SPRINGER BRIEFS IN APPLIED SCIENCES AND TECHNOLOGY · COMPUTATIONAL INTELLIGENCE

António M. S. B. S. Gorgulho
Rui F. M. F. Neves · Nuno C. G. Horta

Intelligent Financial Portfolio Composition Based on Evolutionary Computation Strategies

Springer

SpringerBriefs in Applied Sciences and Technology

Computational Intelligence

Series Editor

Janusz Kacprzyk

For further volumes:
http://www.springer.com/series/10618

António M. S. B. S. Gorgulho
Rui F. M. F. Neves · Nuno C. G. Horta

Intelligent Financial Portfolio Composition Based on Evolutionary Computation Strategies

Springer

António M. S. B. S. Gorgulho
Instituto de Telecomunicações
Instituto Superior Técnico
Lisbon
Portugal

Nuno C. G. Horta
Instituto de Telecomunicações
Instituto Superior Técnico
Lisbon
Portugal

Rui F. M. F. Neves
Instituto de Telecomunicações
Instituto Superior Técnico
Lisbon
Portugal

ISBN 978-3-642-32988-3 ISBN 978-3-642-32989-0 (eBook)
DOI 10.1007/978-3-642-32989-0
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012949455

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The management of financial portfolios or funds constitutes a widely known problem in financial markets which normally requires a rigorous analysis in order to select the most profitable assets. This subject is becoming popular among computer scientists who try to adapt known Intelligent Computation techniques to the market's domain. Among those intelligent methodologies, it is possible to highlight techniques such as Genetic Algorithms, Genetic Programming, Neural Networks, Simulated Annealing, and Tabu Search. The mentioned techniques can be applied to financial markets in a variety of ways; as to predict the future movement of a stock's price, or to optimize a collection of investment assets, such as a fund or a portfolio. This innovation is of special importance due to the high volume of securities (financial instruments) involved; normally, it is very hard to a simple investor optimize his profits without requiring the skills of financial market's specialists.

The goal of this work is to provide an application which tries to replace those specialists in order to help an investor or an investment company to achieve a significant profit on buying and selling (trading) financial instruments. In order to apply such procedures we need to believe that the historical data related to stocks and markets form appropriated indications about the market future performance. This premise constitutes the basis of Technical Analysis which simply tries to analyze the securities past performance in order to evaluate these investments at the present time. This philosophy relies on three bases: (1) the fact that market action discounts everything; (2) the fact that price moves in trends; and (3) that history tends to repeat itself. These considerations allow us, through the study of charts and financial data, recognizing which way the market is most likely to go. Despite the fact that technical analysis is becoming widely used, there are still some criticisms to this perception on market's evolution. For instance, Burton Malkiel stated that the "past movement or direction of the price of a stock, or overall market cannot be used to predict its future movement". His findings become popular, leading to a new investment theory called The Random Walk Theory where the author stipulates that if we cannot beat the market, then the best investment strategy we can apply is buy-and-hold in which an investor buys stocks and holds them for a long period of time, regardless of market fluctuations. For the technical community, this idea of purely random movements of prices is totally rejected, and more recent studies try to evidence their beliefs. Also, if we consider the price movement as unpredictable, how can we explain that price moves in trends? If we observe several stock charts considering a predefined period we can easily detect an uptrend or a downtrend.

The work presented in this book proposes a potential system, based on those techniques, in particular Genetic Algorithms, which aims to manage a financial portfolio by using technical analysis indicators (EMA, HMA, ROC, RSI, MACD, TSI, OBV). In order to validate the developed solution an extensive evaluation was performed, comparing the designed strategy against the market itself (DJI, S&P500) and several other investment methodologies, such as Buy & Hold, Momentum, and a purely random strategy. The time span (2003–2009) employed on the evaluation allowed the performance investigation under distinct market conditions, culminating with the most recent financial crash. The results are promising since the developed approach beats the remaining procedures during the crash. Also, to highlight the fact that this application is available to be used on a practical and realistic point of view since it is capable of considering real time data, and presenting a potential set of market assets to invest.

This book is organized in five chapters and three appendices.

Chapter 1 presents a brief description on the problematic addressed by this book, namely the management of financial portfolios using intelligent computation techniques. Additionally, the main goals for the work presented in this book, as well as, the document's structure are, also, highlighted in this chapter.

Chapter 2 addresses some of the fundamental concepts needed to understand the developed work. Further, a substantial part of the several methodologies applied to the portfolio problematic are analyzed and the problem related with portfolio theory and investment's analysis is discussed. Subsequently, the evolutionary techniques which can be used to solve this problem are focused. Finally, the connection between the presented financial domain and the evolutionary techniques is presented, through an extended analysis on the existing solutions.

Chapter 3 provides the description of the developed solution to approach the portfolio management problem. First, presents an overview on the application's architecture, followed by the delineation of the strategies employed, and a detailed characterization of the several modules within the system.

Chapter 4 describes the validation approach used to evaluate the defined system, in particular the employment of the Backtesting test strategy.

Chapter 5 summarizes the provided book and supplies the respective conclusion and future work.

The appendices provide the Markowitz's Model, a list of available applications for both portfolio management and trading, and, also, a description of the classification parameters.

António M. S. B. S. Gorgulho
Rui F. M. F. Neves
Nuno C. G. Horta

Contents

1 Introduction 1
1.1 Computational Finance 1
1.2 Work's Purpose 2
1.2.1 General Goals 2
1.2.2 Concrete Goals 3
1.3 Book's Structure 4
References 4

2 Related Work 5
2.1 Portfolio Theory 5
2.1.1 Diversification 6
2.1.2 Management 6
2.2 Market Analysis 7
2.2.1 Fundamental Analysis 7
2.2.2 Technical Analysis 8
2.2.3 Fundamental Versus Technical 9
2.3 Evolutionary Computation 10
2.3.1 Genetic Algorithms 10
2.3.2 Genetic Programming 12
2.4 Existing Solutions 13
2.4.1 Portfolio Optimization Theory 13
2.4.2 Solving Markowitz's Model 15
2.4.3 Metaheuristics Approaches to Portfolio Optimization . . . 16
2.4.4 Technical and Fundamental Analysis in Portfolio Management 22
2.4.5 Overview and Discussion 25
2.5 Conclusions 25
References 29

3 Solution's Architecture 31
3.1 Overall Architecture 31
3.2 Data Flow 33
3.3 Financial Data Processing Module 34
3.3.1 Implementation and Functionality 35
3.4 Technical Rules Module 36
3.4.1 Extensibility and Technical Rules Module Implementation 38
3.4.2 Exponential Moving Average 39
3.4.3 Hull Moving Average 39
3.4.4 Double Crossover 40
3.4.5 Rate of Change 41
3.4.6 Relative Strength Index 42
3.4.7 Moving Average Convergence Divergence 43
3.4.8 On Balance Volume 44
3.4.9 True Strength Index 46
3.5 Optimization Module 47
3.5.1 Chromosome Representation 47
3.5.2 Selection 48
3.5.3 Mutation 49
3.5.4 Crossover 49
3.5.5 Initial Generation 50
3.5.6 Constraints Handling 50
3.5.7 Evaluation Function 51
3.5.8 Optimization Module Implementation 51
3.6 Investment Simulator Module 52
3.6.1 Implementation and Functionality 52
3.7 Conclusions 54
References 54

4 System Validation 57
4.1 Performance Measures 57
4.1.1 Return on Investment 57
4.1.2 Sharpe Ratio 58
4.1.3 Sortino Ratio 59
4.2 Classification Parameters 59
4.3 Strategies Employed 60
4.4 Case Studies 61
4.4.1 Case Study I—DJI Between 2003 and 2009 61
4.4.2 Case study II—SP Between 2006 and 2009 65
4.5 Conclusions 67
References 68

5 Conclusions and Future Work 69
5.1 Conclusions 69
5.2 Future Work 70
5.2.1 Current Limitations 70
5.2.2 Possible Improvements 70
References 71

Appendices 73

Abbreviations

ANN	Artificial neural networks
B&H	Buy and hold
CLA	Critical line algorithm
EA	Evolutionary algorithm
EC	Evolutionary computation
EF	Efficient frontier
EMA	Exponential moving average
GA	Genetic algorithm
GP	Genetic programming
HMA	Hull moving average
IS	Investment simulator
MA	Memetic algorithm
MACD	Moving average convergence divergence
MO	Multi-objective
MOOP	Multi-objective optimization problem
OBV	On balance volume
PSO	Particle swarm optimization
QP	Quadratic programming
ROC	Rate of change
ROI	Return on investment
RSI	Relative strength index
SA	Simulated annealing
SMA	Simple moving average
SO	Single-objective
SVM	Support vector machines
TS	Tabu search
TSI	True strength index
WMA	Weighted moving average

Chapter 1
Introduction

Abstract This chapter presents a brief description on the problematic addressed by this book, namely the management of financial portfolios using intelligent computation techniques. Additionally, the main goals for the work presented in this book, as well as, the document's structure are, also, highlighted in this chapter.

1.1 Computational Finance

Nowadays, more than ever, with the quick increasing of technology and the significantly evolvement of financial markets, there is a constant need of helping investors to correctly apply their money, in order to achieve a significant profit.

This field is becoming popular among computer scientists, especially to Computational Intelligence specialists who try to combine elements of learning, evolution and adaptation in order to create intelligent software. In particular, subjects such as Neural Networks, Swarm Intelligence, Fuzzy Systems and Evolutionary Computation are becoming extremely notorious on the market's domain. The mentioned techniques can be applied to financial markets in a variety of ways; as to predict the future movement of a stock's price, or to optimize a collection of investment assets, such as a fund or a portfolio. This innovation is of special importance due to the high volume of securities (financial instruments) involved, normally, it is very hard to a simple investor to optimize his profits without requiring the skills of a financial market's specialists. The goal of this work is to provide a description of the development of an application which tries to replace those specialists in order to help an investor or an investment company to achieve a significant profit on buying and selling (trading) financial instruments. In order to apply such procedures one must assume that the historical data related to stocks and markets forms appropriated indications about the market future

A. M. S. B. S. Gorgulho et al., *Intelligent Financial Portfolio Composition Based on Evolutionary Computation Strategies*, SpringerBriefs in Computational Intelligence,
DOI: 10.1007/978-3-642-32989-0_1,

performance. This premise constitutes the basis of Technical Analysis which simply tries to analyze the securities past performance in order to evaluate these investments at the present time. This philosophy relies on three bases [1]; the fact that market action discounts everything, the fact that price moves in trends, and that history tends to repeat itself. These considerations allow us, through the study of charts and financial data, to recognize which way the market is most likely to go. Despite the fact that technical analysis is becoming widely used, there are still some criticisms to this perception on market's evolution. For instance, Burton Malkiel [2] stated that the "past movement or direction of the price of a stock, or overall market cannot be used to predict its future movement". His findings become popular, leading to a new investment theory called The Random Walk Theory where the author stipulates that if we cannot beat the market, then the best investment strategy we can apply is buy-and-hold in which an investor buys stocks and holds them for a long period of time, regardless of market fluctuations. For the technical community, this idea of purely random movements of prices is totally rejected, and more recent studies [3, 4] try to evidence their beliefs. For instance, in [3] the author demonstrated the validity of technical analysis using more than 70 technical indicators which showed that market movements can be predicted at a certain degree. Also, if we consider the price movement as unpredictable, how can we explain that price moves in trends? If we observe several stock charts considering a predefined period we can easily detect an uptrend or a downtrend.

1.2 Work's Purpose

As stated before, the goal of this work consists on describing the implementation of an application capable of automatically manage a financial fund or portfolio by using evolutionary intelligence techniques. From this work it is possible to extract several other goals, on what the reader can expect about the presented book.

1.2.1 General Goals

To achieve the intent proposed the following research was carried:

- Performing a rigorous investigation on several evolutionary computational techniques;
- Understanding the involved domain, in this case, it was necessary to develop a hard study on financial markets;
- Thoroughly comprehend how financial assets can be selected to compose a portfolio, by considering different methodologies.

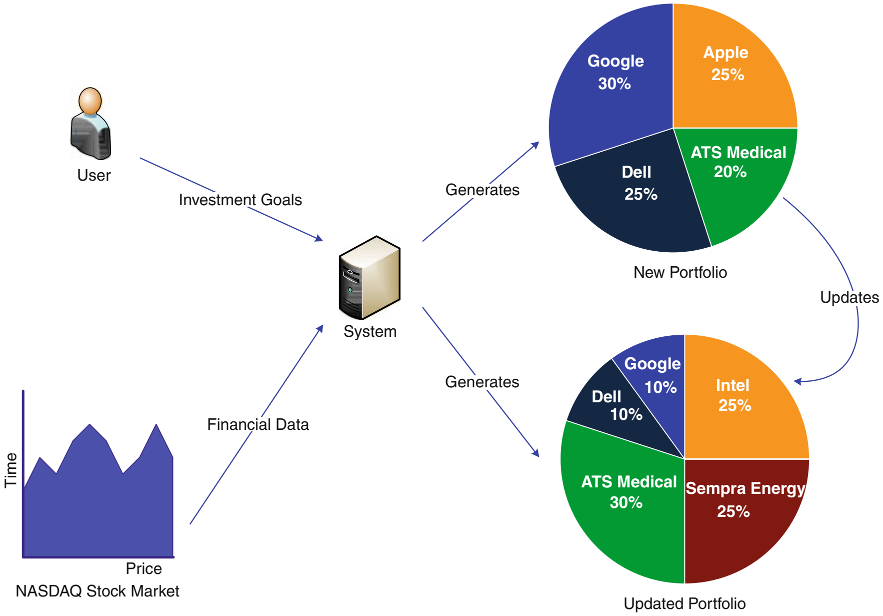


Fig. 1.1 An illustration of the system main purpose

1.2.2 Concrete Goals

From this work the reader can have an insight on:

- Understanding how evolutionary computational techniques such as Genetic Algorithms, Neural Networks, Genetic Programming, Tabu Search and Simulated Annealing can be applied to the problem of portfolio optimization. The reader will grasp the application of those techniques on picking the most attractive stocks on the market;
- Comprehend the process developed on this work on applying a Genetic Algorithm coupled with Technical Analysis rules to find the most interesting instruments on the market [5, 6]. The underlined architecture is well documented on this report;
- Perceive the evaluation process of such strategies. How can this solution be compared with other investment strategies and the market itself.

The diagram in Fig. 1.1 tries to express, very briefly, the behavior of the proposed system. As the reader can observe, in the figure above, the application asks two distinct inputs; the user parameters, his investment goals, and a set of financial data, in this case, the prices obtained by several stocks within the market through a specific period of time. Based on those configurations, the system will generate a financial portfolio, and subsequently, has the responsibility of updating it over time.

1.3 Book's Structure

This book is structured as following:

- Chapter 2 addresses the theory behind the developed work, namely the concepts of financial portfolio, portfolio management, technical analysis, and evolutionary computation. Also, in this chapter, it is given an extended overview about different methodologies which can be used to address the portfolio problematic;
- Chapter 3 illustrates the solution's architecture of the developed application;
- Chapter 4 proposes the validation procedure to evaluate the developed system by providing an exhaustive study on the solution's performance and robustness;
- Chapter 5 summarizes the provided report and supplies the respective conclusion and future work.

References

1. Murphy J (1999) Technical analysis of the financial markets: a comprehensive guide to trading methods and applications. Prentice Hall Press, New York
2. Burton M (1973) A random walk down wall street. W. W. Norton and Company, New York
3. Canegrati E (2008) A non-random walk down Canary Wharf. Dissertation, University of Munich
4. Irwin SH, Park CH (2007) What do we know about the profitability of technical analysis. J Econ Surveys 21:786–826
5. Gorgulho A, Neves R, Horta N (2009) Using GAs to balance technical indicators on stock picking for financial portfolio composition. In: Genetic and evolutionary computation conference (GECCO) Montreal, Canada, pp 2041–2046
6. Gorgulho A, Neves R, Horta N (2011) Applying a GA kernel on optimizing technical analysis rules for stock picking and portfolio composition. Expert Syst Appl 38:14072–14085

Chapter 2
Related Work

Abstract In this chapter some of the fundamental concepts necessary to understand the developed work are addressed, particularly the domain relative to financial markets. Further, a substantial part of the several methodologies applied to the portfolio problematic are analyzed; throughout the first two sections, the problem related with portfolio theory and investment's analysis is presented. Subsequently, the evolutionary techniques which can be used to solve this problem are focused. Finally, Sect. 2.4 presents the connection between the presented financial domain and the evolutionary techniques, through an extended analysis on the existing solutions.

2.1 Portfolio Theory

A financial portfolio [1] consists of a group of financial assets, also called securities or investments, such as stocks, bonds, futures, CFDs, or groups of these investment vehicles known as exchange-traded-funds (ETFs). In order to one construct a portfolio, it is capital to define investment objectives that should focus on a certain and accepted degree of risk, i.e. the chance of incurring in a loss.

The core of this work is related to *portfolio management* [1], the act of deciding which assets need to be included in the portfolio, how much capital should be allocated to each kind of security and when to remove a specific investment from the holding portfolio. During this process, it is required to take into account the investor's preferences since some investors are more willing to accept a specific degree of risk than others, hoping that way to achieve better returns.

A. M. S. B. S. Gorgulho et al., *Intelligent Financial Portfolio Composition Based on Evolutionary Computation Strategies*, SpringerBriefs in Computational Intelligence, DOI: 10.1007/978-3-642-32989-0_2, © The Author(s) 2013

2.1.1 Diversification

As it was explained in the paragraph above, one of the fundamental goals of any investor consists on reducing his portfolio's risk. The main technique used in finance to reduce this chance of losing capital, is called diversification [1]. Diversification means that the risk needs to be spread, mixing a variety of investment vehicles, in order to minimize the loss impact of one investment in the portfolio. To understand better this concept, it is important to distinguish between two forms of risk [2]:

- *Systematic risk* the risk inherent to a market segment or the entire market which cannot be removed through diversification. Wars and economic recessions constitute an example of this kind of risk;
- *Specific risk* also known as unsystematic risk, which corresponds to the risk related to a short number of assets. Company's strikes, accidents or specific news affecting one company can be caste as unsystematic risks, which can be easily surpassed recurring to diversification.

Independently of the diversification degree of a portfolio, it is fundamental to understand that the intrinsic risk can never be shrank down to zero since there is always a form of risk (systematic) which cannot be removed. However, using a risk-managing technique, such as diversification, the specific risk can be easily reduced.

This methodology can be accomplished with the help of strategies such as the following:

- Selection of different investment vehicles, such as stocks, bonds, futures or ETFs;
- Mixing assets from different industries, countries, and sectors.

Defined this concept, it is capital to understand that diversification cannot guarantee that a losing investment is avoided. However, it can prevent loss, reducing the impact of a specific investment in the overall portfolio.

2.1.2 Management

As it was already mentioned, the goal of this work is concentrated on the automatic management of a portfolio. It is important to notice that distinct forms of management can be applied [1]:

- *Passive Management* in which the investor concentrates his objective on tracking a market index. This is related to the idea that it is not possible to beat the market index, as stated by the Random Walk Theory [3]. More concretely, a passive strategy aims only at establishing a well-diversified portfolio without trying to find under or overvalued stocks;

- *Active Management* in which the main goal of the investor consists on outperforming an investment benchmark index, buying undervalued stocks and selling overvalued ones.

Although the differences between both forms of management seem quite clear, the question active versus passive is still widely debated. Most part of the mathematical formulations used to model the aspect of portfolio optimization problem, such as the Mean–Variance Model, proposed by Markowitz [4] and which is considered as the holy grail of portfolio management theory are classified as passive management. However, as stated by Beverly Goodman [5], "passive management strives to beat (and historically does beat) the overall market by proper asset allocation and cost management." According to him, most people wrongly think that diversification implies reducing risk while getting the market average returns. However, as stated in his article "they didn't give (economist) Harry Markowitz the Nobel Prize for coming up with a theory that generates average returns." For instance, Aranha and Hitoshi [6] proposed a portfolio optimization application based on the Markowitz's model which constantly beats the index over distinct periods.

