Intelligent Financial Portfolio Composition based on Evolutionary Computation Strategies

António Gorgulho
Instituto de Telecomunicações
Torre Norte, AV. Rovisco Pais, 1
1049-001 Lisboa, Portugal
(351) – 218418093
[antoniogorgulho@lx.it.pt]

Rui Neves
Instituto Superior Técnico

Nuno Horta

ABSTRACT
The management of financial portfolios or funds constitutes a widely known problematic 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 which 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 presented paper 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 preliminary results are promising since the developed approach beats the remaining procedures during the crash.

Categories and Subject Descriptors

General Terms
Algorithms, Economics.

Keywords

1. INTRODUCTION
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 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 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 forms 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, 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, 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 [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 seventy 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.

The presented paper provides the description of a portfolio management system based on the former processes. The report is structured as following:

- Section 2 addresses the theory behind the developed work, namely the concepts of financial portfolio, portfolio management, and technical analysis. Also, in this section, it is given a brief overview about different methodologies which can be used to address the portfolio problematic.
- Section 3 illustrates the system’s architecture.
- Section 4 proposes the validation procedure used to evaluate the developed strategy.
- Section 5 summarizes the provided document and supplies the respective conclusion.

2. RELATED WORK
To get a better understanding about the underlined problem, some of the fundamental concepts related to the financial portfolio theory are explained during the following sections.

2.1 Financial Portfolio
A financial portfolio [5] consists in 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 [5], 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 for better returns.

2.2 Portfolio Management
As it was already mentioned, the goal of this work is concentrated on the automatic management of a portfolio. So, it is important to understand that we can apply two forms of management [5]:
- 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 [2]. 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.

In the case of the system here described, the intent was to adopt an active management approach by using technical analysis indicators.

2.3 Technical 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 so we can select potential profitable securities. Usually, investment analysts perform a fundamental or a technical analysis of the market. In this work, we exclusively employ the Technical Analysis [6] methodology. 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 we only need to study those factors in order to forecast market behaviour.

The applied strategies based 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 analyzed 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, we can start 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. Observe the smoothness on the SMA line, which allow us to easily perceive the market movements, in contrast with the zigzags performed by the stock prices.

![Figure 1: SMA application.](image)

Regarding an indicator such as the former one, we can formulate a strategy for defining buying and selling signals:
- 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 we can define different rules which allow us to score the distinct stocks within the market and subsequently pick the best securities according to the indicators employed.

2.4 Existing Solutions
In respect to the solutions already developed to address this problem, mostly of them focus on a passive management approach by using the Mean-Variance model [7] proposed by Harry Markowitz. 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. Based on this assumption, Markowitz provided a mathematical model which can be easily solved by metaheuristics such as Simulated Annealing (SA), Tabu Search (TS) or Genetic Algorithms (GA).

Generally, solutions [8], [9], [10], based on this model, focus their goal on optimizing a single-objective; the risk inherent to the portfolio, in order to determine the optimal portfolio composition and the weights assigned to each of the chosen stocks. Besides this single-objective formulation, other approaches [11], [12] try
to optimize simultaneously two conflicting objectives, the global risk and the expected returns of the securities within the portfolio.

Besides the referred works which mainly generate a diversified portfolio maintaining it for a specific set of time, Aranha and Hitoshi [13], [14], [15] provided a very interesting active management approach, by coupling the Markowitz’s model with a modelling cost mechanism, responsible for rebalancing the portfolio through time while, at the same time, minimizing the transaction costs. In their works, a completely different portfolio representation is used, based on a tree structure, which allowed them to obtain very interesting results.

Although Markowitz’s model is widely used to design the portfolio optimization problem, other models can also be considered. For instance, Black and Litterman [16] 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.

In addition to this passive approach which only tries to maintain a well diversified portfolio, recurring to the Markowitz’s model for picking the assets from the market and assigning the respective weight within the portfolio. We can also adopt an active strategy which tries to find under or overvalued stocks in order to achieve a significant profit with price’s rise or fall. For instance, Wagman [17] provided a simple framework based on Genetic Programming (GP) which tries to find an optimal portfolio with recurrence to a simple technical analysis indicator, the Moving Average (MA). The provided solution starts by generating a set of random portfolios (population), and the GP algorithm tries to converge in an optimal portfolio by using an evaluation function which considers the weight of each asset within the portfolio and the respective degree of satisfaction against the MA indicator, using different period parameters.

