] A. W. Lo and A. C. MacKinlay. Stock market prices do not follow random walks: evidence from a simple specification test. *Review of Financial Studies*, 1(1):41–66, January 1988.

[68] A.A.W.C. Lo and A.C. MacKinlay. *A Non Random Walk Down Wall Street.* Princeton University Press, 1999.

[69] Andrew W. Lo. The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *The Journal of Portfolio Management*, 2004.

[70] Andrew W. Lo and Dmitry V. Repin. The psychophysiology of real-time financial risk processing. *J. Cognitive Neuroscience*, 14(3):323–339, April 2002.

[71] Dome Lohpetch and David Corne. Discovering effective technical trading rules with genetic programming: Towards robustly outperforming buy-and-hold. In *NaBIC*, pages 439–444, 2009.

[72] Dome Lohpetch and David Corne. Multiobjective algorithms for financial trading: Multiobjective out-trades single-objective. In *IEEE Congress on Evolutionary Computation*, pages 192–199, 2011.

[73] Osbome. M. F. M. Brownian motion in the stock market. *Operations Research*, 7:145–173, 1959.

[74] O. Maitre, L. Baumes, N. Lachiche, A. Corma, and P. Collet. Coarse grain parallelization of evolutionary algorithms on gpgpu cards with easea. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, GECCO '09, pages 1403–1410, New York, NY, USA, 2009. ACM.

[75] B.G. Malkiel. *A Random Walk Down Wall Street [By] Burton G. Malkiel.* 1973.

[76] RBC Golbal Asset Management. U.s. market structure: Is this what we asked for. Technical report, RBC, 2012.

[77] R. Messe and Rogoff K. Empirical exchange rate models of the seventies, do they fit out-of-samle? *Journal of international Economics*, 14:3–24, 1983.

[78] Melanie Mitchell. *An Introduction to Genetic Algorithms.* MIT Press, Cambridge, MA, USA, 1996.

[79] M.A Mittermayer. Forecasting intraday stock price trends with text mining techniques. In *System Sciences*, 2004.

[80] Jason H. Moore and Lance W. Hahn. Petri net modeling of high-order genetic systems using grammatical evolution. *Biosystems*, 72(12):177 – 186, 2003. Computational Intelligence in Bioinformatics.

[81] Mark. Moskowitz, Tobias J.; Grinblatt. Do industries explain momentum? *Journal of Finance*, 54(4):1249–1290, 1999.

[82] A. Munawar, M. Wahib, M. Munetomo, and K. Akama. Hybrid of genetic algorithm and local search to solve max-sat problem using nvidia cuda framework. *Genetic Programming and Evolvable Machines*, 10:391–415, December 2009.

[83] Laura Núñez. Trading systems designed by genetic algorithms. *Managerial Finance*, 28:87–106, 2002.

[84] William J. O'Neil. *How to Make Money in Stocks.* McGraw-Hill, 1988.

[85] M. O'Neill and C. Ryan. *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language.* Kluwer Academic Publishers, 2003.

[86] Robert Pardo. *The Evaluation and Optimization of Trading Strategies.* Wiley Trading, 2008.

[87] Cheol-Ho Park and Scott H. Irwin. What do we know about the profitability of technical analysis? *Journal of Economic Surveys*, 21(4):786–826, 2007.

[88] Petr Pospichal. Gpu-based acceleration of the genetic algorithm. In *Proceedings of the 16th Conference Student EEICT 2010 Volume 5*, pages 234–238. Faculty of Information Technology BUT, 2010.

[89] Sander Pronk, Per Larsson, Iman Pouya, Gregory Bowman, Imran Haque, Kyle Beauchamp, Berk Hess, Vijay Pande, Peter Kasson, and Erik Lindahl. Copernicus : A new paradigm for parallel adaptive molecular dynamics. In *Proceedings of 2011 SC - International Conference for High Performance Computing, Networking, Storage and Analysis*, page 60, 2011. QC 20120223.

[90] P.Siepmann, C.P. Martin, I. Vancea, P.J. Moriarty, and N. Krasnogor. A genetic algorithm approach to probing the evolution of self-organised nanostructured systems. *Nano Letters*, 7(7):1985–1990, 2007. (for the latest version please check the official journal site).