What should be understood here is the fact that passive management also tries to beat an index as the active form. At first sight, the risk and transaction costs involved when using a passive strategy are not so high when compared to an active one. Probably, most of the published articles apply this kind of approach for that reason. However, it should be noticed that an active strategy, using technical indicators, can possibly guarantee us with higher profitability levels.

In this work, both solutions are examined when coupled with evolutionary computation techniques.

For more information on the Markowitz's model, the reader is referred to Appendix A.

2.2 Market Analysis

When defining a financial fund or portfolio our goal is to pick the best potential assets within the market in order to avoid losses and maximize our returns. There are several ways to perform a reasonable evaluation of the market and select potential profitable securities. Usually, investment analysts perform a fundamental or a technical analysis of the market. Notice that these strategies are not exclusive and both can be applied. However, a fundamental analyst tries to avoid the antagonist approach.

2.2.1 Fundamental Analysis

Fundamental Analysis [7] evaluates each security by measuring its intrinsic value through the study of overall economy, industrial conditions, and the financial situation of a specific company. When this intrinsic value is calculated it is compared against the current security's price. If this value is inferior to the market value, then the market is possibly overvalued and the investor should sell all the company's shares, otherwise the market is undervalued and it can be extracted a potential buy signal. Summarizing the previous definition; fundamental analysis tries to understand the factors which can possibly affect market behaviour. After a rigorous probe the investor should be able to answer the following questions, and subsequently take a decision:

- Is the company's income (from business activities) growing?
- Is the company actually generating profit?
- Is the company able to repay the assets owed?

2.2.2 Technical Analysis

At the other end of the rope there is Technical Analysis [7]. A technical analyst believes that market action, namely the volume of transactions and the securities prices include all the fundamentals that can possibly affect market's price; political, economical, or psychological. Following this premise there is only the need to study those factors in order to forecast market behaviour. The applied strategies on technical analysis normally embody a set of technical indicators which try to give us a future perspective of market development according to what is visible on price charts. A technical indicator consists in a formula that is normally applied to stock's prices and volumes. The resulting values are plotted and then analysed in order to offer us a perspective on price evolution. More specifically, a technical indicator tries to capture the behaviour and investment psychology in order to determine if a stock is under or overvalued.

In order to illustrate the behaviour of such approach, the technical analyst starts by applying a simple technical indicator as the Simple Moving Average (SMA). The SMA plots per each day, the average on prices observed during the last x days. Depending on the considered data, it is also possible to employ the indicator to weekly or monthly prices. The following picture illustrates the usage of a moving average with a duration period of 12 weeks when applied to Intel weekly prices. Notice that the blue line identifies the stock price and the red line corresponds to the SMA. Observe the smoothness on the SMA line, which allow us to easily perceive the market movements (Fig. 2.1).

Regarding an indicator such as the former one, a strategy for defining buying and selling signals can be formulated:

- *Entry Signal*. Price line crosses above the SMA line;
- *Exit Signal*. Price line crosses below the SMA line.

Based on entry/exit signals and other plot characteristics different rules can be defined which allow us to score the distinct stocks within the market and subsequently pick the best securities according to the indicators employed.

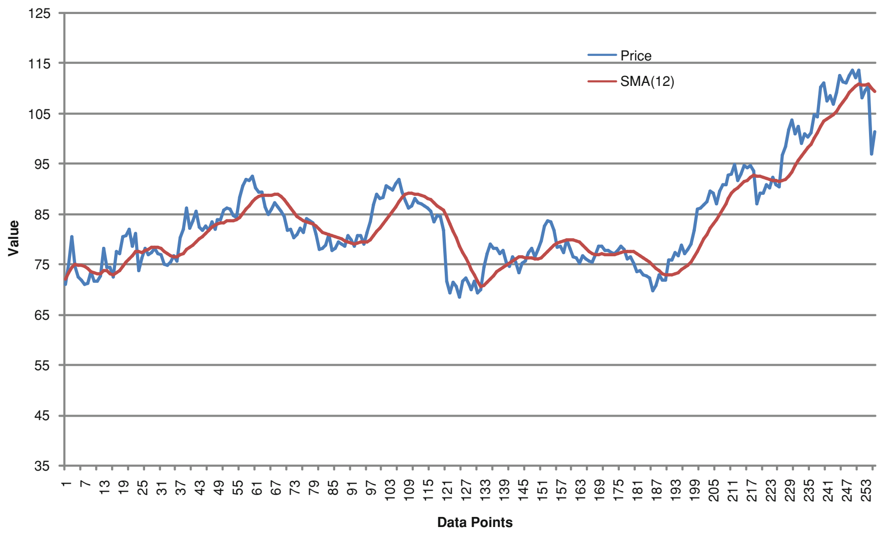


Fig. 2.1 SMA application

2.2.3 Fundamental Versus Technical

The big question which normally an inexperienced investor can formulate is; which kind of market analysis should he employ, a fundamental or a technical one?

The major drawback present on a fundamental analysis approach corresponds to the difficulty on obtaining such data and the fact that most part of this financial data is not reliable due to company's self interests. Besides, technical analysis already includes fundamental analysis because if a technical analyst believes that all factors that can possibly influence the price are already included on it, then they only need to be studied in order to evaluate the market and consequently forecast its development. Another major difference between both forms of market analysis is the time-horizon used when investing. The financial data used by a fundamental analyst is only released over long periods of time. Normally, each company announces its results following a quarterly basis, which is completely different from using daily or weekly data, such as the volume or price data employed on technical analysis.

Although both strategies seem to be on opposite sides they can coexist. Several analysts can couple fundamental data and technical analysis to provide an efficient evaluation of the market. For instance, first use fundamental analysis to pick potential profitable companies and then technical analysis for defining entry and exit signals.

2.3 Evolutionary Computation

One of the major concepts presented within this work is the subfield of artificial intelligence designated as Evolutionary Computation [8]. This methodology embodies the application of a procedure based on biological mechanisms of evolution which tries to progress iteratively to converge on an optimal solution for a combinatorial optimization problem. Normally, these evolutionary techniques involve metaheuristic optimization algorithms, i.e. algorithmic frameworks specialized in solving optimization problems. These metaheuristics, besides being based on biological evolution, can also have their groundwork on a naturally appearing phenomenon, such as Simulated Annealing (SA) and Tabu Search (TS). Since major part of this work is based on evolutionary procedures, further sub-sections will start by explaining some of these concepts, namely the Genetic Algorithms (GA) and Genetic Programming (GP).

2.3.1 Genetic Algorithms

A Genetic Algorithm corresponds [9] to a search technique used to find optimal or sub-optimal solutions to search problems. Its behaviour is inspired on evolutionary biology, by defining an initial set of random solutions, which is iteratively refined, until an optimal or a sub-optimal solution to the problem is encountered. The following diagram tries to express the behavioural process defined by the standard GA (Fig. 2.2):

As you can see from the above figure, the algorithm proceeds as following:

- The execution starts by generating an initial population, a set of potential solutions for the problem, randomly defined;
- Following, the initial population is evaluated by a fitness function, also designated as evaluation function. Based on the values previously calculated, the best individuals are selected for reproduction. A set of operators are applied to those individuals, in order to generate a more refined population;
- If a specific finish criterion is fulfilled, for instance, the best individual has the desired fitness value, the algorithm terminates, and the best individual, i.e. the best solution is returned, otherwise this new refined population is evaluated and the same process is applied.

2.3.1.1 Individual Representation

Depending on the target problem, the first step when defining a genetic algorithm consists on specifying the representation of each solution, also designated as individual or chromosome. Normally, a chromosome is represented by a set of variables, also known as genes, depending on the considered problem.

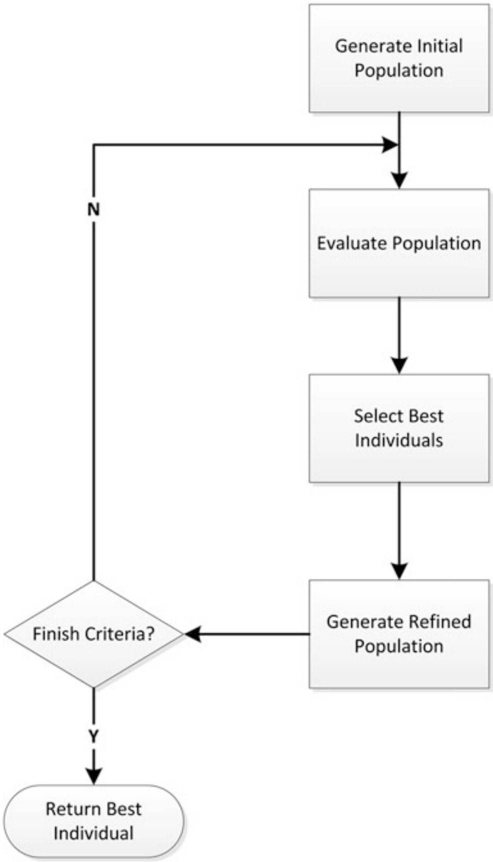


Fig. 2.2 GA general behaviour

2.3.1.2 Initial Generation

After specifying the underlined representation it is necessary to define the initial set of individuals. According to the standard genetic algorithm, a set of random individuals is initially created, which means that random values are assigned to the variables contained within each chromosome, in order to cover a vast area of the search space.

2.3.1.3 Selection

During each iteration of the algorithm it is fundamental to pick the best individuals, i.e. solutions with the best fitness according to the evaluation function, in order to guide the search more effectively. These individuals are chosen according to a specific procedure designated as Selection. The selection operator is responsible for selecting the chromosomes for reproduction. The fitter the chromosome, more likely it is that it will be reproduced. Several selection procedures are available [9]. On Chap. 3 further details will be given on selection procedures.

2.3.1.4 Offspring Generation

After the selection operation, the picked chromosomes are combined, and subsequently generate new individuals designated as offspring. The application of this procedure is fundamental to guide the search space. This operation is normally known as Crossover, it works as a reproduction function, it combines the characteristics of two individuals, the parents, and generates a new individual, or more than one, the offspring. Like the selection procedure, several crossover operators are also available [9].

Besides the crossover procedure, another function which is normally applied corresponds to the Mutation operator. The mutation procedure is fundamental within a GA in order to avoid the algorithm to concentrate on a specific search space, converging too quickly on a local maximum. A mutation operation normally corresponds to a random alteration on the genes of a specific or random chromosome. On current literature, there are several ideas on how to apply this procedure, depending on the considered chromosome representation and the meaning given to a specific parameter called Mutation Rate. The reader is referred to [10] for further details.

2.3.2 Genetic Programming

A Genetic Programming [9] procedure consists on applying a GA to write computer programs. The variables correspond to different program constructs and the algorithm tries to find the one which best achieves its goals. A simple way of viewing a genetic program can be defined as the following:

- Assume distinct numbers and several operators are available. Then the goal consists on determining the equation that best achieves a specific goal; for instance, return the maximum value as possible, given this set of operands and operators.

To solve a problem such as the presented one the reader could opt for a genetic programming where each solution or chromosome is represented as a tree structure (traditional representation for GP). Within the tree structure; a node represents an operator and each terminal node an operand. Figure 2.3 provides an example of the stipulated representation.

As can be observed from the previous figure, the tree is evaluated in a recursive manner and from that the following equation can be extracted:

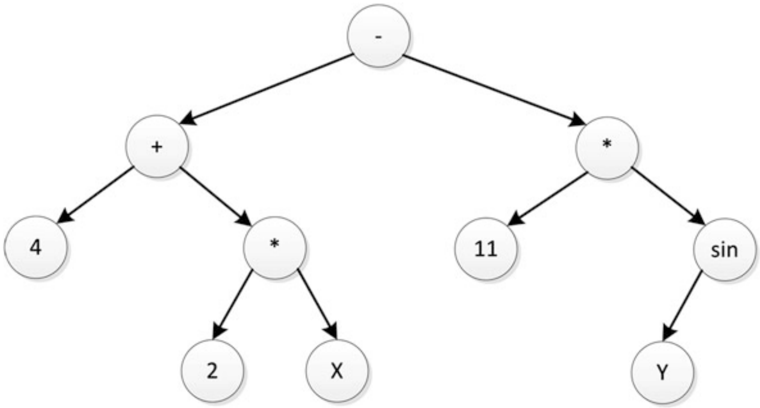


Fig. 2.3 Tree structure for GP

$$(4 + (2 * X)) - (11 * \sin(Y)) \tag{2.1}$$

Iteratively evolving the algorithm as defined under the previous section, by applying a set of genetic operators, it is possible to achieve the equation which best fits our purposes, returning the maximum value as possible, in this example.

2.4 Existing Solutions

Through this section a substantial part of the work developed in this domain is presented. The works here addressed use optimization techniques by evolving two different ways on handling this problem. The first one, given within Sects. 2.4.1, 2.4.2 and 2.4.3 consists on coupling optimization algorithms with mathematical models for portfolio optimization. The procedure's goal concentrates on splitting a fixed amount of capital between different securities, each one with a specific weight within the portfolio, and majorly maintaining a passive management approach. The second strategy, given under Sect. 2.4.4 involves the use of technical and fundamental analysis to define the portfolio composition, based on an active management approach.

2.4.1 Portfolio Optimization Theory

Through this section the main mathematical formulations used to model the portfolio optimization problem are addressed, more specifically, the principles employed to calculate the risk and return measures when coupling portfolio mathematical models with optimization techniques such as GAs.

2.4.1.1 Markowitz's Pioneer Work

The problem related to portfolio management suffered a major revolution during the fifties with Harry Markowitz [4]. The author is pioneer in the Modern Portfolio Theory (MTP) after analyzing the effects related with risk, correlation and diversification over the expected returns of investment portfolios.

After completing his study, Markowitz concluded that rational investors should diversify their investments, in order to reduce the respective risk and increase the expected returns. The author's assumption focus on the basis that for a well-diversified portfolio, the risk which is assumed as the average deviation from the mean, has a minor contribution to the overall portfolio risk. Instead, it is the difference (covariance) between individual investment's levels of risk that determines the global risk.

See appendix A for more details regarding the Markowitz's model.

2.4.1.2 Alternative Models

Although Markowitz's model is widely used to design the portfolio optimization problem, other models can also be considered. For instance, Black and Litterman [11] suggested a new formulation, the Black-Litterman model. In their work they propose means of estimating expected returns to achieve better-behaved portfolio models. The designed model is very similar to Markowitz's one, the main difference is concentrated on the calculation of the expected returns which generates portfolios considerably different when using the original model.

According to the authors their new design tries to rectify some of the flaws presented on Markowitz's one. They address the fact that the "expected returns are very difficult to estimate and that historical returns provide poor guides to future returns" when using the patriarch model. Also, one of the major drawbacks pointed to the original model is the time that is necessary to compute the covariance matrix from historical data and solving the resulting problem. With recent technology this problem is not an issue anymore.

Although this new model could seem a better approach according to the authors and more recent studies [12], its implementation to portfolio optimization is not so common. The main reason is due to its complexity and also because the Markowitz's one is widely used by security analysts with the respective evidence given. However, it is shown that this new model has been increasing in popularity.

Other critics pointed to the original model revealed that it fails on capturing the real essence of risk, which is the chance of incurring in a loss. Sing and Ong in one of their works [13] proposed a new method of calculating it which results on portfolios less risky than the ones generated by the Markowitz's model, when both are compared using the same risk measure. Although this new approach, normally designated as Downside Risk Framework has gained some interest by portfolio managers, the argument which states its advantage, in respect to produce less risky portfolios, cannot be of major importance when choosing one of the models since in real-life, investors are more concerned with the total return of the portfolio than with risk. More details about this comparison are addressed by Cheng and Wolverton [14].

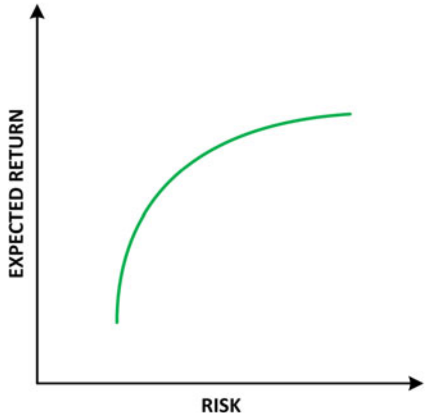


Fig. 2.4 The efficient/pareto frontier

2.4.2 Solving Markowitz's Model

Nowadays there are several techniques which can be employed to compute an efficient combination of the portfolio's expected return and the variance between its assets, in order to follow Markowitz's maxim. More concretely, these methods concentrate their efforts on computing the Efficient Frontier (EF), a line composed of optimal portfolios. In the following, different methodologies used to calculate the EF are explored.

2.4.2.1 Quadratic Programming

Given Markowitz's model presented in appendix A, if the problem is solved as a function of R, one can obtain a set of optimal solutions which constitute the efficient frontier. This curve, also known as Pareto Frontier gives for each expected return the minimum associated risk. As stated by Markowitz, from the EF the set of all efficient portfolios can be obtained. A financial portfolio is efficient if for any given expected return there is no other portfolio with a lower risk, and for any given risk there is no other portfolio with a higher value of expected return. Figure 2.4 exemplifies the EF.

Given a set of assets, there are several tools capable of computing a single point in the efficient frontier or the whole curve. If the goal is to compute a single point, then a Quadratic Programming (QP) solver is sufficient, given Markowitz's model. A list of QP solvers can be found at [15].

If the objective is to compute the whole frontier, then a subtle change in Markowitz's model is necessary; the expected return RP is removed from the set of constraints and its maximization is added as a new objective. In order to calculate it, it is possible to use an active set algorithm for QP such as the Critical Line Algorithm (CLA) [2].

2.4.2.2 Modeling Real World

Although Markowitz's model became revolutionary for the portfolio selection problem, it's important to take into account that his design only forms a theoretical point of view since in the real world much more restrictions are necessary to consider, like transaction costs or industrial regulations.

At the present day, when applying a computational procedure to solve this problem in a real world, several restrictions are considered, namely cardinality constraints, buy-in thresholds, floor, ceiling, round-lots and transaction cost inclusions. In order to get a better understanding about these restrictions, the following definitions are presented:

- *Cardinality Constraints*. The maximum and minimum number of assets that a portfolio manager wants to include in the portfolio;
- *Floor and Ceiling*. The lower and upper proportion limit specified for each security;
- *Buy-in Thresholds*. Common name to design the floor constraint;
- *Round-lot*. The number of any asset included in the portfolio must be multiple of normal trading lot (100). This constraint is applied in several of the presented publications. However, nowadays, it is not applied anymore;
- *Budget Constraint*. Requires that all the capital should be invested in the portfolio.

Taking into account these more complex constraints, two different approaches can be used to find solutions for the portfolio selection problem. One uses a suitable mixed integer solver, and the other one, metaheuristics to compute the solution. Stein et al. [16] explored both techniques and defined the respective advantages and disadvantages. They concluded that using a mixed integer approach has a major drawback since exact solutions are unsuccessful when applied to large-scale problems. Despite the fact that the use of metaheuristics comes with several handicaps, such as the requirement of extensive parameter tuning which implies the realization of a variety of tests in order to find the appropriated values, most of the articles recently published focus on this methodology since they are capable of finding reasonable solutions very quickly, allow the use of alternative risk measures, and can be easily applied to different models of the problem. The most usual metaheuristics applied on the portfolio selection problem are Genetic Algorithms (GAs), Simulated Annealing (SA) and Tabu Search (TS).

2.4.3 Metaheuristics Approaches to Portfolio Optimization

During this section, several heuristic approaches to solve the portfolio optimization problematic are addressed as well the respective variants.