Another solution, based on the same kind of analysis, was provided by Yan, Clack et al. [18], [19]. Their solution is based on a genetic programming approach which tries to find an optimal model to classify the stocks within the market. The top stocks adopt long positions, the bottom ones, short positions. This approach is very interesting since it is capable to get a very realistic experience on financial portfolio management, besides being very robust.

Their model is based on the employment of a Fundamental Analysis approach which consists on studying the underlying forces of the economy to forecast the market development.

The following table summarizes some of the existing solutions to approach the portfolio problematic, specified according to several parameters.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Date</th>
<th>Metaheuristic</th>
<th>Additional Features</th>
<th>Constraints</th>
<th>Portfolio Analysis</th>
<th>Portfolio Representation</th>
<th>Evaluation Function</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ceiling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cardinality</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Turnover</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Trading</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Decision Model</td>
<td></td>
<td>Decomposition</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Round-Lots</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[10]</td>
<td>2008</td>
<td>Particle Swarm</td>
<td>---</td>
<td>Floor</td>
<td>Markowitz</td>
<td>Hybrid Structure</td>
<td>Lambda Trade-Off Function</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Optimization</td>
<td></td>
<td>Ceiling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cardinality</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. SOLUTION’S ARCHITECTURE
The system’s architecture can be structured on a traditional layer architecture composed by three distinct layers:

![System overall architecture](image)

**Figure 2: System overall architecture.**

Each layer is associated with several modules, represented by the oval shapes. The presented modules correspond to distinct units of implementation with a specific functional responsibility within the system.

### 3.1 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 following:

- 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.
- Afterwards, the system applies a set of technical indicators in order to calculate the values given by those indicators on the available data prices.
- 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 set of data calculated in the previous step.
- In order to evaluate each individual, an Investment Simulator is necessary for rank each stock within the market and subsequently, picking the best stocks for defining a financial portfolio. Afterwards, the portfolio is updated and evaluated during the training period in order to classify the attractiveness of the current classifier equation in terms of its performance on the end of the considered time period.
- 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.
- 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 tries to illustrate the defined procedure.

![Data flow example](image)

**Figure 3: Data flow example.**

### 3.2 Data Layer
The Data Layer is responsible for managing financial data and providing the respective access. Its behaviour is decomposed on two distinct modules, the Financial Data Processing Module and the Technical Rules Module.

#### 3.2.1 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. The process of retrieving all the historical data was performed just once. Afterwards, it is only necessary to update the database with new available information.

In respect to the considered data, two major market indexes were used:

- The DJI, Dow Jones Industrial Average Index [22], which contains the stock prices of 30 of the largest held companies in the United States.
- The S&P500, Standard & Poors 500 [23], 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 [24]. The complete retrieving process can be described as following:

- Specify the desired index. Each index is identified with an unique keyword. For instance, the Dow Jones Industrial Average is tagged with the acronym DJI.
- 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, responsible for defining csv files with the desired financial data. Each record within these stored files has the following configuration:
3.2.2 Technical Rules Module

One of the major problems we face on portfolio management is the right choice of assets; when we pick a specific stock we don’t know if its price is going to rise or fall. However, we can use technical indicators to give us a future perspective on its behaviour in order to determine the best choice. So, in order to classify each asset within the market, we employ a set of rules based on technical indicators. As stated already under the previous section, 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 behaviour 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 we can define different rules, which allow us to score the distinct stocks within the market and subsequently pick the best securities according to the indicators employed.