[91] Jasdeep Anand Abhishek Rao D Tripati, Mandia. Risk-return analysis of sectorial portfolios of stocks. *Economics, Management and Financial Markets*, pages 108–126, 2012.

[92] David E. Rapach and Mark E. Valuation ratios and long-horizon stock price predictability. *Journal of Applied Econometrics*, 20:327–344, 2005.

[93] M. Reinganum. Selecting superior securities charlottesville. the tesearch foundation of the institute of chartered financial analysts. Technical report, The Tesearch foundation of the institute of Chartered Financial Analysts., 1988.

[94] H. Roberts. Stock market patterns and financial analysis: methodological suggestions. *Journal of Finance*, 14:1–10, 1959.

[95] Richard Roll. Industrial structure and the comparative behavior of international stock market indexes. *Journal of Finance*, (47):3–41, 1992.

[96] Richard N. Rosett. Review: The random character of stock market prices. *Econometrica*, 36(1):191–192, 1968.

[97] Ronald de Wolf Rudi Cilibrasi, Paul Vitanyi. Algorithmic clustering of music based on string compression. *Computer Music Journal*, Vol. 28(4):Pages 49–67, 2004.

[98] Conor Ryan, Michael O'Neill, and JJ Collins. Grammatical evolution: Solving trigonometric identities. In *In Proceedings of Mendel '98: 4th International Conference on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks and Rough Sets*, pages 111–119, 1998.

[99] Philip Saks and Dietmar G. Maringer. Evolutionary money management. volume 5484 of *Lecture Notes in Computer Science*, pages 162–171. Springer, 2009.

[100] Richard Schabacker. *Stock Market Theory and Practice*. Financial Time Press, 1930.

[101] Richard Schabacker. *Technical Analysis and Stock Market Profits: A Course in Forecasting*. Financial Times Press, 1932.

[102] Richard Schabacker. *Stock Market Profits*. Financial Times Press, 1934.

[103] Robert A. Schwartz, John Aidan Byrne, and Antoinette Colaninno. *Electronic vs. floor based trading*. Springer, 2006.

[104] H. P. Schwefel. Evolutionsstrategie und numerische optimierung. *PhD Thesis*, 1975.

[105] William F. Sharpe. Mutual Fund Performance. *The Journal of Business*, 39(1):119–138, 1966.

[106] RJ Shiller. From efficient markets theory to behavioral finance. *Journal of Economic Perspectives*, 17:83–104, 2003.

[107] F. Soltero, D. Bodas-Sagi, J. Hidalgo P. Fernández, and F. Fernandez. Optimization of technical indicators in real time with multiobjective evolutionary algorithms. In *GECCO (Companion)*, pages 1535–1536, 2012.

[108] Gilbert Sywerda. Uniform crossover in genetic algorithms. In *Proceedings of the third international conference on Genetic algorithms*, pages 2–9, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[109] Stephen J. Taylor. Tests of the random walk hypothesis against a price-trend hypothesis. *Journal of Financial & Quantitative Analysis*, 17:37–61, 1982.

[110] G. Terrazas, P. Siepman, G. Kendal, and N. Krasnogor. An evolutionary methodology for the automated design of cellular automaton-based complex systems. *Journal of Cellular Automata*, 2(1):77–102, 2007. (for the latest version of this paper please refer to the journal website).

[111] Robert R. Trippi and Efraim Turban, editors. *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance*. McGraw-Hill, Inc., New York, NY, USA, 1992.

[112] Robert Choate Tryon. *Cluster Analysis*. Oxford, 1939.

[113] J.W. Wilder. *New concepts in technical trading systems*. Trend Research, 1978.

[114] B. K. Wong, V. S. Lai, and J. Lam. A bibliography of neural network business applications research: 1994-1998. *Computers and Operations Research*, 27(11-12):1045–1076, 2000.

[115] R.A.J. Woolley, J. Stirling, A. Radocea, N. Krasnogor, and P. Moriarty. Automated probe microscopy via evolutionary optimization at the atomic scale. *Applied Physics Letters*, 98(25):253104, 2011.

[116] Sifa Zhang and Zhenming He. Implementation of parallel genetic algorithm based on cuda. In *Proceedings of the 4th International Symposium on Advances in Computation and Intelligence*, ISICA'09, pages 24–30, Berlin, Heidelberg, 2009. Springer-Verlag.