2.4.3.1 Single-objective Evolutionary Algorithms

Starting with GAs, the first approaches to emerge consisted on considering a single-objective optimization problem by using a trade-off function relating risk and return, instead of considering a Multi-objective Optimization Problem (MOOP) where the goal is to optimize simultaneously two conflicting objectives, in this case, minimizing risk and maximizing the return of the portfolio. This original approximation was made by Loraschy et al. [17]. Two years later the same authors proposed a distributed version of their former algorithm with much better results [18]. Instead of considering the variance as a risk measure, as it is proposed by Markowitz, they opt to use the Downside Risk approach, referred on Sect. 2.4.1.2. Their distributed version was based on an island model where a GA is used with multiple independent subpopulations running on distinct processors. From time to time, highly-fit individuals migrate between those subpopulations.

Later, in 2000, Chang et al. [19] conducted an investigation where they experimented a variety of metaheuristics, namely GAs, SA and TS. The accomplished tests on deciding which heuristic performed better were not conclusive. First, they tried to check which one was the best to approximate the efficient frontier taking into account the original Markowitz's model. The results showed that genetic algorithms were the best approach, immediately followed by simulated annealing and by last, tabu search. In the second experience, they start to enrich Markowitz's model with cardinality constraints and then applied the algorithms. This time the differences between the three heuristics were not so clear, concluding that the best approach was to run all three and combine their results. Again, the same consideration was used in respect to have a single objective which relates risk and return, and that needs to be minimized. This claim was also confirmed by Busetti [20]. However, in 2002, Schaerf [21] developed an improved version of Chang et al.'s TS algorithm, after combining different neighborhood relations. His results were contradictory with the work already done since he proved that TS performed clearly better when compared with SA.

In all the referred proposals [17–19] the portfolio's used representation was based on two distinct lists, one identifying the assets included in the portfolio (Q) and another one with the respective investment allocation (S), as defined in the following example:

$$Q = \{\text{AMZN},\ \text{GOOG}\}\ S = \{0.6, 0.9\}$$

The portfolio is composed by two distinct securities, AMZN and GOOG, and the respective investment allocation corresponds to a total of sixty percent on Amazon and ninety percent on Google.

Notice that not all portfolios considered representations correspond to feasible solutions, i.e. solutions where the considered constraints are not violated which explains why the sum of the percentage allocations is not equal to 100 %. When applying optimization methods as the mentioned ones, several considerations can be made on how to handle these infeasible solutions.

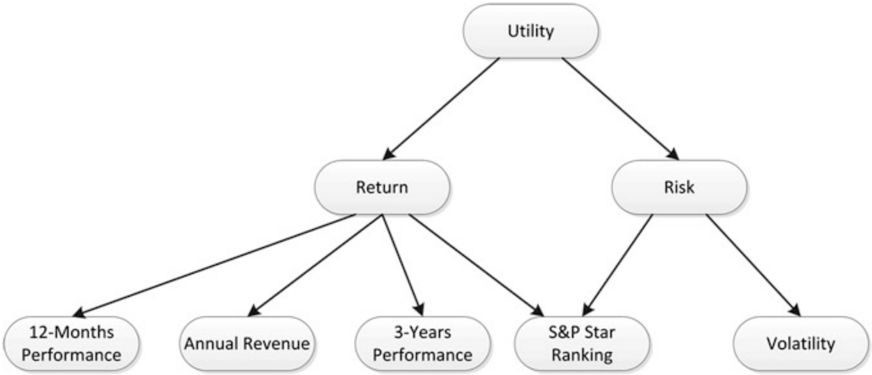


Fig. 2.5 Objective decomposition

Later, Crama and Schyns [22] developed a more sophisticated approach of the SA algorithm. They started to enrich Markowitz's classical model with additional realistic constraints, such as floor and ceiling, turnover, trading and cardinality constraints, solving the problem via a SA algorithm. Their portfolio's representation assumed to be the same as the former proposals. As the previous authors who approached this problem, the most difficult task was in how to handle the considered constraints. Solving this question, they concluded that the proposed method and similar ones like genetic algorithms were versatile enough, not requiring any modification, in case of considering other risk measures or arbitrary constraints.

Other approaches using completely different metaheuristics were also tried. Cura [23], for instance, used a Particle Swarm Optimization (PSO) technique. The author compared his heuristic performance with the three heuristics used by Chang et al. [19]. The results showed that none of the tested methods clearly outperformed the others, although this new model gave better results "when dealing with problem instances that demand portfolios with a low risk investment". Similar comparisons with TS, GAs and SA were made by Férnandez and Gómez [24] when using Artificial Neural Networks (ANN).

Regarding the use of these simple metaheuristics, another interesting publication was made by Ehrgott et al. [25]. The three authors proposed an extension to the Markowitz's model. Instead of considering only the risk and return associated to the portfolio, they used an alternative decision criteria based on an objective hierarchy. They establish a decomposition of risk in two criteria, the volatility of an investment and an S&P investments fund ranking. The return was split in four objectives, such as 12-Month Performance, 3-Year Performance, annual revenue and also the S&P ranking. Defined this model based on a Multi Decision Criteria, they applied SA, TS and GAs to solve the resulting problem. After comparing the different heuristics, the GA approach seemed to be the most reasonable one, presenting better results. The proposed model can be defined by the diagram presented in Fig 2.5.

A similar approach was taken by Lin and Gen [26] after considering a multi-stage decision-based algorithm. They start to select 20 % of the considered assets based on the past 3-Month Performance returns and only then apply the algorithm, in this case a genetic one. Their conviction settles on the fact that this initial process of restraining the set of considered securities can produce portfolios with higher returns.

2.4.3.2 Multi-objective Evolutionary Algorithms

Subsequently, the first approaches using Multi-objective Evolutionary Algorithms (MOEAs) start to arise, being Streichert et al. [27] the patriarchs. The authors made several tests using different solution representations and considering three real world constraints, namely cardinality constraints, buy-in thresholds and round-lots.

First, they formulated the problem to optimize, extending Markowitz's model with the mentioned constraints. Since it is a MOEA, their goal was to optimize two conflicting objectives, maximizing return and minimizing risk. Besides their different representation evaluation, they also employed a Memetic Algorithm (MA). A MA extends an EA approach by adding a new procedure on the EA process. This new step consists on performing a local search algorithm before evaluating a population in order to refine its individuals. This mechanism updates the decision variables (allocation investment percentages) so they can be inherited to the next generation which is known as the Lamarckism mechanism. In their case, the local search algorithm was applied to convert an infeasible solution to a feasible one which respects the considered constraints (cardinality, buy-in, round-lots). Their solution achieved better results when compared with one which tries to punish infeasible individuals.

Until that time, almost all the published works which were based on the use of GAs to solve the portfolio selection problem focused their genomic representation on a real-valued array [26–28] where each element represents the investment allocation on a specific asset, a binary string array [29] where each element expresses the asset allocation on a binary form, a hybrid approach [28, 30] where a real-valued array is used with a bit mask array; the value one indicates the inclusion of the asset on the portfolio and zero its absence, or recurring to the use of two distinct lists Q and S, as it was already mentioned. Streichert et al. [29] were the first to address an experiment in order to determine which representation was the most appropriated to handle the portfolio representation. They easily concluded that a hybrid representation where the investment allocation is represented by an array value clearly surpasses the other ones. In order to understand better this kind of approach, the following table is provided (Table 2.1).

The presented representation is easy to understand, a genome identifies a portfolio composed by five assets. The assets AMZN, INTL and YHOO are included on the portfolio with the respective allocations expressed by the weight's array. These values are then changed in order to maintain the model constraints, such as the budget constraint which specifies that the weight's sum is equal to one.

Table 2.1 A portfolio hybrid representation

Stock	AMZN	GOOG	INTL	MSFT	YHOO
Inclusion	1	0	1	0	1
Weight	0.32	0.17	0.02	0.44	0.12

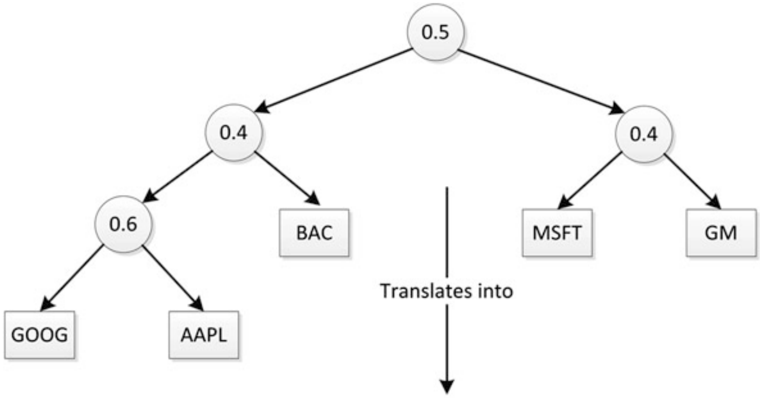
Table 2.2 Fuzzy rules. Retrieved from [30]

Return ǀ Risk	Min	Very low	Low	Moderate	High	Very high
Max	Certain	Highly Likely	Highly likely	Likely	Likely	Probably
Very high	Highly Likely	Highly Likely	Likely	Likely	Probably	Probably
High	Highly Likely	Likely	Likely	Likely	Probably	Probably
Moderate	Likely	Likely	Likely	Probably	Unlikely	Highly unlikely
Low	Likely	Probably	Probably	Unlikely	Unlikely	Highly unlikely
Very low	Probably	Probably	Probably	Unlikely	Highly unlikely	Never

Another interesting paper related with MOEAs was made by Skolpadungket, Dahal and Harnporncha [30]. The authors investigated the performance of several multi-objective evolutionary algorithms to solve the portfolio optimization problem, considering cardinality, floor and round-lot constraints. Their experiments focus on determining which algorithm performed better among three different ones. The first one to be evaluated was the Vector Evaluated Genetic Algorithm (VEGA) which consists on an extended version of the single GA to handle multi-objectives. Secondly, they used the traditional MOEA which was specified by the Multi-objective Genetic Algorithm (MOGA). Finally, they tested advanced algorithms such as Strength Pareto Evolutionary Algorithm II (SPEA2) and a Non-dominated Sorting Genetic Algorithm II (NSGA2). Their experiments determined that SPEA2 performed better when compared with its cohorts. Another interesting point to retrieve from their findings is the fact that the simplest GA, in particularly VEGA, when extended with a fuzzy logic mechanism suffered major improvements. The authors employed the following fuzzy rules (Table 2.2).

This fuzzy mechanism specifies the probability of selection for each individual through the implementation of a fuzzy decision rule which combines two objectives, risk and return. This technique is helpful in order to facilitate the trade-off between these two measures; if the return is maximum and the risk minimum, then it is certain the selection of that individual, if the return is very low and the risk is very high, then that individual will never be selected.

Although these approaches are also based on the use of evolutionary algorithms as the previous ones listed on Sect. 2.4.3.1, the main difference between single objective EAs and MOEAs is the way the solutions are ranked. Single objective EAs are characterized by evaluate a portfolio solution through a trade-off function that relates risk and return. MOEAs try to rank solutions evaluating risk and return



Asset	GOOG	AAPL	BAC	MSFT	GM
Weight	0.12	0.08	0.3	0.2	0.3

Fig. 2.6 A portfolio tree-based representation

separately as two distinct objectives to achieve. One point it should be noticed is that is still not clear if multi-objective genetic algorithms perform better than using a single objective genetic algorithm with a trade-off function to relate portfolio's risk and return. However, the MOEAs have as major advantage the fact that is possible to generate the set of solutions, i.e. an approximation of the efficient frontier in a single run.

2.4.3.3 Extensions to Genetic Algorithms

More recently, extensions to the classical single-objective genetic algorithm's approach were experimented; Aranha and Hitoshi [6] proposed a completely distinct representation of the portfolio using a tree-based structure, represented in Fig. 2.6. They conducted several experiments in order to support their choice, concluding that this new representation accelerates the evolution of a good solution. They were able to produce concise portfolios with the same utility as the ones generated when using an array-based structure. This fact brings several benefits since it permits the trading costs reduction and the increase of the portfolio's understandability. Although this is still an early work since this representation was never proposed before, it clearly gives a good starting point on a portfolio's representation when using genetic algorithms coupled with Markowitz's model. However, notice that more tests must be done since the authors considered only the original Markowitz's model without additional constraints.

The same authors [28] also tried a new approach, extending the traditional GA version with a modeling cost mechanism which can be employed to take into consideration the previous held investments. The authors had in consideration the asset's weights held previously in the portfolio so they could take into account the costs related with buying and selling stocks that are needed to change the portfolio structure. Their goal was to minimize transaction costs by minimizing the difference between the previous held portfolio and the actual portfolio. In order to accomplish this feature, they defined the minimization of the Euclidean distance between the portfolios as a secondary objective, reached via a technique called Objective Sharing, avoiding that way the necessity of defining a MOOP. In order to maintain the consistency between the portfolios over time, they also introduced a mechanism called Population Seeding. This experience allows the possibility to get a more realistic approximation to the practical portfolio management. Although these authors were not the first on addressing the problem of considering transaction costs, the proposed approach seems to be the more realistic on how to handle this problematic. The same authors on the following year provided a more robust solution using the previous mechanism with a Memetic Algorithm [31].

In respect to extensions regarding the multi-objective evolutionary approach, Branke et al. [32] developed a system based on the combination between a multi-objective evolutionary algorithm and the Critical Line Algorithm (CLA) [2]. The considered process consists on dividing the problem into a subset of problems recurring to a MOEA. The CLA is then executed in each subset in order to produce a solution which forms a partial front designated as an envelope. Further, the EA is used to find a sequence of such envelopes which form a solution to the starting problem. Instead of representing each solution as a single-point in the efficient frontier, each solution passes to be represented as a partial front, the envelope.;

2.4.4 Technical and Fundamental Analysis in Portfolio Management

A completely different way on handling the portfolio problematic consists on performing a market evaluation based on technical and fundamental analysis, already explained in the beginning of this chapter.

It was already mentioned that technical analysis consists on studying stock charts in order to find over or undervalued stocks. Fundamental Analysis evaluates each security by measuring its intrinsic value through the study of overall economy, industry conditions and financial conditions of a specific company in order to produce a value which can be compared to the current company's price.

The first problem addressed by the exclusive use of these indicators consists on guarantying the diversification of the portfolio due to the absence of a model such as the Markowitz's one, to reduce the correlation between assets. Secondly, the risk involved in their utilization can be substantially high, and thirdly, how to decide the investment allocation percentage on each security, without doing it uniformly. Despite the presence of these problems on using such methodologies, the use of such procedures can reward us with a greater profitability since their

Table 2.3 A new portfolio representation

Stock number	1	2	…	10
Stock identification	1–300	1–300	1–300	1–300
Normalized percentage	0–1	0–1	0–1	0–1
Value added by indicators	0–1	0–1	0–1	0–1

Table 2.4 Technical rules. Based on [39]

Risk	1. Today's price—average price of the previous 12 trading days
	2. Today's price—average price of the previous 50 trading days
	3. Today's price—maximum price of the previous 5 trading days
	4. Today's price—maximum price of the previous 50 trading days
Return	5. Today's price—minimum price of the previous 5 trading days
	6. Today's price—minimum price of the previous 63 trading days

main goal consists on finding overvalued and undervalued stocks to produce profit. Due to its potential, it's possible to achieve better returns, not only with the rising of security prices but also with their decline.

Liad Wagman [33] proposed a Genetic Programming (GP) approach based on the use of technical analysis. The author starts to form an initial population constituted by 1000 different portfolios, each of them composed by ten randomly stocks retrieved from an index formed with 300 distinct stocks. The investment percentage allocation for each stock is randomly assigned over these initial portfolios.

Each individual in the population, i.e. portfolio, is represented as the following (Table 2.3):

- Stock Number—Identifies the stock within the portfolio;
- Stock Identification—Identifies the stock within the index;
- Normalized Percentage—Investment allocation percentage;
- Value Added by Indicators—Percentage provided by the satisfaction of a variety of technical rules.

Each portfolio is evaluated through six technical rules responsible for generating "buy" or "not-buy" signals. These rules are generated from return and risk measures which calculation is based on the following technical indicators (Table 2.4):

Stipulated those indicators, the following rules are defined:

- Moving Average Rules (1) and (2)—Generate "buy" signals if equations (1) and (2) are greater than zero, respectively;
- Trading Range Breakout Rules (3) and (4)—Generate "buy" signals if equations (3) and (4) are greater than zero, respectively;
- Filter Rules (5) and (6)—Generate "buy" signals if today's price has risen 1 % in respect to the minimum of previous 5 or 63 days, respectively.

All the presented values are based on [34].

These six rules are latter mapped to a percentage value, according to the respective weight. The author considered a 60 % risk value versus a 40 % return value. Each of these percentage values is uniformly distributed over the respective rules; 15 % for each risk rule and 20 % for the two return rules. When each rule is satisfied it adds the respective weight to the overall fitness of the solution. For example, considering the past six months performance, if for stock number 6 which has a weight of 10 % in portfolio, only rule (3) is satisfied, and within the months 3, 4 and 5, then:

$$Value\ Added\ by\ Indicators_6 = 0.10 * \frac{(0.15 + 0.15 + 0.15)}{6} \tag{2.2}$$

The total fitness of the portfolio is calculated via the weighted average of these indicators' value, considering the normalized percentage values as the respective weights. Notice that the proposed work only aims on establishing a specific portfolio and maintain it indefinitely without having management consideration issues.

Another interesting approach was followed by Wei Yan et al. [34, 35]. The authors provide a portfolio construction system based on two distinct techniques, GP and Support Vector Machines (SVM). Both techniques are extended with a voting mechanism, and subsequent comparison is performed. Their GP application consists on a genetic programming algorithm coupled with an investment simulator. Each time an individual is evaluated through the fitness function, the investment simulator is executed. Each individual is represented by a factor model which consists on a table with 19 factors described by 18 fundamental indicators and one technical indicator, the Moving Average Convergence/Divergence (MACD). That individual is calculated considering that month's data. Further, based on this model, each market's stock is ranked. The stocks are then grouped on four market sectors and within each one they are ranked according to their expected return. The simulator then performs the following decisions:

- Top 3 stocks of each sector are bought, the bottom 3 are sold or go short;
- Sectors are equally weighted and each stock is given equal weight in the portfolio.

At the end of each month all the positions are closed and the profit or loss is calculated.

Although there aren't practically any published approaches using a variety of technical indicators, the referred works employ them, but in a very limited way; one only uses the Moving Average (MA) indicator and the other one a MACD indicator. Since there are many works that validate the application of technical indicators to buy or sell individual stocks, it will be interesting to deeply investigate more of those indicators in order to generate profitable portfolios. There's an infinity of technical indicators, the most widely used are described on [36]. Although it seems that if everyone uses those indicators it will get the same results, the premise is incorrect since there's a lot to explore on using them, such as the parameter specification. Also, the preferences of each investor can change. The person can opt for a more aggressive or more passive strategy, adapting the indicators to his will. Blanco et al. [37, 38] conducted an interesting study on investigating the optimization of some of these indicators using EAs.

2.4.5 *Overview and Discussion*

It's extremely difficult to evaluate the designed strategies in terms of profitability since most of them are applied to different market periods. Regarding the active versus passive question, an active design will try to beat the market which can probably produce higher levels of profitability when compared with a simple passive strategy, using the Markowitz's formulation. Possibly, its application conjugated with evolutionary computation is not so common due to the fact that technical and fundamental analysis requires a deep investigation on his functionality in order to one develop a solution based on its potential. Normally, it is an unfamiliar subject to most of the computational intelligence specialists which results on the employment of the widely known formulation, the Markowitz's model. When applying the notorious model, these scientists can concentrate their efforts on improving its expertise area, changing the structure and combining additional mechanisms in order to produce better and faster metaheuristics to solve this mathematical model, rather than studying other approaches which will require a deeper knowledge on economical facts.