There are several problems that can show up when we use 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 we 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 in several ways, as to determine the best time span; for instance, Fernández-Blanco et al. [25], [26] applied an EA to determine the best settings for the MACD and RSI indicators. However, in this work we apply the algorithm 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 us, we try 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 highly 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 build a basket of different types of technical indicators; momentum oscillators and trend following devices:

- A trend following indicator tries to identify trends in the market. A trend represents a consistent change in prices, the investors’ expectations.
- 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 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 to go short.
- **High Score**: Assigns 0.5 points, indicates a reasonable buy signal.
- **Very High Score**: Assigns 1.0 points, indicates a strong buy signal.

Taking into account all the historical data, for each period a specific score was assigned taking into account the following technical indicators and defined rules. For instance, for the Rate of Change (ROC) indicator, the following rules were established:

<table>
<thead>
<tr>
<th>Table 2: ROC rules.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null Score</td>
</tr>
<tr>
<td>Down Score</td>
</tr>
<tr>
<td>Up Score</td>
</tr>
<tr>
<td>Full Score</td>
</tr>
</tbody>
</table>

The previous table demonstrates the rules applied for the ROC indicator. Besides this indicator, we have employed; Moving Average Convergence Divergence (MACD), Exponential Moving Average (EMA), Relative Strength Index (RSI), Hull Moving Average (HMA), True Strength Index (TSI), On Balance Volume (OBV), and the Double Crossover procedure.
3.3 Business Logic Layer

This layer is accountable 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. The layer is structured on two distinct modules: Optimization Module, responsible for the optimization process, and the Investment Simulator Module, accountable for simulate the portfolio management during the training/validation period.

3.3.1 Optimization Module

Since a GA is composed by several components we will start to describe how each component of the algorithm was defined.

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 asset 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, the following table is presented.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Last Rule</th>
<th>Buy Limit</th>
<th>Short Limit</th>
<th>Close Buy</th>
<th>Close Short</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0, 1]</td>
<td>[0, 1]</td>
<td>[0, 1]</td>
<td>[-1, 0]</td>
<td>[-1, 1]</td>
<td>[-1, 1]</td>
</tr>
</tbody>
</table>

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

\[
\sum_{i=0}^{N} W_i \cdot \text{Score}(X, i) = (1)
\]

\[
0 \leq W_i \leq 1 = (2)
\]

\[
0 \leq \sum_{i=0}^{N} W_i \leq 1 = (3)
\]

Where:

- \( W_i \) is the weight/importance assigned to the technical rule \( i \).
- \( \text{Score}(X, i) \) corresponds to the score given by the technical rule \( i \) to stock \( X \).

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 so a specific position within the portfolio can be closed.

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 [27] 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.

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:

\[
\text{Mutations} = \text{Mut. Rate} \cdot \text{Cr. Size} \cdot (\text{Pop. Size} - 1) = (4)
\]

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 [27] technique. However, the convergence process was worse when compared with the standard mutation operator.

Crossover

Considering the crossover operator, different types of crossover operators were implemented, in particular, the Single Arithmetic Recombination [28], the Whole Arithmetic Recombination [28], and the One-Cut Point Crossover [28] 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.

Constraints

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 [29], discarding infeasible individuals within the population. Although it seems an extremely basic approach, this methodology has as major problem the fact of not exploring any information from the infeasible individuals, in order to guide the search more effectively. To surpass this complication, we have employed a simulated artificial immune system [30] 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 behaviour 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 [30].

**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.

### 3.3.2 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. 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.

### 3.4 Presentation Layer

The presentation layer is responsible for the application interface. Since a detail explanation of this layer is outside of the context of this paper, we briefly describe the necessary inputs, to be specified by the average user:

- **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.

### 4. SYSTEM VALIDATION

In this chapter it is described the validation approach for evaluating the defined system.

In order to validate the designed application during its development, it was necessary to employ a testing strategy designated as Backtesting [31]. This process consists on testing a specific strategy on a prior time period in order to determine its effectiveness. For instance, suppose we intend to evaluate our strategy in terms of its year performance. Instead of waiting one whole year to do it, we can extract the past data and subsequently evaluate the procedure on the considered periods. Applying the developed strategy to prior data can be substantially beneficial, in order to detect strategy flaws and improve its potential.

