Comparative study of four optimization algorithms applied to stock price forecasting

Miguel Ângelo Rocha Magalhães
Instituto Superior Técnico, Lisboa, Portugal
ist179032miguelmgm@gmail.com

Abstract—The proposed work presents an innovative investment strategy, capable of making a profit on a regular basis with low associated risk. The investment strategy is based on a forecast for the rise or fall of a company’s share price, calculated using various technical indicators. The intelligent trading system developed was optimized by 4 algorithms separately, in order to compare the performance of the algorithms: Cuckoo Search (CS), Genetic Algorithm (GA), Gray Wolf Optimization (GWO) and Moth Flame Optimization (MFO). To ensure that the results obtained are valid, tests were carried out for 104 companies from various areas, present in the NASDAQ 100 index (NDX). The algorithm that obtained the best overall performance of the tests performed was GA, although the other algorithms had very similar results. Even so, the MFO algorithm managed to accumulate an average of 60.3% profit over a 6-year period, reaching an average hit rate of 58.4% in the predictions made, being the algorithm that managed to accumulate more profit in a specific case study.

Key Words: Autonomous Trading System, Stock Market, Technical Analysis, Machine Learning, Artificial Intelligence (AI), Optimization Algorithms, Cuckoo Search (CS), Genetic Algorithm (GA), Gray Wolf Optimization (GWO), Moth Flame Optimization (MFO)

I. INTRODUCTION

The development of an autonomous system capable of predicting the exact time to enter and exit the stock market constitutes a more difficult challenge than it may seem, due to the fact that the stock market is essentially dynamic, non-linear, complicated to understand, non-parametric and chaotic in nature. In General there are three schools of thoughts regarding Stock Market prediction. The first school defends that no investor can achieve above average trading advantages based on historical and present information [10], contradicted by research [13] that provide appealing evidences to reject the random walk hypothesis and therefore researchers have been encouraged to suggest better models for market price prediction. The second school support the use of Fundamental analysis [11], although fundamental analysis is widely accepted, it is strongly dependent on the financial statements published by the companies. Since these reports are often only published every quarter of the year, an investment strategy driven by fundamental analysis can be slow to react to certain events that can impact the asset’s valuation. This limitation is overcome by the third school, which has a much lower reaction time, based on technical analysis [12]. Technical analysis as we know it today was first introduced by Charles Dow and the Dow Theory in the late 1800s. The basic idea of Dow Theory is that market price action reflects all available information, including fundamental factors. Later, in 1999 Murphy publish his book [14] covering the the latest developments in computer technology, technical tools, and indicators, in there are three premises on which the technical analysis approach is based: Market Action Discounts Everything, the statement “market action discounts everything” forms the basis of technical analysis. Many other principles follow from this idea. What it means is basically that anything that can possibly affect the market price (fundamentally, politically, psychologically and otherwise) is actually reflected in the market price. Prices Move in Trends, another premise that is crucial to technical analysis is that prices move in trends. That is, a price in motion is more likely to persist than to reverse. The entire trend-following approach is predicated on riding an existing trend until it shows signs of reversal. History Repeats Itself, the last assumption of technical analysis is that human nature does not change. Therefore, since market action is based on human psychology, history tends to repeat itself. By analyzing historical data, analysts use technical indicators to predict future price movements. The combination of an intelligent system with cognitive technologies, allows the autonomous analysis of large amounts of data, making it an effective way to overcome the challenges imposed by the stock market forecast. Currently, there are numerous approaches published in the available literature, which use Machine Learning combined with technical analysis to invest autonomously with some precision in the stock market.

There are several optimization algorithms that are commonly used to optimize autonomous trading systems in the available literature. In addition, in recent years, several studies have been published presenting promising optimization algorithms, which, as they are more recent and still little known, are not usually used for current optimization problems in the available literature. The main objective of this thesis is to develop an innovative autonomous trading system capable of generating returns on a regular basis, with the lowest possible risk associated. Comparing the performance of the system when being optimized by 4 different optimization algorithms, 3 of which are very recent in the available literature, Moth Flame Optimization (MFO)[1], Cuckoo Search (CS)[2] and Grey Wolf Optimization (GWO)[3]. Our motivation is to find out if these latest optimization algorithms are capable of surpassing the most reputable optimizer, the Genetic Algorithm (GA)[4]. To get an idea of the time scale, the GA algorithm was first introduced by John Holland, from the University of Michigan [4], in 1975, 33 years before any of the other 3 was published.
II. RELATED WORK

The work developed in this thesis was based, directly or indirectly, on these scientific works gathered. The literature gathered led to the development of the system, or specific mechanisms, used for the stock price forecasting. Starting by introducing the literature related with the existing methodologies to analyze and predict the stock market, specifically based on Technical analysis. Then, the papers used as inspiration for the development of the trading system and for the optimization algorithms used are described.

A. Stock Market Analysis

Gorgulho proposed a investment strategy based on a Genetic Algorithm merged with technical indicators by optimizing the weight of each indicator in the investment strategy for an efficient portfolio composition [5]. They utilize as Fitness Function the technical indicator Return on Investment (ROI) to evaluate each individual within the population. Their system was tested over the period of 2003 to 2009 against the Buy and Hold (B&H) strategy and a random strategy. The best GA achieved a ROI of 62.95% over this period, with 88.46% of profitable positions, proving the superiority of the GA system based in technical signals.