2.5 Conclusions

From the several presented works given on this chapter, and which are briefly summarized on the following tables, it is possible to observe that most part of these solutions apply GAs to approach the portfolio problematic. Notice that a ranking involving all the different approaches presented on Table 2.5 was not performed because it is extremely hard to evaluate most part of these strategies since they are applied to different market conditions and periods. Also, the major part of these works has as principal objective the calculation of the efficient frontier in order to validate the proposed algorithm. Although the comparison is difficult, it is clear that the majority of the presented works use GAs; on several of these works where distinct optimization techniques were compared, the results showed that GAs were capable of surpassing the competitor methodologies. Based on these results, the intent of this work was to develop an application using a GA as an optimization technique.

In respect to the question active versus passive, from the previous table it is possible to observe that most part of the solutions concentrate their work on using the Markowitz's model to analyze the market, and subsequently pick the most promising stocks, according to the formulation. However, active management approaches using technical analysis can reward us with higher profitability levels, since their major intent is to beat the market, saying this; the best and most innovative way of approaching this problem is to use technical analysis to find under and overvalued stocks in the market. Given the reasons explained above and

Table 2.5 An overview over different approaches on portfolio optimization. Consult Appendix C to understand the parameters

Work	Years	Meta Heuristic	Additional features	Constraints	Portfolio analysis	Portfolio Rep.	MOOP?	Evaluation function	Data used	Training period	Testing period
[18]	1995	GA	Distributed version	–	Downside risk framework	Unknown	No	Lambda trade-off function	Prices from 100 unknown assets	Unknown	*
[19]	2000	SA GA TS	–	Floor Ceiling Cardinality	Markowitz	Real value array structure	No	Lambda trade-off function	Weekly prices from Hang Seng, DAX100, FTSE100, S&P100, Nikkei225	March 1992 – Sept. 1997	*
[21]	2002	HC TS SA	– Token-ring procedure –	Floor Ceiling Cardinality	Markowitz	Hybrid structure	No	Linear combination between portfolio's variance and portfolio's return	Weekly prices from Hang Seng, Dax100, FTSE100, S&P100, Nikkei225	March 1992 - Sept. 1997	*
[22]	2003	SA	–	Floor Ceiling Cardinality Turnover Trading	Markowitz	Real value array structure	No	Portfolio's variance	Weekly prices from 151 US stocks	Jan. 1988 – April 1997	*
[23]	2008	PSO	–	Floor Ceiling Cardinality	Markowitz	Hybrid structure	No	Lambda trade-off function	Weekly prices from Hang Seng, Dax100, FTSE100, S&P100, Nikkei225	March 1992 – Sept. 1997	*

(continued)

Table 2.5 (continued)

Work	Years	Meta Heuristic	Additional features	Constraints	Portfolio analysis	Portfolio Rep.	MOOP?	Evaluation function	Data used	Training period	Testing period
[24]	2007	ANN	–	Floor Ceiling Cardinality	Markowitz	Hybrid structure	No	Lambda trade-off function	Weekly prices from Hang Seng, Dax100, FTSE100, S&P100, Nikkei225	March 1992 – Sept. 1997	*
[26]	2007	GA	Multistage decision	–	Markowitz	Real value array structure	No	Sharpe ratio	40 assets from Taiwan's stock market	Unknown	*
[27]	2003	GA	Lamarckism mechanism	Cardinality Buy-in thresholds Round-lots	Markowitz	Binary string structure Real value array structure Hybrid real structure	Yes	Minimization portfolio's variance Maximization portfolio's return	Prices from Hang Seng	Unknown	*
[6]	2008	GA	–	–	Markowitz	Tree structure	No	Sharpe ratio	Prices from Nasdaq100 and Nikkei225	2003 2004 2005	Jan. 2004 Jan. 2005 Jan. 2006
[32]	2008	NSGA2	Critical line algorithm integration	Buy-in thresholds Cardinality 5-10-40	Markowitz	Hybrid real structure	Yes	Minimization portfolio's variance Maximization portfolio's return	Weekly prices from Hang Seng, S&P100, Nikkei225	Unknown	*

(continued)

Table 2.5 (continued)

Work	Years	Meta Heuristic	Additional features	Constraints	Portfolio analysis	Portfolio Rep.	MOOP?	Evaluation function	Data used	Training period	Testing period
[28]	2007	GA	Objective sharing Population seeding	Multi-period consideration	Markowitz	Hybrid real structure	No	Sharpe ratio	Prices from Nasdaq 100	Nov. 2000 – Oct. 2006	Last 53 months of the indicated periods.
[31]	2009	GA	MA inclusion	Multi-period consideration	Markowitz	Tree structure	No	Sharpe ratio	Prices form Nasdaq100, S&P500	2006 2007	2007 2008
[30]	2007	VEGA MOGA NSGA2	Fuzzy – –	Cardinality Buy-in thresholds Round-lots	Markowitz	Hybrid real structure	Yes	Minimization portfolio's variance Maximization portfolio's return	Weekly prices from Hang Seng market	Unknown	*
[33]	2003	GP	–	Cardinality	Technical rules	Predefined structure	No	Maximization of technical rules satisfaction	Prices from Down Jones industrial average	1979 1980	1980
[25]	2004	GA	MCDM	Cardinality Round-lots	Markowitz's based on risk and return decomposition	Predefined structure	Yes	Minimization portfolio's variance Maximization portfolio's return	Unknown	Unknown	*
[34]	2008	GP	–	Cardinality Multi-period consideration	1 technical rule 18 fundamental indicators	Predefined structure	Yes	Sharpe ratio	33 Malaysian stocks	Jan. 1999 – Dec. 2004	Jul. 1997 - Dec. 1998

the performed study on the developed works, it is proposed a solution based on technical analysis coupled with GAs.

Table 2.5 summarizes the approaches given, classified according to specific parameters. For a better understanding of this table, consult Appendix C.

Besides the presented table below, under Appendix B it is possible to observe a list of commercial applications based on technical analysis and portfolio management.

References

1. Maginn JL, Tuttle DL, McLeavey DW, Pinto JE (2007) Managing investment portfolios: a dynamic process. CFA Institute Investment Series. Wiley, New Jersey
2. Markowitz HM, Todd GP, Sharpe WF (2000) Mean variance analysis in portfolio choice and capital markets. Wiley, New Jersey
3. Malkiel B (1973) A random walk down wall street. W. W. Norton & Company, New York
4. Markowitz HM (1972) Portfolio selection. J Financ 7:77–91
5. Goodman B (2002) Passive management: it's not an oxymoron. http://www.thestreet.com/author/1651333/BeverlyGoodman/all.html. Accessed 20 Aug 2009
6. Aranha C, Hitoshi I (2008) A tree-based GA representation for the portfolio optimization problem. Genetic Evolutio Computation Conference (GECCO), Atlanta, pp 873–880
7. Achelis S (2000) Technical analysis from A to Z. McGraw-Hill, New York
8. Hamsapiya SS, Surekha P (2008) Evolutionary intelligence: an introduction to theory and applications with matlab. Springer, Heidelberg
9. Mitchell M (1999) An introduction to genetic algorithms. MIT Press, Cambridge
10. Eiben AE, Smith JE (2007) Introduction to evolutionary computing. Springer, Berlin
11. Black F, Litterman R (1992) Global portfolio optimization. Financ Anal J 48:28–43
12. Salomons A (2007) The Black-Litterman model hype or improvement? Dissertation, University of Groningen
13. Sing TF, Ong SE (2000) Asset allocation in a downside risk framework. J Real Estate Portfol Manage 6:213–223
14. Cheng P, Wolverton M (2001) MPT and the downside risk framework: a comment on two recent studies. J Real Estate Portfol Manage 7:125–131
15. Gould N, Toint P (2008) Quadratic programming solvers. http://www.numerical.rl.ac.uk/qp/qp.htm. Accessed 15 Aug 2009
16. Stein M, Branke J, Schmeck H (2005) Portfolio selection: how to integrate complex constraints. J Financ Plan 68–78
17. Arnone S, Loraschi A, Tettamanzi A (1993) A genetic approach to portfolio selection. Neural Netw World III 6:597–604
18. Loraschi A, Tettamanzi A, Tomassini M, Verda P (1995) Distributed genetic algorithms with an application to portfolio selection problems. In: Pearson D, Steele N, Albrecht R (ed) In Artificial neural nets and genetic algorithms, Springer-Verlag, Wien
19. Chang TJ, Meade N, Beasley JE, Sharaiha YM (2007) Heuristics for cardinality constrained portfolio optimization. Comput Oper Res 27:1271–1302
20. Busetti FR (2000) Metaheuristic approaches to realistic portfolio optimization. Dissertation, University of South Africa
21. Schaerf A (2002) Local search techniques for constrained portfolio selection problems. J Comput Econ 20:177–190
22. Crama Y, Schyns M (2003) Simulated annealing for complex portfolio selection problems. Eur J Oper Res 34:1177–1191

23. Cura T (2008) Particle swarm optimization approach to portfolio optimization. Nonlinear Anal: Real World Appl 10:2396–2406
24. Fernández A, Gómez S (2004) Portfolio selection using neural networks. Comput Oper Res 34:1177–1191
25. Ehrgott M, Klamroth M, Schwehm C (2004) An MCDM approach to portfolio optimization. Eur J Oper Res 155:752–770
26. Lin C, Gen M (2007) An effective decision-based genetic algorithm approach to multiobjective portfolio optimization problem. Appl Math Sci 1:201–210
27. Streichert F, Ulmer H, Zell A (2003) Evolutionary algorithms and the cardinality constrained portfolio selection problem. Oper Res Proc 3–5
28. Aranha C, Iba H (2007) Modelling cost into a genetic algorithm-based portfolio optimization system by seeding and objective sharing. In: Genetic and evolutionary computation conference (GECCO), London, pp 196–203
29. Streichert F, Ulmer H, Zell A (2004) Comparing discrete and continuous genotypes on the constrained portfolio selection problem. In: Genetic and evolutionary computation conference (GECCO), Seattle, pp 1239–1250
30. Skolpadungket P, Dahal K, Hampomchai N (2007) Portfolio optimization using multi-objective genetic algorithms. In: Congress evolutionary computation, Singapore, pp 516–523
31. Aranha C, Hitoshi I (2009) Using memetic algorithms to improve portfolio performance in static and dynamic trading scenarios. In: Genetic and evolutionary computation conference (GECCO), Montreal
32. Branke J, Scheckenbach B, Stein M, Deb K, Schmeck H (2008) Portfolio optimization with an envelope-based multi-objective evolutionary algorithm. Eur J Oper Res 199:684–693
33. Wagman L (2003) Stock portfolio evaluation: an application of genetic-programming-based technical analysis. Gen Algorithms Gen Program, Stanford, pp 213–220
34. Yan W, Sewell M, Clack C (2008) Learning to Optimize Profits Beats Predicting Returns—Comparing Techniques for Financial Portfolio Optimization. Proc 10th annu conf Gen and evol computation, 1681-1688. Atlanta, USA
35. Hassan G, Clack C (2009) Robustness of multiple objective GP stock-picking in unstable financial markets. In: Genetic and evolutionary computation conference (GECCO), Montreal, pp 1513–1520
36. Murphy JJ (1999) Technical analysis of the financial markets: a comprehensive guide to trading methods and applications. Prentice Hall Press, New York
37. Férnandez-Blanco P, Bodas-Sagi D, Hidalgo JI (2008) Technical market indicators optimization using evolutionary algorithms. In: Genetic and evolutionary computation conference (GECCO), Atlanta, pp 1851–1858
38. Bodas-Sagi D, Fernández P, Hidalgo JI, Soltero F, Risco-Martín J (2009) Multiobjective optimization of technical market indicators. In: Genetic and evolutionary computation conference (GECCO), Montreal, pp 1999–2004
39. Tsang E, Li J (1999) Improving technical analysis predictions: an application of genetic programming. Proceedings of Florida Artificial Intelligence Research Symposium. USA

Chapter 3
Solution's Architecture

Abstract The goal of this chapter is to provide the description of the developed solution to approach the portfolio management problem. It will start by presenting an overview on the application's architecture, followed by the delineation of the strategies employed, and a detailed characterization of the several modules within the system.

3.1 Overall Architecture

In order to handle the portfolio management problematic it is necessary to specify which steps should be addressed to construct such capable system, as well, answer the fundamental questions about which data can be used and what will be the composition of such application. The following diagrams (Figs. 3.1, 3.2) propose the architecture of a possible system which tries to handle the portfolio management issue.

The presented architecture is based on module structures, which correspond to distinct units of implementation with a specific functional responsibility within the system. As can be observed within the next figure, the system is defined by five fundamental modules which functionality can be decomposed in several others:

- *User Presentation Module*. Responsible for the user interface, its functionality can be divided across two distinct modules; the *Input Data Module* which is responsible for reading the user desired parameters, his investment goals; and the *Output Data Module* which is accountable for presenting the calculated portfolio;
- *Financial Data Processing Module*. Controls the financial data processing. Its behavior is decomposed in three distinct modules; the *Download Module* which is responsible for downloading the company's data from financial websites.

A. M. S. B. S. Gorgulho et al., *Intelligent Financial Portfolio Composition Based on Evolutionary Computation Strategies*, SpringerBriefs in Computational Intelligence,
DOI: 10.1007/978-3-642-32989-0_3,

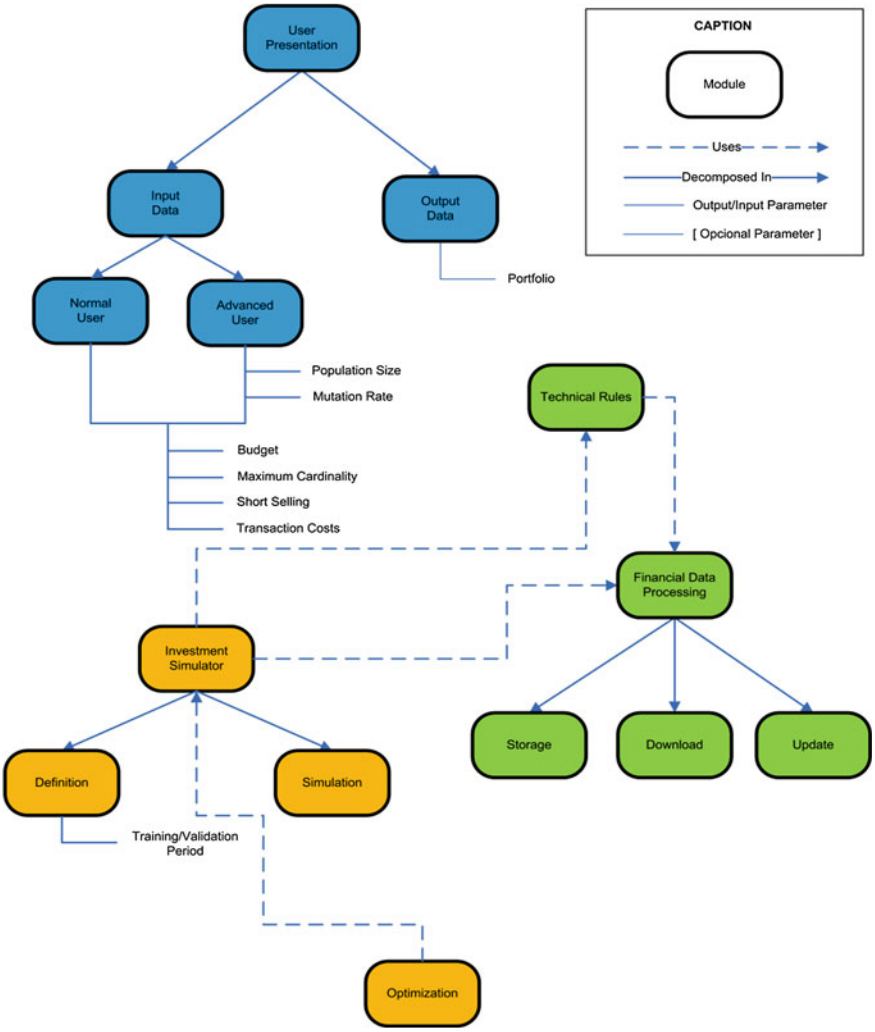


Fig. 3.1 System's architecture. A module decomposition view

The *Storage Module* which contemplates the storage of all the downloaded data over distinct files where each one is attributed to a specific company. Finally, there is an *Update Module* which will be accountable for updating the financial data on the necessary files;

- *Optimization Module*. This modules constitutes the core within the system, it is responsible for defining an optimal model for classifying different assets within the market;
- *Investment Simulator Module*. Accountable for simulating the creation of a portfolio and subsequently managing it over a specific period of time;

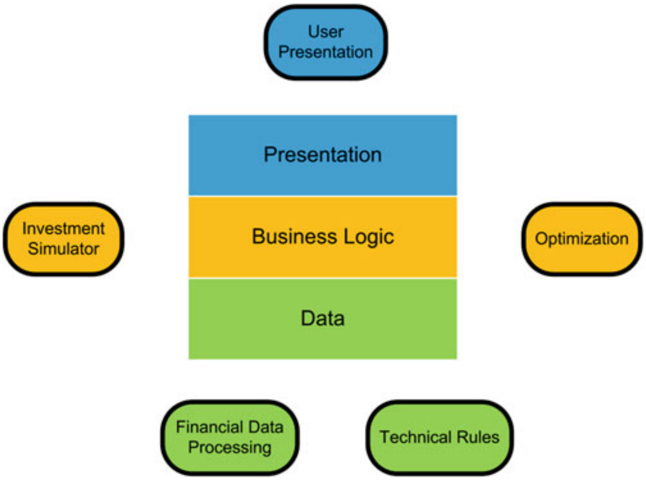


Fig. 3.2 Layered architecture

- *Technical Rules Module*. Defines the technical indicators used to predict market movements and calculates the associated data.

To simplify the previous description, the modules can be structured on a traditional three-layer style architecture.

3.2 Data Flow

In respect to the data flow within the application, very generally, the system starts to ask distinct inputs from the user, executes the optimization algorithm, and then provides the recommended portfolio. More specifically, the complete process is performed as follows:

1. The user starts by specifying the desired parameters, depending on its role, which can be *normal* or *advanced*, according to its knowledge on optimization techniques;
2. Afterwards, the system applies a set of technical indicators in order to calculate the values given by those indicators on the available data prices;
3. After this process, the GA starts its execution by defining several random individuals, which correspond to different models for classifying the market's assets. These different models, called *Classifier Equations* take into account the data calculated in the previous step;
4. In order to evaluate each individual, an *Investment Simulator* is necessary to rank each stock within the market and subsequently, picking the best stocks for defining a potential financial portfolio. Afterwards, the portfolio is updated and evaluated during the training period in order to classify the attractiveness of the

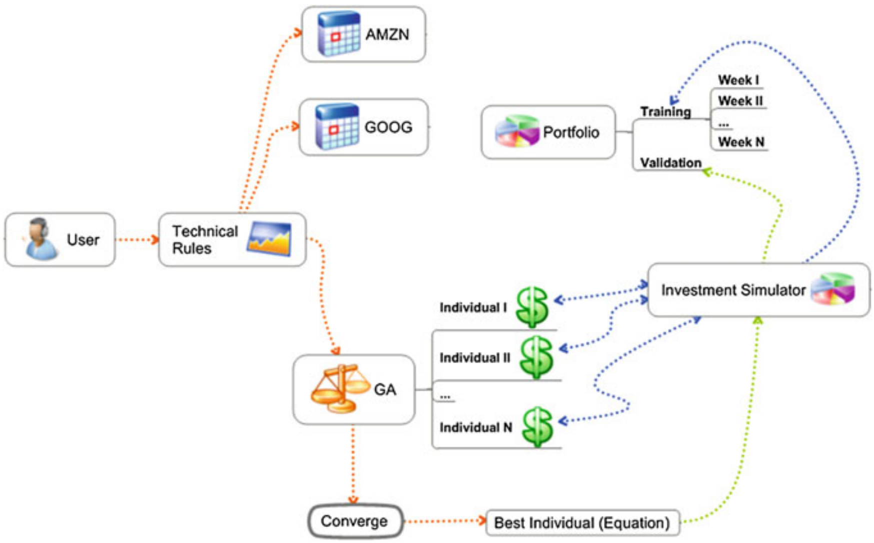


Fig. 3.3 Data flow example

current classifier equation in terms of its performance on the end of the considered time period;

5. When the GA converges in a final solution, the system executes again the investment simulator system, but to the current date period, in order to provide the recommended portfolio taking into account today's date;
6. Every week the *Investment Simulator* is again executed to update the current portfolio, adding new positions or closing former ones. From time to time, the GA process is repeated so that a new classifier equation is determined considering the most recent data.