### 4.1 Strategies Employed

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

- **Buy and Hold**: According to some theories [2], already addressed on the first chapter, prices are independent to each other, meaning that we cannot use past data to forecast market development, so the best strategy we can employ is buy and hold on which we maintain a specific set of assets regardless of market fluctuations. The major problem on implementing this strategy is in which assets should we concentrate to form the initial portfolio? Normally, experienced investors perform a fundamental evaluation of several companies and then compose their portfolio. Since this kind of data was not considered by the application, the adopted buy and hold strategy picks the assets which presented best average returns during the last year.

- **Random**: The random strategy implemented has 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 over the last six months adopt long positions. The ones who show worst performance take short positions. The portfolio is then maintained over three months. After those three months, new positions are taken according to the former process.
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.

4.2 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 to 2009. For the designed case study, the following configuration was applied:

Table 4: Case study I - Configuration.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market</td>
<td>All stocks from DJI</td>
</tr>
<tr>
<td>Period</td>
<td>01/01/03 – 31/06/09</td>
</tr>
<tr>
<td>Budget</td>
<td>100 000 USD</td>
</tr>
<tr>
<td>Maximum Size Portfolio</td>
<td>10</td>
</tr>
<tr>
<td>Short Selling?</td>
<td>TRUE</td>
</tr>
<tr>
<td>Commissions</td>
<td>0.02 / Share Minimum Fee: 14.00 USD</td>
</tr>
<tr>
<td>Number of Executions</td>
<td>100</td>
</tr>
</tbody>
</table>

In respect to the evolutionary strategy, the following parameter’s specification was employed:

Table 5: Case study I - Evolutionary configuration.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>64</td>
</tr>
<tr>
<td>Mutation Rate (%)</td>
<td>10</td>
</tr>
<tr>
<td>Generations</td>
<td>350</td>
</tr>
<tr>
<td>Trunc. Threshold (%)</td>
<td>50</td>
</tr>
<tr>
<td>Sliding Window</td>
<td>6 Months/6 Months</td>
</tr>
</tbody>
</table>

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

4.2.1 Return on Investment (ROI)

The following graph exhibits the results obtained for the considered strategies within the years of 2003 to 2009. Each curve represents the return on investment achieved by the respective investment methodology.

4.2.2 Classification Parameters

The following table 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 performed to thoroughly evaluate each methodology. The results for those strategies correspond to the confidence interval achieved when using a confidence degree of 95%.

Table 6: Case Study I - Classification parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>B&amp;H</th>
<th>Random</th>
<th>GA</th>
<th>Best GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROI (%)</td>
<td>7.17</td>
<td>[ -21.97 , -14.77]</td>
<td>[16.68, 25.29]</td>
<td>62.95</td>
</tr>
<tr>
<td>Sharpe Ratio</td>
<td>0.08</td>
<td>-0.93</td>
<td>0.21</td>
<td>0.67</td>
</tr>
<tr>
<td>Sortino Ratio</td>
<td>0.07</td>
<td>-1.25</td>
<td>0.40</td>
<td>21.03</td>
</tr>
<tr>
<td>Positions</td>
<td>10</td>
<td>[1371, 1389]</td>
<td>[151, 159]</td>
<td>156</td>
</tr>
<tr>
<td>Profitable Positions (%)</td>
<td>60.00</td>
<td>[46.83, 47.55]</td>
<td>[80.24, 81.50]</td>
<td>88.46</td>
</tr>
<tr>
<td>Non Profitable Positions (%)</td>
<td>40.00</td>
<td>[52.45, 53.17]</td>
<td>[18.50, 19.76]</td>
<td>11.54</td>
</tr>
<tr>
<td>Avg. Profit Per Position (%)</td>
<td>7.18</td>
<td>[-0.16, -0.07]</td>
<td>[1.93, 2.53]</td>
<td>4.00</td>
</tr>
</tbody>
</table>
Table 6: Case study I - Classification parameters (cont.).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>B&amp;H</th>
<th>Random</th>
<th>GA</th>
<th>Best GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Profit (%)</td>
<td>75.28</td>
<td>[63.04, 78.21]</td>
<td>[104.69, 136.57]</td>
<td>59.66</td>
</tr>
<tr>
<td>Min. Profit (%)</td>
<td>-53.01</td>
<td>[-42.96, -39.03]</td>
<td>[-36.46, -34.94]</td>
<td>-30.28</td>
</tr>
</tbody>
</table>

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

Figure 6: Case study I - Year ROI.