The following scheme (Fig. 3.3) tries to illustrate the defined procedure.

In the following sections, the reader can have a more detailed view on the considered strategy and each of the presented components. The clarification will start by describing the first layer of this application which corresponds to the financial data access, and subsequent processing.

3.3 Financial Data Processing Module

This module is accountable for processing all the financial data which is of primary use on the developed application. In order to provide to the system the ability of generating real-life portfolios, it is necessary to first download a complete history of all the available data on distinct markets. This action is performed by the *Download Module*. The process of retrieving all the historical data was performed

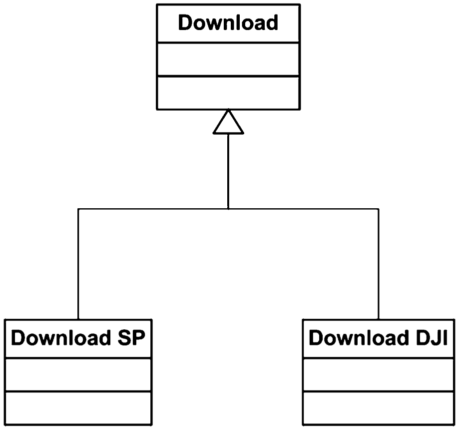


Fig. 3.4 Download decomposition

just once. Afterwards, all the data relative to the stock quotes was stored on distinct files, one for each company. This process is done by the *Storage Module*.

After the first execution, it is no longer necessary to download all the historical data. The *Update Module* is called in order to update the necessary files, according to new available information.

3.3.1 Implementation and Functionality

The *Financial Data Processing Module* was coded through the use of a main class, the *Download class* which is decomposed in several classes, one for each of the considered markets: (Fig. 3.4)

As you can see from the previous figure, two major market indexes were used:

- The DJI, *Dow Jones Industrial Average Index* [1], which contains the stock prices of 30 of the largest held companies in the United States;
- The S & P500, *Standard & Poors* 500 [2], composed by 500 of the biggest publicity held companies which trade on the two largest American stock markets; NASDAQ and the New York Stock Exchange.

All the financial data relative to the former indexes is downloaded through the Yahoo Finance Database [3]. The complete retrieving process can be described as following:d through the Yahoo Finance Database [3]. The complete retrieving process can be described as following:

1. Specify the desired index. Each index is identified with a unique keyword. For instance, the *Dow Jones Industrial Average* is tagged with the acronym DJI;
2. After defining the target index, the download process is executed and a single file containing the tickers (specific group of letters representing a particular security) of all companies composing the previously defined index, is stored.

The second process consists on downloading all the historical data, from a specific date until today's date for each of the previously acquired companies. The designer has the possibility of indicating the desired data period through a single parameter; *daily, weekly, or monthly*;

Within this download process, the storage functionality is executed, which corresponds to a specific method within the *Download* class responsible for defining *csv* files with the desired financial data. Each record within this stored file has the following configuration:

Date	Open	High	Low	Close	Volume	Adj. close

where:

- Date. The date record, using the format "dd-mm-yyyy";
- Open. The opening price in which the security was traded during a specific date;
- High. The highest price in which the stock was traded during a specific date;
- Low. The lowest price in which the equity was traded during a specific date;
- Close. The closing price in which the asset was traded during a specific date;
- Volume. The number of shares traded in a security during a specific date;
- Adj. Close. The adjusted closing price in which the stock was traded during a specific date.

3. After the complete historical data has been downloaded, when the application is again executed, the *Update* module is invoked, which corresponds to a specific method within the *Download* class, accountable for processing all companies' files and for each one identifying the last record, in particular the date of the last available record. After this processing phase, each company's file is updated; the new needed records are inserted. Notice that each data file is ordered from the oldest date to the most recent one, to allow the append process on the end of each file.

3.4 Technical Rules Module

One of the major problems that an investor faces on portfolio management is the right choice of assets; when he picks a specific stock he does not know if its price is going to rise or fall. However, several technical indicators can be used to give a future perspective on its behavior in order to determine the best choice. So, in order to classify each asset within the market, the user needs to employ a set of rules based on technical indicators. As stated already under the previous chapter, a technical indicator consists in a formula that is normally applied to stock's prices and volumes. The resulting values are plotted and then analyzed in order to offer us a perspective on price evolution. More specifically, a technical indicator tries to capture the behavior and investment psychology in order to determine if a stock is under or overvalued.

When using a technical indicator it is necessary to specify several parameters, such as the considered period of calculation. For instance, a simple indicator such as *Moving Average(x)* plots per each day, the average on prices observed during the last x days. Depending on the considered data, it is also possible to employ the indicator to weekly or monthly prices.

Based on entry/exit signals and other plot characteristics different rules can be defined, which allows scoring distinct stocks within the market and subsequently pick the best securities according to the indicators employed.

There are several problems that can show up with the use of technical indicators. First, there's not a better indicator, the indicators should be combined in order to offer us different perspectives. Sometimes a technical indicator gives false signals, so our best option is to combine different technical indicators. Second, a technical indicator always needs to be applied to a specific time span; it can be 10 days, 50 days, more or less. Determining the best time window is a hardly choice; in this case it was used the time window proposed by the technical analysis specialists, for each of the used indicators.

Regarding the GA aspects, the algorithm can be applied to technical indicators in several ways, as to determine the best time span; for instance, Fernández–Blanco et al. [4, 5] applied an EA to determine the best settings for the MACD and RSI indicators. However, in this work the algorithm is applied in the context of obtaining the best model to classify the assets, an optimal balance between different technical indicators. Since only one indicator cannot possibly serve, the software tries to find which were the best indicators to use in the past to form a basket of securities and subsequently, pick the most attractive assets. This is a hard problem, especially due to the high volume of data involved, it can be enormous when considering just one market index as the S&P 500.

In this work several technical indicators were applied to find attractive stocks in the market. The indicators were chosen in order to use a basket of different types of technical indicators; momentum oscillators and trend following devices:

- A trend indicator tries to identify trends in the market. A trend represents a consistent change in prices; A momentum based indicator tries to measure the velocity of directional price movement in order to identify the speed/strength of a price movement and the enthusiasm of buyers and sellers involved in the price development.

For each technical indicator calculated for each period (day, week, or month) in the considered data set, a score was assigned. Four distinct scores were used:

- *Very Low Score*. Assigns −1.0 points, indicates a strong sell/short signal;
- *Low Score*. Assigns −0.5 points, indicates an underperformed signal, potentially to sell or go short;

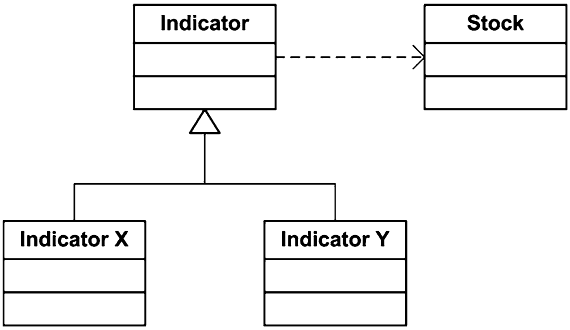


Fig. 3.5 Class dependencies for technical rules module

- *High Score*. Assigns 0.5 points, indicates a reasonable buy signal;
- *Very High Score*. Assigns 1.0 points, indicates a strong buy signal.

3.4.1 Extensibility and Technical Rules Module Implementation

As stated before, the intent was to mix different kinds of technical indicators; oscillators and trend following mechanisms. In order to respect that guideline, the indicators employed were the *Exponential Moving Average* (EMA), the *Hull Moving Average* (HMA), the *Rate of Change* (ROC), the *Double Crossover Method*, the *Relative Strength Index* (RSI), the *Moving Average Convergence Divergence* (MACD), the *True Strength Index* (TSI), and the *On Balance Volume* (OBV). Notice however, that is possible to easily extend the developed solution with more technical indicators. The algorithm is adapted for any technical indicator, the only requirement is to implement the desired indicators and define the respective rules. On adding more indicators, the confirmation of a possible buy or sell signal is possibly more accurate, improving the results of the designed solution. The parameters assigned to each technical indicator can also be changed to any value desired by the designer.

In respect to the module implementation, an *Indicator Class* was created, which is responsible for creating a set of technical indicators to each stock within the market (Fig. 3.5).

The main class, the *Indicator Class* is specified in several technical indicators, each one associated to a specific stock within the market. The codification can be easily extended; by defining a new class *Indicator Z* which implements a specific technical indicator.

In the following sections, the reader can have an insight on the indicators applied as well the respective classification rules. Moreover, notice that some of the figures presented below were retrieved by the Best Trading Pro platform [6].

Table 3.1 Rules developed for the EMA indicator

	EMA(12)
Very low score	If price line crosses below the EMA line
Low score	EMA line is decreasing
High score	EMA line is rising
Very high score	If price line crosses above the EMA line

3.4.2 Exponential Moving Average

The *Exponential Moving Average* (EMA) [7] is a trend following indicator. The goal of this device is to identify that a trend has begun, or it is finishing its cycle. In order to accomplish it, the EMA averages the price data, in order to produce a smooth line which can be easily perceived, in contrast to the irregular curve signaling the prices. There are several kinds of moving averages; the exponential one assigns more weight to the most recent data in order to give more importance to it and is described by:

$$EMA_{\mathrm{t}}(n) = EMA_{\mathrm{t}-1}(n) * \left(1 - \frac{2}{n+1}\right) + X_t * \frac{2}{n+1} \tag{3.1}$$

where:

- n is the length of the Moving Average;
- X corresponds to the stock's price;
- t defines the considered period (day, week, or month).

Based on this indicator, the rules were defined in Table 3.1.

The following picture provides an example of the EMA line. As you can see, it defines a smoothing curve which can be easily analyzed, in contrast to the zigzag performed by the stock's prices (Fig. 3.6).

3.4.3 Hull Moving Average

Like the EMA, the *Hull Moving Average* (HMA) [8] tries to identify the prevailing market trend. However, it can define a smoother curve and can follow the price graph much more closely, reducing the lag present on its predecessor moving average and is described by:

$$HMA_{\mathrm{t}}(n) = WMA_{\mathrm{t}}\big(floor\big(\sqrt{n}\big)\big)\ of\ \left(2 * WMA_t\left(floor\left(\frac{n}{2}\right)\right) - WMA_t(n)\right) \tag{3.2}$$

where:

- n is the length of the Moving Average;

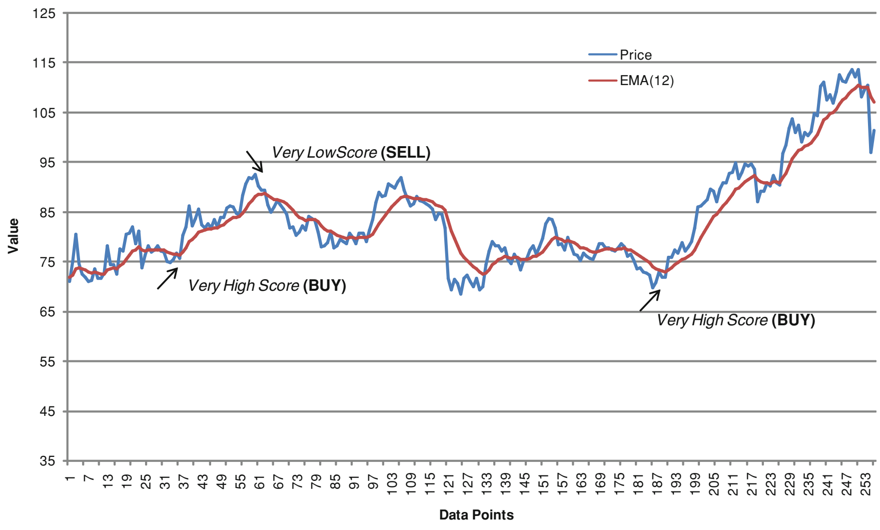


Fig. 3.6 An example of the EMA indicator

Table 3.2 Rules developed for the HMA indicator

	HMA(16)
Very low score	HMA slope changes to a downward direction
Low score	HMA line is decreasing
High score	HMA line is rising
Very high score	HMA slope changes to an upward direction

- *WMA* corresponds to the weighted moving average;
- *t* defines the considered period (day, week, or month).

For the HMA indicator, the rules were defined in Table 3.2.

The following picture provides an example of the HMA line (Fig. 3.7).

3.4.4 Double Crossover

The *Double Crossover* [7] method is characterized by the using of two distinct moving averages to generate market signals. Normally, it is made a couple between a shorter moving average (more sensible to the market signal and consequently faster, although it can produce false signals) and a longer moving average which has a longer lag, although it can produce better trend signals. In this work, the couple was made between an exponential moving average of 5 weeks and one with 20 weeks.

For the Double Crossover method, the rules were defined in Table 3.3.

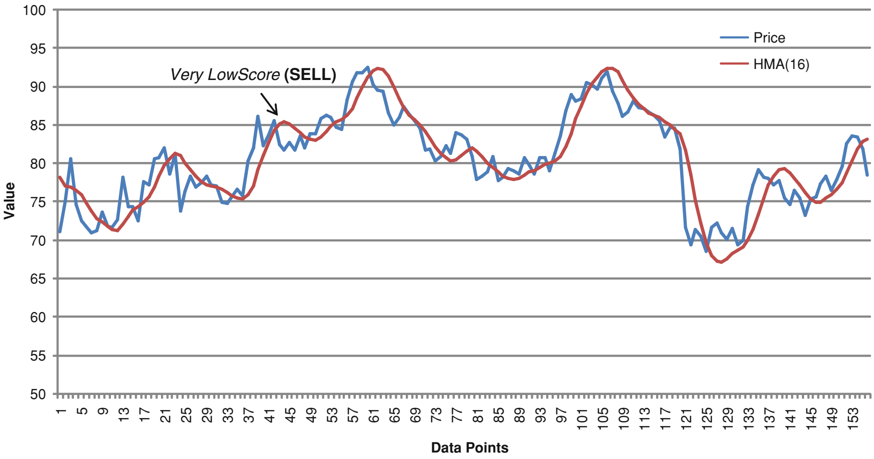


Fig. 3.7 An example of the HMA indicator

Table 3.3 Rules developed for the double crossover method

	EMA(5)–EMA(20)
Very low score	EMA(5) crosses below the EMA(20) line
Low score	Both EMAs are decreasing
High score	Both EMAs are rising
Very high score	EMA(5) crosses above the EMA(20) line

The picture below demonstrates how to apply the double crossover procedure (Fig. 3.8).

3.4.5 *Rate of Change*

The *Rate of Change* (ROC) [7] ratio presents the percentage difference between the current closing price and the price n time periods ago. On doing so it allows us to measure how rapidly the price of a specific stock is moving. If the price is rising or falling too quickly it will probably indicate overbought or oversold conditions. The ROC is described by:

$$ROC_t(n) = \frac{X_t - X_{t-n}}{X_{t-n}} * 100 \tag{3.3}$$

where:

- n is the number of periods considered;
- X_t corresponds to the stock's price on period t.

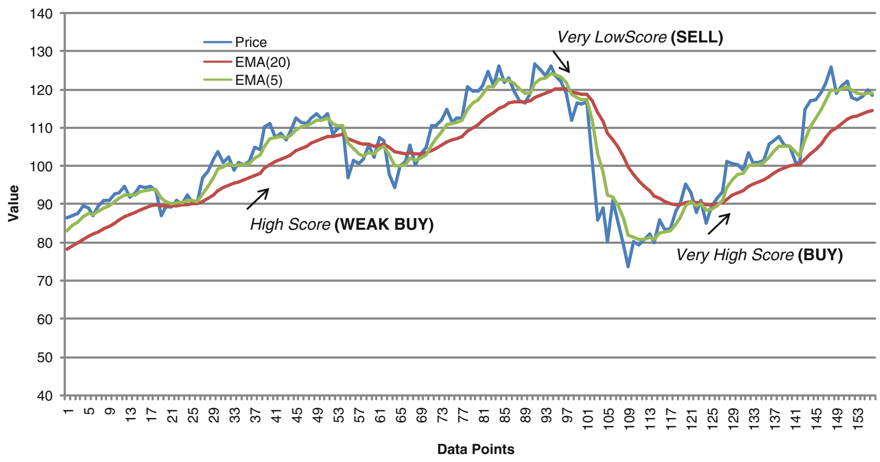


Fig. 3.8 An example of the double crossover method

Table 3.4 Rules developed for the ROC indicator

	ROC(13)
Very low score	ROC line crosses below 0
Low score	Bullish divergence
High score	Bearish divergence
Very high score	ROC line crosses above 0

For the ROC indicator, the rules were defined in Table 3.4.
An example of the ROC indicator application can be seen below (Fig. 3.9).

3.4.6 Relative Strength Index

The *Relative Strength Index* (RSI) [9] indicator is a momentum oscillator used to compare the magnitude of a stock's recent gains to the magnitude of its recent losses, in order to determine overbought or oversold conditions. The RSI is described by:

$$RSI_{\mathrm{t}}(n) = 100 - \frac{100}{1 + RS(n)} \tag{3.4}$$

where:

- $RS = \frac{AverageGains}{AverageLosses}$. See [9] for more details;
- t defines the considered period (day, week, or month).

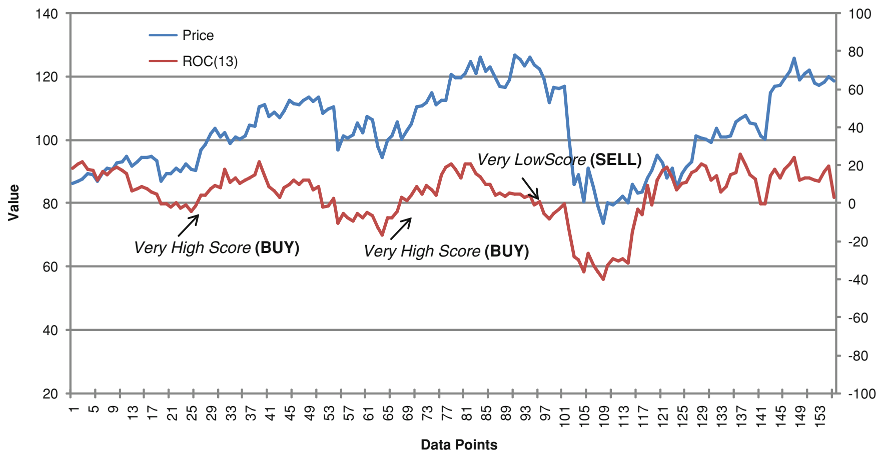


Fig. 3.9 An example on the use of a ROC line

Table 3.5 Rules developed for the RSI indicator

	RSI(14)
Very low score	RSI line crosses below 70
Low score	RSI line is decreasing between the extreme levels
High score	RSI line is rising between the extreme levels
Very high score	RSI line crosses above 30

When calculated, the RSI line forms a signal between 0 and 100, which specifies determined overbought or oversold conditions when its value is above or below specific levels. For the RSI indicator the rules were defined in Table 3.5.

Figure 3.10 applies the RSI formula to a specific market stock price.

3.4.7 Moving Average Convergence Divergence

The *Moving Average Convergence Divergence* (MACD) [7] indicator constitutes one of the most reliable indicators within the market. It is a trend following momentum indicator that exhibits the relation between two distinct moving averages. Essentially, it defines two lines; the MACD line which corresponds to the difference between a 26-week and 12-week EMA and a trigger line which corresponds to an EMA of the MACD line. The difference between the former lines allows us to obtain a histogram which can be easily analyzed and offering us perspectives on price evolution.

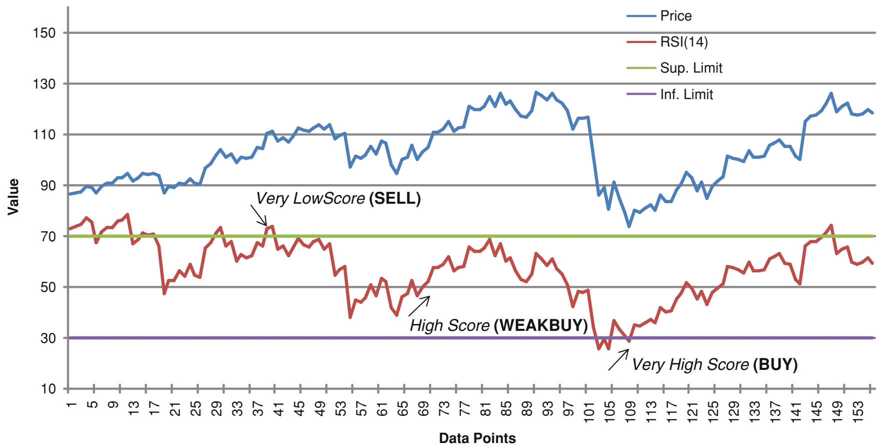


Fig. 3.10 RSI application

Table 3.6 Rules defined for the MACD indicator

	MACD(12,26,9)
Very low score	Histogram crosses below 0
Low score	Histogram is decreasing on negative direction
High score	Histogram is rising on positive direction
Very high score	Histogram crosses above 0

$$\begin{aligned} MACD_t(s,l) &= EMA_t(s) - EMA_t(l) \\ Trigger_t(n) &= EMA_t(n)\ of\ MACD_t(s,l) \\ Hist_t &= MACD_t(s,l) - Trigger_t(n) \end{aligned} \tag{3.5}$$

where:

- n is the number of periods considered for the trigger signal;
- s corresponds to the number of periods considered for the shorter MA;
- l corresponds to the number of periods considered for the longer MA.

For the MACD indicator, the rules were defined in Table 3.6.

The following picture exemplifies the application of the MACD histogram (Fig. 3.11).

3.4.8 On Balance Volume

The *On Balance Volume* (OBV) [9] indicator is a momentum indicator that relates volume with price change. It tries to show if volume is flowing into or out of a

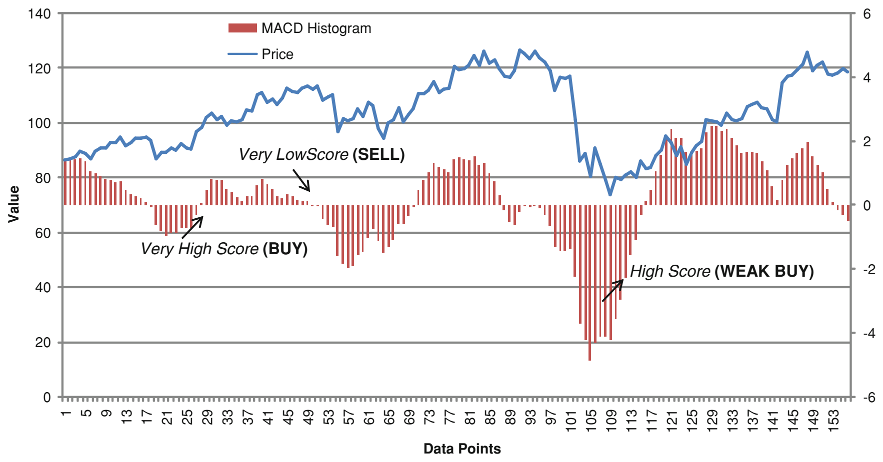


Fig. 3.11 Example of the MACD indicator

Table 3.7 Rules developed for the OBV indicator

	OBV
Very low score	OBV is falling simultaneously with price indicating a clear down trend
Low score	OBV is decreasing and price is rising indicating a possible uptrend breakout
High score	OBV is rising and price is declining indicating a possible downtrend breakout
Very high score	OBV is rising simultaneously with price indicating a clear up trend

security, assuming that volume changes precede price changes. For instance, a rising volume can indicate the presence of smart money flowing into a security preceding its rise on price. The OBV line can be determined as follows:

$$\begin{aligned} IF\ X_t > X_{t-1} &\rightarrow OBV_t = OBV_{t-1} + V_t \\ IF\ X_t < X_{t-1} &\rightarrow OBV_t = OBV_{t-1} - V_t \\ IF\ X_t = X_{t-1} &\rightarrow OBV_t = OBV_{t-1} \end{aligned} \tag{3.6}$$

where:

- X_t corresponds to the stock's price on period t;
- V_t is the volume referent to period t.

The OBV always takes a direction, a rising OBV line indicates that the volume is heavier on up days, confirming a possible up trend.

For the OBV indicator, the rules were defined in Table 3.7.

The following picture exemplifies the application of the OBV indicator (Fig. 3.12).

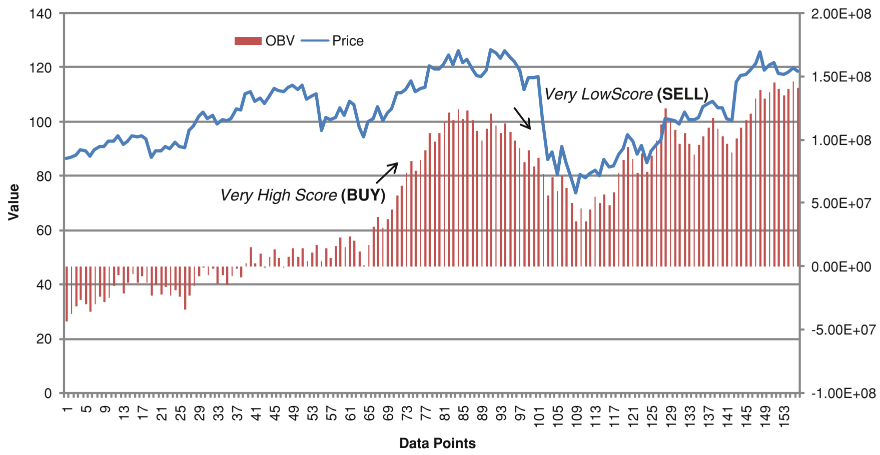


Fig. 3.12 Example of the OBV indicator

3.4.9 True Strength Index

The True Strength Index (TSI) [10] is a momentum-based indicator which tries to determine both trend and overbought or oversold conditions. In order to accomplish these features, the TSI corresponds to a 1-day/week/month momentum which is double smoothed with two moving averages to show the trend and specifying, at the same time, the overbought and oversold conditions. The TSI is described by:

$$\begin{aligned} MNT &= X_t - X_{t-1} \\ TSI_t(r,s) &= 100 * \frac{EMA(s)\ of\ (EMA(r)\ of\ MNT)}{EMA(s)\ of\ (EMA(r)\ of\ |MNT|)} \\ Trigger_t(n) &= SMA_t(n)\ of\ TSI_t(r,s) \end{aligned} \tag{3.7}$$

where:

- X_t corresponds to the stock's price on period t;
- MNT corresponds to the momentum line which calculates the difference between the current price and the price observed on the previous period;
- r corresponds to the number of periods considered for the first EMA;
- s corresponds to the number of periods considered for the second EMA;
- n corresponds to the number of periods considered for the trigger line.

The rules in Table 3.8 were defined for the former indicator.

Figure 3.13 exemplifies the application of the TSI indicator.

Table 3.8 Rules specified for the TSI indicator

	TSI(25,13,7)
Very low score	TSI crosses below trigger on overbought region (25)
Low score	TSI is declining between the −25 and 25 levels
High score	TSI is rising between the −25 and 25 levels
Very high score	TSI crosses below trigger on oversold region (−25)

3.5 Optimization Module

This module corresponds to the main core of the developed application. It is responsible for defining the optimizer techniques and correspondent representation in order to result on a classifier system capable of defining models to score the different assets within the market. As stated before under the previous chapter, the GA optimization technique was chosen after the analysis performed. Since a GA is composed by several components, this section will start by describing how each component of the algorithm was defined.

3.5.1 Chromosome Representation

Starting with the chromosome representation, an individual in the population is represented by a real valued array structure where each element corresponds to the weight, importance given to a specific technical rule within the classifier equation. Besides the described weights, assigned to each technical rule, four bound values are also employed to define the necessary score that an equity needs to obtain, so, it can adopt a long or a short position within the portfolio, or to close the former position. In order to get a better understanding on the considered representation, Table 3.9 is presented.

As can be observed from the previous table, each rule has a specific weight within the classifier model. The classifier is given by the following equation:

$$\begin{gathered}\sum_{i=0}^{N} W_i \cdot Score(X,i) \\ 0 \le W_i \le 1 \\ 0 \le \sum_{i=0}^{N} W_i \le 1\end{gathered} \tag{3.8}$$

where:

- *Wi* is the weight or relevance assigned to the technical rule *i*;
- *Score(X, i)* corresponds to the score given by the technical rule *i* to stock *X*.

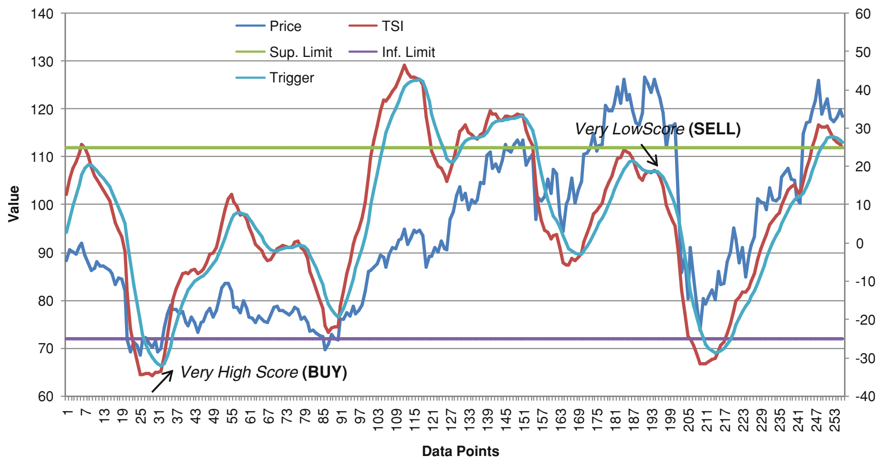


Fig. 3.13 Example of the TSI indicator

Table 3.9 Chromosome representation

1st Rule	2nd Rule	…	Last rule	Buy limit	Short limit	Close buy position	Close short position
[0, 1]	[0, 1]	…	[0, 1]	[0, 1]	[−1, 0]	[−1, 1]	[−1, 1]

After the optimization performed by the algorithm, resulting on a classifier equation, where a set of technical indicators are correctly balanced, all the assets within the market are classified. The stocks whose classification is higher than the value given by the *Buy Limit* field adopt long positions. The ones whose classification is below the *Short Limit* adopt short positions. The last two bound values; *Close Buy Position* and *Close Short Position* determine the necessary score to achieve so a specific position in the portfolio can be closed. Notice, however, that more conditions need to be fulfilled for a specific position within the portfolio can be closed.

3.5.2 Selection

After defining the encoded representation it is necessary to specify how the algorithm will choose the individuals that will generate offsprings for the next generation. This process is performed via a *Truncation Selection* [11] methodology which mainly consists on sorting the population according to their fitness, and subsequently, selecting the best individuals for reproduction. From the set of best individuals a roulette procedure is applied, in order to choose the breeders.

The number of considered parents in given by the *Trunc Threshold* parameter, which is set to be half of the population, by default.

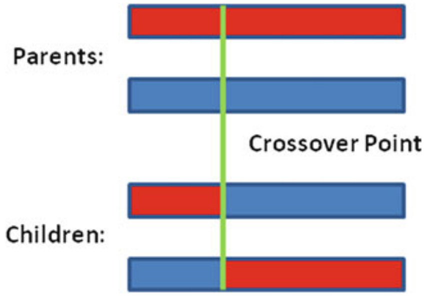


Fig. 3.14 One-cut point crossover

3.5.3 Mutation

In respect to the mutation procedure, a new random value is generated for each variable selected for mutation. The number of variables to be mutated depends on the value given to the *Mutation Rate* parameter, the chromosome size, and the number of population individuals as you can see below:

$$Mutations = Mutation\,Rate * Chromosome\,Size * (Population\,Size - 1) \quad (3.9)$$

As you can observe from the previous equation, the number of mutations largely depends on the number of total variables considered by the algorithm. Notice, however, that one single individual was discarded, as you can see from the minus one within the equation. The purpose of this restriction is to maintain the best individual in the current population, in each generation of the algorithm. This technique is normally referred as *Elitism*. Other mutation procedures were experimented such as the *Insert Mutation* [11] technique. However, the convergence process was worse when compared with the standard mutation operator.

3.5.4 Crossover

Considering the crossover operator, different types of crossover operators were implemented, in particular, the *Single Arithmetic Recombination* [12], the *Whole Arithmetic Recombination* [12], and the *One-Cut Point Crossover* [12] method, contemplating the generation of two offsprings. After performing a rigorous testing on the algorithm convergence, it was concluded that the one-cut point methodology allowed us to obtain the best results for the represented chromosome. As you can see from Fig. 3.14, this process is extremely trivial and consists on randomly selecting a crossover point and from that point all variables are swapped between both parents, giving birth to two new individuals.

3.5.5 Initial Generation

In order to avoid the initial generation of infeasible individuals when the initial population is randomly generated, some considerations were necessary:

- The sum of all weights assigned to the technical rules cannot be higher than one. In order to avoid it, each weight is normalized, as follows:

$$w_i = \frac{w_i}{\sum_{i=0}^{N} w_i} \tag{3.10}$$

- All the remaining variables need to obey to certain restrictions. The *Buy Limit* must be superior to the *Sell Limit*. The *Close Buy Position* value must be less than the *Close Short Position* value. In order to comply with the former restrictions, the values are swapped in case of violation.

3.5.6 Constraints Handling

One of the major problems presented by the defined chromosome concentrates on the restrictions over the different weights assigned to the stipulated technical rules.

A trivial way on handling an inequality constraint such as the former one consists on applying a death penalty function [13], discarding infeasible individuals within the population. Although it seems an extremely basic approach, this methodology has a major problem of not exploring any information from the infeasible individuals, in order to guide the search more effectively. To surpass this problem, a simulated artificial immune system [14] was employed, which provides an efficient way of guiding the search, taking into account the information generated by the infeasible individuals. Besides the fact of being easy to implement, this strategy is also very effective on the proposed goal of exploring information gathered by the non-feasible genes. Very generically, the algorithm maintains in each generation a population of infeasible individuals designated as antibodies which suffers the same kind of evolution of the main population. However, the evaluation function is much easier which allows us to rapidly execute the convergence process within this smaller population. This convergence procedure corresponds to the process of executing a genetic algorithm inside the main genetic algorithm. The principle behind this algorithm corresponds to the *Negative Selection Model* which tries to capture the behavior of the human immune system on knowing what is really part of the human system, and what is not. To get the complete algorithm description, the reader is referred to [14].

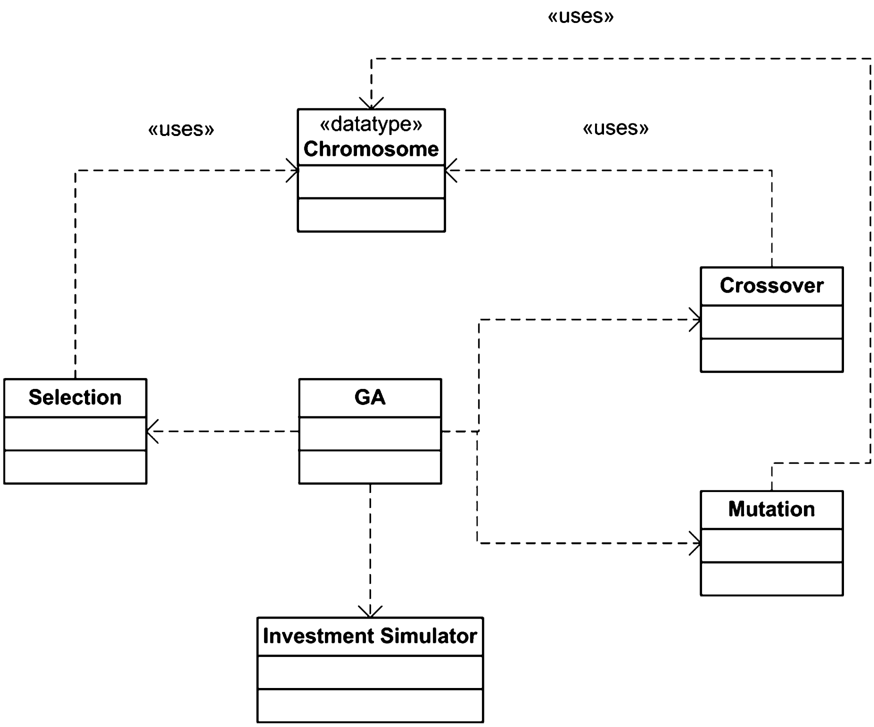


Fig. 3.15 Optimization module implementation

3.5.7 Evaluation Function

In order to evaluate each individual within the population, so the algorithm can pick the best ones for reproduction, and consequently, converge on an optimal solution, the *Return On Investment* (ROI) function was applied. The ROI is used to evaluate the efficiency of different investments during a specific period of time.

As you can see, a simple objective was considered for evaluating each solution, i.e., the goal of the algorithm is to maximize the ROI. However, the solution could be easily extended with a multi-objective consideration, where the goal was to optimize simultaneously two conflicting objectives; the ROI and the risk involved, which could be measured by the volatility of returns, for instance.

3.5.8 Optimization Module Implementation

In respect to the implementation, the module was implemented with the following composition (Fig. 3.15):

As you can observe from Fig. 3.15, the *Optimization Module* is defined by five classes and one simple structure:

- *Chromosome Structure*. This structure defines the chromosome, i.e., the real variables presented within the genome, the respective fitness, and one simple boolean value which is merely used to optimize the algorithm in order to avoid fitness recalculation;
- *Selection Class*. Responsible for defining the selection operator;
- *Crossover Class*. Accountable for specifying a set of crossover operators which can be used by the genetic algorithm;
- *Mutation Class*. Defines different mutation operators;
- *Investment Simulator Class*. The purpose of this class corresponds to the fitness calculation;
- *GA Class*. Class responsible for defining the execution procedure of the genetic algorithm, handling the population of chromosomes and the respective constraint handling mechanism.

3.6 Investment Simulator Module

In order to evaluate each individual, an investment simulator is necessary for generating a portfolio according to the classifier equation, and managing it through time. This management module is used by the genetic algorithm, in order to classify each chromosome and performing test/real-life simulations. To get a realistic experience on managing the portfolio, several inputs are considered:

- *Budget*. The capital available to invest;
- *Max Size*. The maximum number of assets included on the desired portfolio;
- *Short Selling*. This parameter is used for specifying if short selling is allowed or the user just want to adopt long positions;
- *Transaction Costs*. Used for the consideration of transaction costs. This parameter is used to include the commission costs involved on buying or selling shares.

Before explaining in detail the behavior of such module, the structures used to implement the considered module will be defined.

3.6.1 Implementation and Functionality

The *IS* module mainly works with five classes, as you can notice under Fig. 3.16.