4.3 Case Study II – SP between 2006 and 2009
The presented case study exhibits the results obtained when evaluating the developed solution during the years of 2006 to 2009. For the designed case study, the following configuration was applied:

Table 7: Case Study II Configuration.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market</td>
<td>150 stocks from S&amp;P500</td>
</tr>
<tr>
<td>Period</td>
<td>01/01/06 – 31/06/09</td>
</tr>
<tr>
<td>Budget</td>
<td>100 000 USD</td>
</tr>
<tr>
<td>Maximum Size Portfolio</td>
<td>10</td>
</tr>
<tr>
<td>Short Selling?</td>
<td>TRUE</td>
</tr>
<tr>
<td>Commissions</td>
<td>0.02 / Share; Minimum Fee: 14.00 USD</td>
</tr>
<tr>
<td>Number of Executions</td>
<td>100</td>
</tr>
</tbody>
</table>

In respect to the evolutionary strategy configuration, we’ve used the settings defined on Table 5.

4.3.1 Return on Investment (ROI)
The following graph exhibits the results obtained for the considered strategies within the years of 2006 to 2009. Each curve represents the return on investment achieved by the respective investment methodology.

Figure 7: Case study II – ROI Evolution.
To highlight the superiority of the evolutionary strategy, on the end of the testing period, the following histogram demonstrates the ROI obtained for the different 100 executions experimented per each investment methodology.

Figure 8: Case study II – ROI distribution.

4.3.2 Classification Parameters
The following table 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 performed to thoroughly evaluate each methodology. The results for those strategies correspond to the confidence interval achieved when using a confidence degree of 95%.

Table 8: Case study II – Classification parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>B&amp;H</th>
<th>Random</th>
<th>GA</th>
<th>Best GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROI (%)</td>
<td>-11.41</td>
<td>[-19.60, -11.32]</td>
<td>[0.21, 7.35]</td>
<td>30.84</td>
</tr>
<tr>
<td>Positions</td>
<td>10</td>
<td>[768, 780]</td>
<td>[106, 117]</td>
<td>101</td>
</tr>
<tr>
<td>Profitable Positions (%)</td>
<td>20.00</td>
<td>[46.50, 47.40]</td>
<td>[75.62, 77.60]</td>
<td>78.22</td>
</tr>
<tr>
<td>Non Profitable Positions (%)</td>
<td>80.00</td>
<td>[52.50, 53.50]</td>
<td>[22.40, 24.38]</td>
<td>21.78</td>
</tr>
<tr>
<td>Avg. Profit Per Position (%)</td>
<td>-11.41</td>
<td>[-0.18, 0.00]</td>
<td>[2.19, 3.29]</td>
<td>6.80</td>
</tr>
<tr>
<td>Max. Profit (%)</td>
<td>83.12</td>
<td>[60.14, 79.80]</td>
<td>[103.63, 140.74]</td>
<td>140.32</td>
</tr>
</tbody>
</table>
5. SUMMARY AND CONCLUSION
The presented report proposed an application capable of automatically manage a portfolio by using a GA conjugated with technical analysis rules. As observed under the previous section, the system has potential and much more can be done in order to achieve a higher level of profitability. Although, several management rules were defined to increase the system performance, EC was extremely important to provide us with a correct balance between several types of technical indicators in order to pick the most promising stocks for portfolio composition.

6. REFERENCES

Table 8: Case study II – Classification parameters (cont.).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>B&amp;H</th>
<th>Random</th>
<th>GA</th>
<th>Best GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. Profit</td>
<td>-81.38</td>
<td>[-47.33, 42.44]</td>
<td>[-35.98, -33.25]</td>
<td>-32.84</td>
</tr>
</tbody>
</table>