- *Stock Class*. Stores all data relative to each stock within the market;
- *Portfolio Class*. Responsible for defining a specific portfolio, calculates the respective shares, calculate its value, update its composition, among other functions;

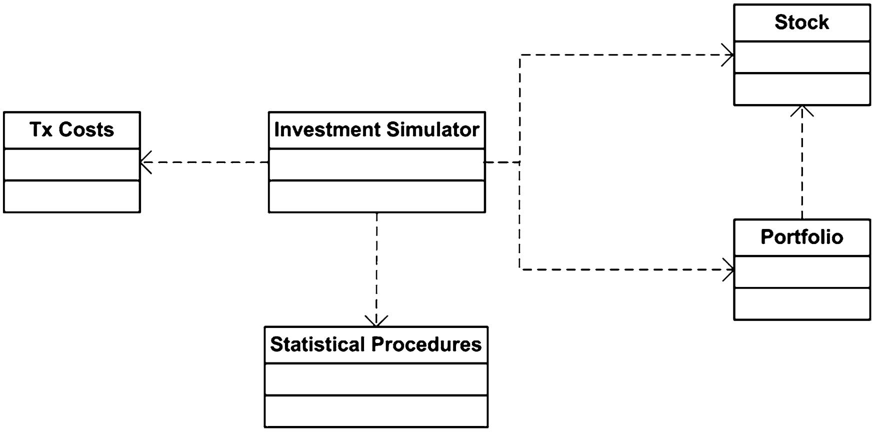


Fig. 3.16 IS class dependencies

- *Tx Costs Class.* Specifies the transaction costs involved in each considered market;
- *Statistical Class.* Defines several statistical functions. Those functions are used to evaluate the portfolio through time and for testing purposes;
- *Investment Simulator Class.* Accountable for managing the portfolio trough time, adding or closing new positions, according to the strategy employed.

There are several specifications that need to be concretely defined over this *Investment Simulator* module. As already stated the IS will use a specific equation to classify the assets within the market.

The complete management process is the following:

- The first step consists on applying 50 % of the available budget on generating the initial portfolio using the equation given by the algorithm;
- In each new week, during the period of validation or training, the portfolio is updated using the following rules:
  - If there are positions in the portfolio presenting a loss of 10 % or higher, the current position is immediately closed. This condition is an insurance to avoid an unexpected crash on the company;
  - If there is a position which presents a score indicating a possible close and it has already given profit, the position is closed;
  - If there are stocks in the market who present a classification possible to add, and the portfolio has not achieved its maximum size, new positions are formed within the portfolio. 50 % of the budget is used for considering these new positions.

This complete process can be briefly summarized with Fig 3.17.

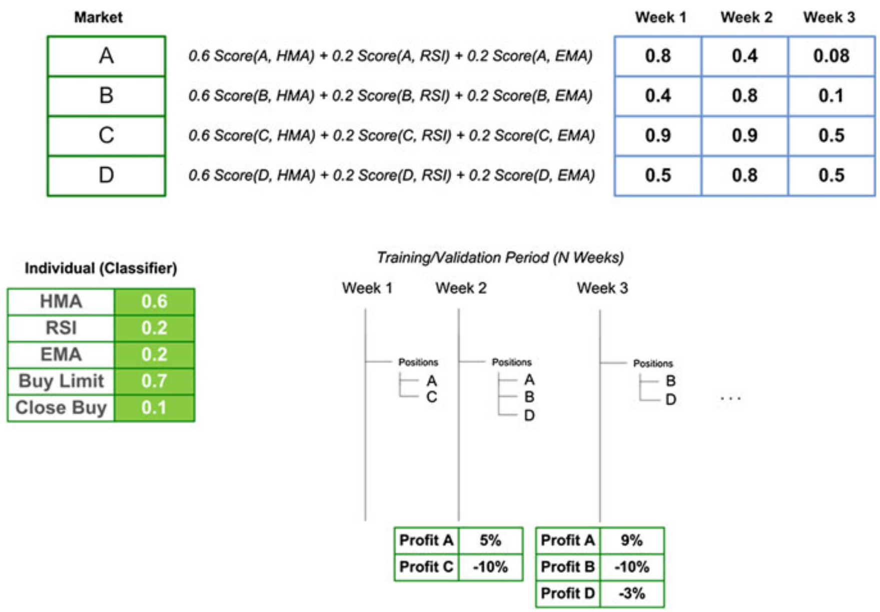


Fig. 3.17 Simple IS example

3.7 Conclusions

Within this chapter the reader had an insight on the system's designed architecture. Notice that several aspects can be easily modified and extended in this developed solution, in particular the considered technical indicators and their period parameters, the time span used on the portfolio update, the value given to the stop orders, and several details regarding the operators used on the GA. The current solution can be thoroughly tested, in order to generate an ideal combination of the previous parameters.

References

1. Down Jones Indexes. http://www.djaverages.com. Accessed June 2009
2. Standard and Poor's. http://www2.standardpoors.com. Accessed June 2009
3. Yahoo! Inc. http://finance.yahoo.com. Accessed June 2009
4. Férnandez-Blanco P, Bodas-Sagi D, Hidalgo JI (2008) Technical market indicators optimization using evolutionary algorithms. Genetic and evolutionary computation conference (GECCO), 1851–1858, Atlanta, USA
5. Bodas-Sagi D, Fernández P, Hidalgo JI, Soltero F, Risco-Martín J (2009) Multiobjective optimization of technical market indicators. Genetic and Evolutionary Computation Conference (GECCO), 1999–2004, Montreal, Canada
6. Banco Best. Best Trading Pro. http://bancobest.iitech.dk/PT/. Accessed 12 Sept 2009

7. Murphy JJ (1999) Technical analysis of the financial markets: a comprehensive guide to trading methods and applications. Prentice Hall Press, New York
8. Allan Hull (2004) Hull moving average. Just Data J
9. Achelis SB (2000) Technical analysis from A to Z. McGraw-Hill, New York
10. Hartle T (2002) The true strength index. Adv strategies
11. Sumathi S, Hamsapriya T, Surekha P (2008) Evolutionary intelligence: an introduction to theory and applications with Matlab. Springer, Berlin
12. Eiben AE, Smith JE (2007) Introduction to Evolutionary Computing. Springer, New York
13. Coello CAC (1999) A survey of constraint handling techniques use with evolutionary algorithms. Technical report, Laboratorio Nacional de Informática Avanzada, Xalapa, Veracruz
14. Coello CAC, Cortés NC (2001) Use of emulations of the immune system to handle constraints in evolutionary algorithms. Intell Eng Systems through Artif Neural Netw 11:141–146

Chapter 4
System Validation

Abstract This chapter describes the validation approach used to evaluate the defined system, in particular the employment of the Backtesting [1] test strategy. This process consists on testing a specific strategy on a prior time period in order to determine its effectiveness. For instance, suppose an evaluation of our strategy, in terms of its year performance, is required. Instead of waiting one whole year to do it, the past data can be extracted and, subsequently, the procedure evaluated on the considered periods. Applying the developed strategy to prior data can be substantially benefic, in order to detect strategy flaws and improve its potential.

4.1 Performance Measures

To calculate the performance of a financial fund or portfolio, distinct measures can be employed, such as ROI, *Sharpe Ratio*, *Compound Monthly Growth Rate*, *Treynor Ratio*, among others. In this section, three of these measures are addressed; the ROI, the *Sharpe Ratio*, and the *Sortino Ratio* due to its vast recognition and application by computer scientists who handle the portfolio problematic. Also, a list of classification parameters is proposed for evaluation.

4.1.1 Return on Investment

The *Return on Investment* (ROI) is used to evaluate the efficiency of different investments during a specific period of time. The standard formula is extremely simple and can be defined as following:

$$ROI = \frac{Gain - Cost}{Cost} \tag{4.1}$$

A. M. S. B. S. Gorgulho et al., *Intelligent Financial Portfolio Composition Based on Evolutionary Computation Strategies*, SpringerBriefs in Computational Intelligence, DOI: 10.1007/978-3-642-32989-0_4,

Table 4.1 An example of a portfolio composition

Stock	Purchase Price (PP)	Current Price (CP)	Shares (Sh)
A	20 €	25 €	10
B	10 €	8 €	10
C	30 €	50 €	15
D	50 €	60 €	4

This expression can be easily adapted to different problems or situations. The following example, see Table 4.1, provides a process of applying the formula over a specific portfolio in order to evaluate its efficiency. Given this portfolio, the goal is to calculate the ROI, one month after its composition.

The initial value of the portfolio PV_0 and its current value PV_1 can be calculated as shown by (4.2) and (4.3).

$$PV_0 = \sum_{i=1}^{4} PP_i * Sh_i = 20 * 10 + 10 * 10 + 30 * 15 + 50 * 4 = 950\ € \tag{4.2}$$

$$PV_1 = \sum_{i=1}^{4} CP_i * Sh_i = 25 * 10 + 8 * 10 + 30 * 15 + 50 * 4 = 1320\ € \tag{4.3}$$

Given the portfolio's value on the beginning (PV_0) and end of the desired month (PV_1), the ROI can be determined as the following:

$$ROI = \frac{PV_1 - PV_o}{PV_0} = \frac{1320 - 950}{950} = 39\,\% \tag{4.4}$$

4.1.2 Sharpe Ratio

The *Sharpe Ratio* [2] is one of the most applied performance measures when evaluating a portfolio. This measure is adapted for well diversified portfolios because it also takes into account the risk of the portfolio besides its return. This ratio describes the excess return that an investor is receiving for holding a portfolio with a specific risk which allows us to understand if the portfolio's return is due to a smart investment decision, or if it is just a result of choosing a higher level of risk. The common formula can be defined as the following:

$$S_P = \frac{R_P - R_f}{\sigma_P} \tag{4.5}$$

where:

- P corresponds to the holding portfolio;
- R_P is the average return of the portfolio P (Consult Appendix A);

- R_f consists on the risk-free rate, i.e. the best available rate of return of a free-risk asset. This measure is used to guarantee if the investor is being properly compensated for the additional risk he is taking on with the risky asset. Normally, the return of security with least volatility in the market is used, such as the U.S Treasury bills or *Euribor* rates;
- σ_P corresponds to the standard deviation of portfolio's returns, which indicates the risk associated to the portfolio as expressed by Markowitz's pioneer work (Consult Appendix A).

The three measures used in the formula can be of any frequency; daily, weekly, monthly or annually.

The main idea of the ratio consists on calculate the additional return of the portfolio for holding risky assets over a risk-free security. Suppose an investor X holds a portfolio with average monthly returns of 11 % while an investor Y gets 9 %. In respect to the standard deviation, X gets 7 % and manager Y notes 5 %. Choosing a risk-free rate of 3 %, the Sharpe Ratio for each investor would be the following:

$$S_X = \frac{11\% - 3\%}{7\%} = 1.14 \quad S_Y = \frac{9\% - 3\%}{4\%} = 1.50 \tag{4.6}$$

Although manager X got a better average return, he followed larger risks than investor Y. According to the values obtained, investor Y has a better risk-adjusted return for its portfolio.

4.1.3 Sortino Ratio

The *Sortino Ratio* [3] is a risk measure similar to the *Sharpe Ratio*. However, in contrast with its predecessor, instead of considering the volatility on all returns, it only includes the negative performances so it can provide a more realistic risk measure. Its formula is similar with the one provided by Eq. 4.2. However, as stated before, the denominator only corresponds to the standard deviation values observed during negative performances.

4.2 Classification Parameters

Besides the three presented measures used to evaluate the performance of the generated portfolios, it is possible to define a list of important parameters which can be used to classify the designed strategy used to compose those portfolios:

- *Number of positions*. The number of positions taken in the portfolio within a specific period;
- *Percentage of profitable positions*. From the total number of positions made in a specific period, it will be important to observe how many of them are profitable;

- *Percentage of non-profitable positions*. As to measure the number of winning positions, it is also desirable to determine the non-profitable ones;
- *Greatest profit*. It will be interesting to observe what was the greatest profit obtained within all the securities that compose the portfolio and understand why;
- *Greatest loss*. As the greatest profit, it is also important to determine the greatest loss;
- *Average Profit*. Indicates the average profit considering all positions taken within a specific period.

4.3 Strategies Employed

In order to validate the designed solution, the developed strategy was compared against the market and three other investment methodologies:

- *Buy and Hold*. According to some theories [4], already addressed in Chap. 1, prices are independent to each other, meaning that one cannot use past data to forecast market development, so the best strategy that can be employed is buy and hold on which a specific set of assets is maintained regardless of market fluctuations. The major question on implementing this strategy is in which assets should the investor concentrate to form the initial portfolio? Normally, experienced investors perform a fundamental evaluation of several companies and then compose their portfolio. Since this form of data was not considered by the application, the adopted buy and hold strategy picks the assets which presented best average returns during the preceding year;
- *Random*. The random strategy implemented adopts a purely random behavior; each new week the portfolio is updated by closing random positions, and picking new random assets from the market, to add to the already existent portfolio. Both long and short positions are considered;
- *Momentum*. This strategy divides the portfolio on an equal number of long and short positions. The assets which exhibit best arithmetic mean returns during the former six months adopt long positions. The ones which present worst performances take short positions. The portfolio is then maintained over three months. After those three months, new positions are taken according to the former process. The authors on [5] demonstrate the feasibility of momentum strategies.

In respect to the market's returns, the index returns observed during the testing periods were used. Notice, however, that the comparison against the index is not fair due to some kind of selection bias. The market index is being constantly updated; some companies bankrupt being immediately discarded, others are replaced since they cannot comply with the index restrictions. In the presented work, some of the discarded companies are still considered due to limitations on the application. It is very hard to maintain an automatic process which is constantly replacing the financial data used, being able to perform frequent index reconstructions.

Table 4.2 Case study I—configuration

Parameter	Value
Market	All stocks from DJI
Period	01/01/03–31/06/09
Budget	100 000 USD
Maximum portfolio size	10
Short selling?	TRUE
Commission costs	0.02/Share Minimum Fee: 14.00 USD
Number of executions	100

Table 4.3 Case study I—evolutionary configuration

Parameter	Value
Population size	64
Mutation rate (%)	10
Number generations	350
Trunc threshold (%)	50
Sliding window [a]	6 months/6 months

[a] *Sliding Window* refers to the training/validation period combination employed on the evaluation. For instance, if the validation starts on January of 2003, the previous six months are used to train the algorithm. After six months of validation the algorithm passes through the same training process

4.4 Case Studies

Under this section, several case studies are described to evaluate the developed solution.

4.4.1 Case Study I—DJI Between 2003 and 2009

The presented case study exhibits the results obtained when evaluating the implemented strategies during the years of 2003–2009. For the designed case study, the configuration described in Table 4.2 was applied, particularly, all stocks from the DJI were considered for performance analysis and for picking the most promising for the portfolio composition. The maximum portfolio size is parameterized and was here set to ten, additionally, both long and short positions were allowed including transaction costs and, finally, the process was executed for 100 times in order to allow a proper statistically analysis.

In respect to the evolutionary strategy, the GA kernel was set with the parameters described in Table 4.3.

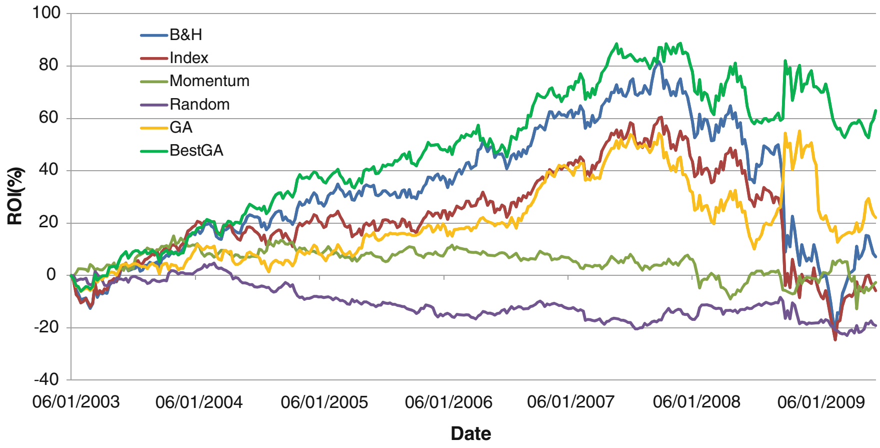


Fig. 4.1 Case study I—ROI evolution

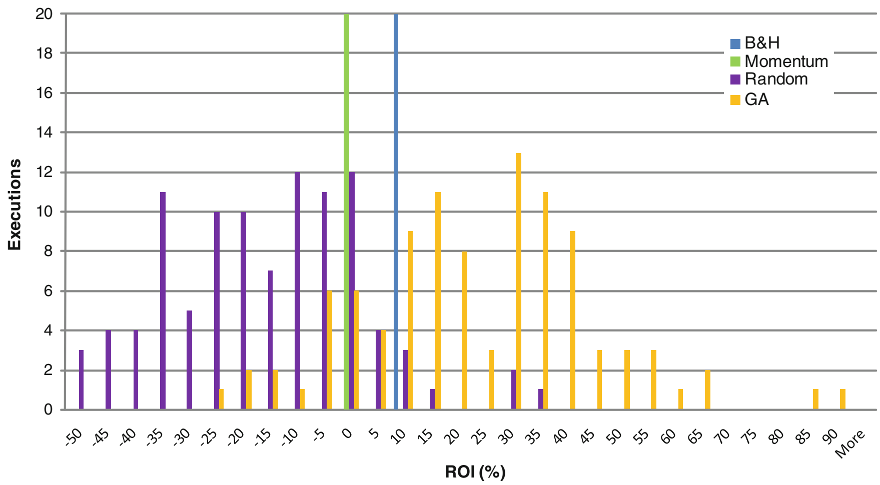


Fig. 4.2 Case study I—ROI distribution

4.4.1.1 Return on Investment

The following graph (Fig 4.1) exhibits the results obtained for the considered strategies within the years of 2003–2009, for the B&H, the Random, the proposed approach (GA) and the Best GA iteration. Each curve represents the return on investment achieved by the respective investment methodology.

To highlight the superiority of the evolutionary strategy, on the end of the testing period, the following histogram (Fig. 4.2) demonstrates the ROI obtained for the different 100 executions experimented per each investment tactic.

Table 4.4 Case study I—classification parameters

Parameter	Buy hold	Momentum	Random	GA	Best GA execution
ROI (%)	7.17	−2.63	[−21.97, −14.77]	[16.68, 25.29]	62.95
Sharpe ratio	0.08	−0.22	−0.93	0.21	0.67
Sortino ratio	0.07	−0.50	−1.25	0.40	21.03
Positions	10	260	[1371, 1389]	[151, 159]	156
Profitable positions (%)	60.00	48.08	[46.83, 47.55]	[80.24, 81.50]	88.46
Non profit. positions (%)	40.00	51.92	[52.45, 53.17]	[18.50, 19.76]	11.54
Avg. Profit position (%)	7.18	0.94	[−0.16, −0.07]	[1.93, 2.53]	4.00
Max. Profit (%)	75.28	61.61	[63.04, 78.21]	[104.69, 136.57]	59.66
Min. Profit (%)	−53.01	−44.37	[−42.96, −39.03]	[−36.46, −34.94]	−30.28

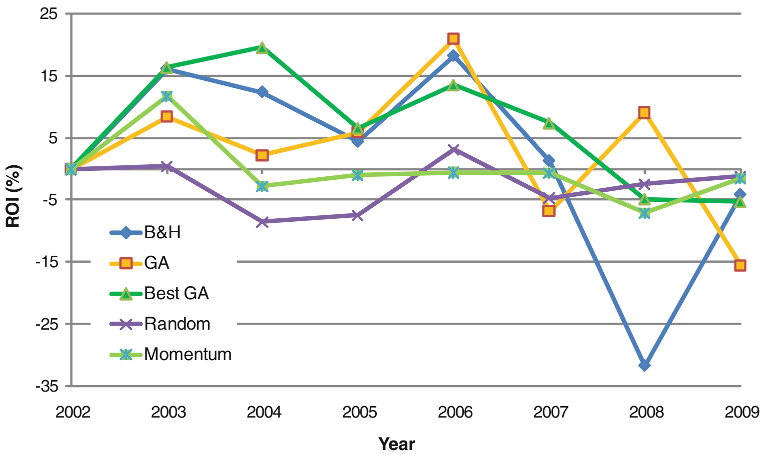


Fig. 4.3 Case study I—year ROI

4.4.1.2 Classification Parameters and Performance Measures

Table 4.4 shows the performance of each strategy according to the parameters described on the first section of this chapter. Notice that for the Random and the Evolutionary strategy, 100 different executions were experimented to thoroughly evaluate each methodology. The results for those strategies correspond to the confidence interval achieved when using a confidence degree of 95 %.

Both; the *Sortino* and the *Sharpe Ratio* were calculated using year returns. To get a general idea, observe the graph in Fig. 4.3; indicating the ROI achieved per year by each of the considered strategies.

In respect to the training evolution, in Fig. 4.4 the reader can observe the fitness evolution obtained for best individual in the algorithm population, during a six month period training for a particular execution of the evolutionary strategy.

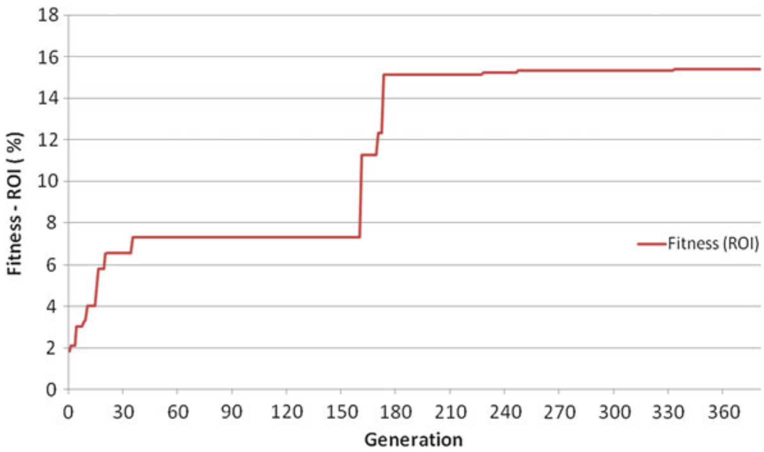


Fig. 4.4 Algorithm convergence

Table 4.5 Example of the resulting chromosome composition—part I

MACD	HMA	EMA	ROC	RSI	DC	TSI	OBV
0.28	0.03	0.01	0.05	0.26	0.01	0.21	0.15

Table 4.6 Example of the resulting chromosome composition—part II

Buy limit	0.78
Short limit	−0.75
Close buy limit	−0.36
Close short limit	0.30

As example, the reader can also observe an example of the best obtained chromosome within a training period during a particular algorithm run (Tables 4.5 and 4.6).

4.4.1.3 Discussion

From the ROI evolution chart (Fig. 4.1) one can observe that the average obtained on 100 executions of the evolutionary strategy is not capable of surpassing the index and the B&H approach during the bull market period. However, when the market crashes the evolutionary strategy is capable of maintaining a reasonable profit without collapsing as the competitor strategies; being less risky and volatile, and subsequently obtaining a much higher ROI on the end of the testing period. The question here is if the developed strategy is capable of beating the market during the crash why the same advantage is not verified during the great bull market from 2003 until the beginning of 2008?

Table 4.7 Case study II—configuration

Parameter	Value
Market	150 stocks from S&P500
Period	01/01/06–31/06/09
Budget	100 000 USD
Maximum size portfolio	10
Short selling?	TRUE
Commission costs	0.02/Share Minimum fee: 14.00 USD
Number of executions	100

Table 4.8 Case study II—evolutionary configuration

Parameter	Value
Population size	32
Mutation rate (%)	10
Number generations	350
Trunc threshold (%)	50
Sliding window	6 Months/6 Months

Although the evolutionary strategy tries to perform intelligent investment decisions it has much more transaction costs when compared with the B&H approach, decreasing its profitability when the market is rising. In contrast, when the crash occurs, the intelligent investment decisions are more notorious which allow the strategy to pick the ideal assets while maintaining its profit.

In respect to the strategies volatility, from Fig. 4.3 it is possible to perceive that the genetic approach is much less risky and volatile than B&H, generating higher *Sharpe* and *Sortino* ratio values. Although the random and the momentum strategies produced a more constant return level, both strategies generate negative risk values, which oblige the investor to pick a risk-free asset such as a government T-Bill. Moreover, the number of positions with positive return exceeds 80 %, for the GA, confirming again the high confidence level of the proposed approach.

Notice also that the best genetic execution, given by the green line, is much more stable and profitable than any other curve presented in the graph. In order to approximate the average obtained to this ideal curve, it will be necessary to increase the robustness and efficiency of the developed GA.

4.4.2 Case study II—SP Between 2006 and 2009

The presented case study exhibits the results obtained when evaluating the implemented strategies during the years of 2006–2009. For the designed case study, the following configuration was applied (Table 4.7):

In respect to the evolutionary strategy, the GA kernel was set with the parameters described in the following table (Table 4.8):

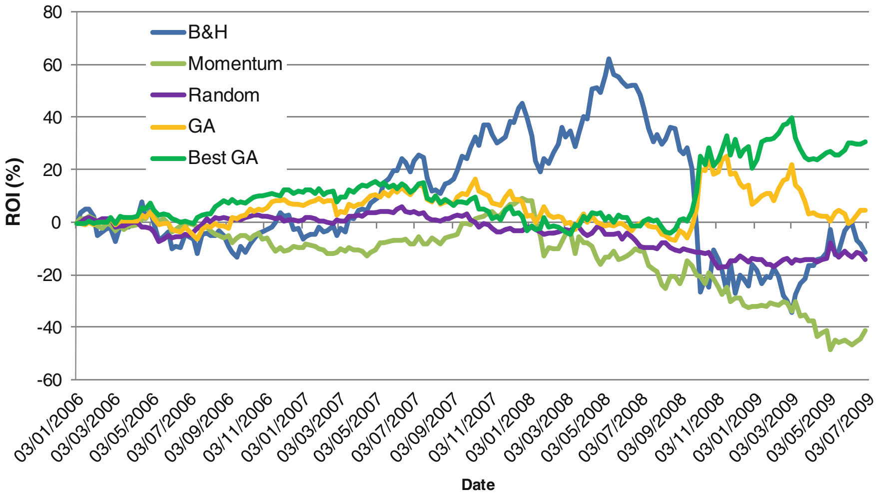


Fig. 4.5 Case study II—ROI evolution

4.4.2.1 Return on Investment

The following graph (Fig. 4.5) exhibits the results obtained for the considered strategies within the years of 2006–2009. Each curve represents the return on investment achieved by the respective investment methodology.

To highlight the superiority of the evolutionary strategy, on the end of the testing period, the following histogram (Fig. 4.6) demonstrates the ROI obtained for the different 100 executions experimented per each investment methodology.

4.4.2.2 Classification Parameters

Table 4.9 exhibits the performance of each strategy according to the parameters described on the first section of this chapter. Notice that for the Random and the Evolutionary strategy, 100 different executions were experimented to thoroughly evaluate each methodology. The results for those strategies correspond to the confidence interval achieved when using a confidence degree of 95 %.

4.4.2.3 Discussion

Similar to the previous case study; the evolutionary strategy is capable to surpass the opponent methodologies during the crash. It is also possible to perceive that when the market is running sideways (January 2006 to May 2007) the genetic approach is able to maintain a reasonable profit in contrast with the remaining strategies. However, when the market suddenly rises the B&H surpasses the

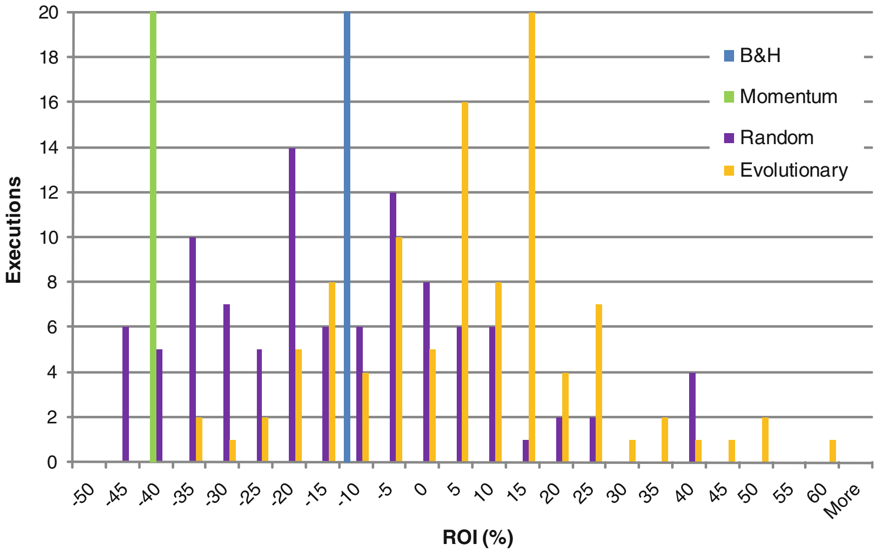


Fig. 4.6 Case study II—ROI histogram

Table 4.9 Case study II—classification parameters

Parameter	Buy hold	Momentum	Random	GA	Best GA execution
ROI (%)	−11.41	−41.08	[−19.60, −11.32]	[0.21, 7.35]	30.84
Positions	10	140	[768, 780]	[106, 117]	101
Profitable positions (%)	20.00	46.43	[46.50, 47.40]	[75.62, 77.60]	78.22
Non profit. Positions (%)	80.00	53.57	[52.50, 53.50]	[22.40, 24.38]	21.78
Avg. Profit position (%)	−11.41	1.65	[−0.18, 0.00]	[2.19, 3.29]	6.80
Max. Profit (%)	83.12	487.29	[60.14, 79.80]	[103.63, 140.74]	140.32
Min. Profit (%)	−81.38	−61.31	[−47.33, 42.44]	[−35.98, −33.25]	−32.84

competitor strategies, and the genetic procedure is not capable of maintain its return levels, starting to slightly decrease, and rising once more when the financial crash occurs.

4.5 Conclusions

Within this chapter the reader had the opportunity to observe the results achieved by the developed solution during the years of 2003 to the first semester of 2009 for the DJI index, and the period of 2006 to the first semester of 2009 for the S&P500

index. Although the designed algorithm was not capable of beating the B&H procedure when the market is rapidly rising, one can easily see that when market crashes or runs sideways, the genetic approach offers a much more powerful and robust solution for these conditions, even when considering the average of 100 distinct executions. For instance, in the ROI evolution figure, obtained for the first case study (Fig. 4.1), the best GA run presents extremely good results on all market conditions. The ideal will be to improve the algorithm robustness as well the respective parameterization in order to approximate its ROI levels as the ones exhibited by this single genetic execution.

References

1. Ni J, Zhang C (2005) An efficient implementation of the backtesting of trading strategies. In: Proceedings of third international conference on parallel distribution process application, Berlin, pp 126–131
2. Sharpe WF (1994) The sharpe ratio. J Portfol Manage 21:49–58
3. Sortino FA, Meer RVD (1991) Downside risk. J Portfol Manage 17:27–31
4. Malkiel B (1973) A random walk down wall street. W. W. Norton and Company, New York
5. Rey DM, Schmid MM (2007) Feasible momentum strategies: evidence from the Swiss stock market. Financ Mark Portfol Manage 21:325–352

Chapter 5
Conclusions and Future Work

Abstract The presented work proposes a portfolio management system using EAs coupled with technical analysis indicators for picking the most promising stocks in the market. In order to validate the designed application, the current strategy was compared against the market itself and several other investment methods, from 2003 to the first semester of 2009. The preliminary results are promising, and much more can be performed to improve them, e.g., the algorithm can be easily extended and parameterized. The present chapter proposes several features to refine the current solution.

5.1 Conclusions

The validation performed showed that the application of EC coupled with technical analysis indicators has a certain potential. There is a lot to explore on using this kind of approach to the portfolio management problematic. In this work the reader had an insight on several investment methodologies, as well, an overview over different computational techniques to approach the presented problem. Although there is the prospect on using this application to automatically manage your financial portfolio, the human skills cannot be totally replaced, and the best way to use this application is to maintain an ear on financial market news to understand if there is a position provided by the system which will possibly fail due to problems or news affecting that specific company, and which cannot be perceived by the developed system.

A. M. S. B. S. Gorgulho et al., *Intelligent Financial Portfolio Composition Based on Evolutionary Computation Strategies*, SpringerBriefs in Computational Intelligence, DOI: 10.1007/978-3-642-32989-0_5, © The Author(s) 2013

5.2 Future Work

Under this section, several limitations of the algorithm are addressed, accompanied with the respective improvements.

5.2.1 Current Limitations

- Since the application only considers weekly data, there is a capital aspect that could be severely affecting the algorithm's results. When a position loses more than 10 % it is immediately closed. However, if we only consider weekly data, as in this case, that position could have been losing for five or more days ago, which lost could reach to 30 % or more. In order to avoid it, daily data should be considered and each day the stop order should be verified;
- There is no measure of risk involved. One of the most important concepts on portfolio management should be the risk considered by the investor. Different portfolios or strategies should be considered to pick stocks on the market, depending on the level of risk desired by the client;
- The current system is only capable of considering a specific market on each execution. It is important to surpass this point, and allow the application to consider simultaneously stocks from distinct market.

5.2.2 Possible Improvements

Due to the high level of code extensibility, the following features are proposed to improve the current algorithm.

- Extend the chromosome with more technical indicators. This extension can be easily performed, it is only necessary to implement the desired indicator and define the respective rules;
- Use an additional bit vector in the chromosome which defines the inclusion or not of a specific indicator to classify the stocks within the market;
- Extend the *Financial Processing Data Module* by considering Euronext market data;
- Consider parallel processing to speed up the algorithm execution;
- Besides defining the optimal balance between distinct technical indicators, the chromosome can also be extended with the time span assigned to each indicator. However, this process will oblige to a recreation of each technical indicator in each generation to each individual which will highly increase the time execution

of the algorithm. It needs to be fully parallelized in order to become a feasible solution;
- Adopt a multi-objective optimization, by maximizing the ROI and minimizing the involved risk;
- To minimize the risk involved on the developed solution, maintain a data structure containing the scheduled dates for announcing the considered companies' results. This is very importance, since after the profits publication, the company stock value can suffer a sudden change which can take an extremely high or down direction;
- Try to understand which is the best strategy in respect to the training question. Should we use a sliding-window? An enormous set of data? Or should we use a mechanism to detect market behavior and perform the training on distinct market conditions?
- Improve the existing GA by trying additional mutation, crossover and selection procedures. Applying other methods for constraint handling, in particular the Stochastic Ranking [1] mechanism.

References

1. Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. IEEE Trans Evol Comput 4:284–294

Appendices

Appendix A: Markowitz's Model

The Markowitz's model, also known as the standard Mean–Variance (M–V) model can be defined as follows:

A portfolio P is defined as a set of N real values (w_1, w_2, …, w_n) where each w_i represents the percentage of the total investment on P that it is allocated over the asset i. All the weights should be greater than or equal to zero, and their sum is necessarily equal to one. These two restrictions can be represented as the following:

$$\sum_{i=0}^{n} w_i = 1 \wedge 0 \leq w_i \leq 1 \tag{A.1}$$

Each asset has an expected return expressed by R_i. The expected return R_P of the portfolio is given by the sum of the expected returns of all securities that compose P times the respective weight:

$$R_P = \sum_{i=0}^{n} R_i w_i \tag{A.2}$$

Also, a risk measure is associated with each asset, represented by σ_i. The value of this risk is given by the variance of that asset's return over time, being the total risk of the portfolio given by the covariance between each of its assets:

$$\sigma_P = \sum_{i=0}^{n} \sum_{j=0}^{n} \sigma_{ij} w_i w_j \tag{A.3}$$

Defined the necessary measures, one can formulate this model as quadratic programming problem, stated as follows:

$$\min \sigma_P \tag{A.4}$$

A. M. S. B. S. Gorgulho et al., *Intelligent Financial Portfolio Composition Based on Evolutionary Computation Strategies*, SpringerBriefs in Computational Intelligence, DOI: 10.1007/978-3-642-32989-0,

subject to:

$$\sum_{i=0}^{n} R_i w_i = R_P$$

$$\sum_{i=0}^{n} w_i = 1$$

$$0 \leq w_i \leq 1$$

Appendix B: List of Applications

Here is presented a list of commercial applications. The first table expresses some of the applications used to automatically manage/optimize a financial portfolio classified according to a specific set of parameters. The second table illustrates several trading applications based on technical analysis.

Table B.1 List of several portfolio management/optimization applications

Application	Version	Date	Technology	Financial data import?	Performance analysis?	Constraints	Multi-period?
PortOpt	1.0	2005	Java	No	No	Floor Ceiling Cardinality	No
Smartfolio www.smartfolio.com/	3.0.63	2008	MS Excel	Yes	Yes	Floor Ceiling Margin Market capitalization Cardinality	Yes
MvoPlus http://effisols.com/mvoplus/	1.6	2007	Unknown	No	No	Floor Ceiling Cardinality	Yes
OptimaTrader http://optimaltrader.net	3.2.6	2008	Unknown	Yes	Yes	Floor Ceiling; cardinality	Yes

Table B.2 List of several trading applications

Application	Heuristics to detect stock price patterns	Creation of new indicators and strategies?	Strategy backtesting?	Display support and resistance lines automatically?	Default trading orders	Display trend lines automatically?	Manually graphical objects creation?
BestTradingPro http://bancobest.iitech.dk	–	No	No	No	Stop Market	No	Yes, such as lines, symbols or alerts
MetaStock http://equis.com/	–	Yes, using their own programming language	Yes	No	Market	No	Yes, such as lines, symbols or alerts
ProRealTime http://prorealtime.com/	–	Yes, using their own programming language	Yes	Yes	Market	Yes	Yes, such as lines, symbols or alerts
TradeStation http://tradestation.com/	–	Yes, using their own programming language	Yes	No	Market	No	Yes, such as lines, symbols or alerts
OptimalTrader http://optimaltrader.net	ANN	No	Yes	No	Market	No	No

Appendix C: Table Parameters Description

Here, an explanation is presented about each of the parameters used to classify Tables 2.5, B.1 and B.2.

Table C.1 Parameters used to classify strategies on Table 2.5

Parameter	Description
Date	Publication date
Metaheuristic	The heuristic method used to solve the computational problem
Additional features	Additional features used to improve the heuristic performance
Constraints	The realistic constraints added to the considered model of the problem
Portfolio analysis	The methodology used to pick assets from the market
Multi-objective	If the goal of the used methodology is to simultaneously improve two conflicting objectives, for instance, minimizing risk and maximizing return
Evaluation function	What is the function used to classify a solution instance during the execution of the considered algorithm
Data used	The financial data used to train the metaheuristic or to build an optimal portfolio
Training data period	The considered period used to train the algorithm
Test period	The period where the portfolio was evaluated

Table C.2 Parameters used to classify portfolio management applications on Table B.1

Parameter	Description
Date	Publication date of the considered version
Technology	The technology used to execute the application
Financial data import	If it is possible to automatically import financial data from financial websites such as YahooFinance or GoogleFinance
Constraints	The realistic constraints that can be added to the model of the problem
Multi-period	If it is possible to rebalance the portfolio, i.e., if when optimizing a portfolio over a specific data, the previous held portfolio is taken into account in order to minimize costs

Table C.3 Parameters used to classify trading applications on Table B.2

Parameter	Description
Heuristics to detect stock price patterns	Which methods are used besides technical indicators to produce selling or buying indications
Creations of new indicators and strategies	If the application allows the possibility to the user to create his own strategies and indicators
Strategy backtesting	The possibility of testing a trading strategy on prior time periods
Display support and resistance lines automatically	If the application draws the respective lines automatically
Default trading orders	Which trading orders can be given to the application
Manually graphical objects creation	If the application allows the possibility of inserting graphical objects over the stock charts