Ivo Pires in his thesis [9], presented an ensemble system that combines a Genetic Algorithm (GA) for dimensionality reduction and parameter optimisation, for the computation of the technical indicators, with a Random Forest (RAF) algorithm, with the goal of maximizing the returns and daily profits while minimizing the risk associated. The system uses daily prices and volume together with a set of technical indicators (such as Trend Following, Volume Indicators, Volatility Indicators) as input. Four different fitness functions were tested in order to perceive their impact on the prevision ability of the system to predict the future behaviour of the tested markets, yielding solutions with good performance. To test the performance of the implemented system, tests were carried out in five markets (the S&P500 index and four stocks from the index’s sectors). The results obtained by the system in the tests carried out, when investing in the S&P 500 index, showed that the only fitness function that allowed the system to surpass B&H strategy was the Risk Return Ratio (RRR) adequacy function.

José Matias Pinto in his thesis [8], used both Single-Objective and Multi-Objective Genetic Algorithms (GA) approaches were applied to perform automatic trading in stock markets. Several Technical Indicators (TI) were used in the trading system. Depending on the signals returned by each individual Technical Indicator, an overall decision is made, based on a weighted sum of all TIs. In addition, several alternatives to the classical fitness function Return On Investment were proposed and tested, both Single Objective and Multi Objective approaches, in order to optimize the core investment strategies. Fitness functions such as Annualized ROI, Risk Exposure, Maximum Draw Down (MDD), Variance of the results (Annually Adjusted Standard Deviation), Minimize the Variance of the Results versus Return (Min Var), Maximize the Number Periods with Positive Results. The tests performed in this work demonstrate the validity of Technical Analysis.

The good performance obtained in the multiple studies gathered proved that GA combined with technical indicators can be very useful in Financial Computing.

B. Investment Strategies Based

João Nadkarni presented in his thesis a model that combine the Principal Component Analysis (PCA) with the NeuroEvolution of Augmenting Topologies (NEAT) [6]. The trading system proposed uses several technical indicators as input, where the PCA method reduce the number of features used by the system by transforming the input data. The transformed low dimensional input was then fed to the NEAT algorithm that evolves an artificial neural network, that generate a trading signal. The proposed approach were tested with daily data from seven financial markets over the period of 27/03/2006 to 13/04/2017 using 80% of the data to train the system and 20% to test it. Three different fitness functions (Mean Daily Profit (MDP), Rate of Return (ROR and ROR/day), Required Rate of Return (RRR)) were considered in the NEAT algorithm to measures the profit obtained by the generated trading signal. The results achieved showed that this approach, using the Mean Daily Profit as fitness function, outperforms the B&H strategy in the majority of the markets tested, with the exception of the Apple stocks and Cotation Assistée en Continu (CAC) 40 index.

Antonio Silva proposed, in his work [7], a new approach to portfolio composition in the stock market. A Multi-Objective Evolutionary Algorithm with two objectives, the return and the risk, was used to optimize the model. First, the best stocks were chosen based on the fundamental indicators and then, second, the technical indicators indicate when to buy or sell. Real world constraints such as transaction costs were considered, investing in long-only positions. Uses two different investment models with real constraints, represented by two different chromosomes. The system was tested using the retrieved time series, with a window that slides one day at each increment. The training and real test simulations have been performed using 410 stocks of S&P 500 index, (considering as transactions costs 2% of the stock value), and using the index as a benchmark for comparing the results obtained in the training and the real tests. The population was trained during the time period from 2010 to 2012, making the real test during the period between 2012 and 2013. It can be seen in the results that the system was able to maintain better returns than the index for the same level of risk. When using the multi-objective algorithm, the best chromosome had a return of 36.4% while the index S&P500 had a return of 25.55% in the same period. It is difficult to get some quantitative numbers that could be used to classify the presented approaches, since the different trading strategies are applied to different market conditions and periods and using different metrics and techniques and schemes to evaluate their performance. Despite the difficulty present in the comparison between the presented solutions, it is possible to reach some conclusions from the analysis of the Trading Systems presented in the available literature. The available literature, based on active strategies of investment, shows that the most precise and with...
higher profitability approaches in stock market forecasting are based on technical analysis combined with Machine Learning techniques, normally using a Genetic algorithm as optimizer. However, new optimization techniques based on Swarm Intelligence have not always been taken into account in the available literature, or as many times as GA, due to the fact that they are very recent. In this thesis the optimization algorithms will be compared, based on the same performance metrics, for the same trading periods, for the same stocks.

III. IMPLEMENTATION

A. System Architecture

A Trading System was developed based on the active trading approach to invest in the stock market, combining technical analysis with an optimizing technique (algorithm). The trading system was optimized, separately, by 4 different optimization algorithms, making a comparison of the performance obtained by the 4 algorithms. Essentially, the trading system developed consists on a technical analysis applied to historical data of stock prices, to identify the right time to enter or exit the market, followed by the simulation of the investments made in the stock market based on the previous data. Calculating the Returns obtained by the trading system, according to the investments made. To maximize the performance of the trading system is applied an Optimization technique, combining the trading system with an Optimization Algorithm. The architecture of the model developed in this thesis is divided into five main modules, where each module has well defined responsibilities, as presented below:

The Data Module is responsible for obtaining data (historical stock price data) from local storage, organizing and pre-processing the collected data. Basically, the Data Module receives a set of data with the stock price history as input and outputs a dataset with the re-sampled pre-processed stock price history.

The Technical Analysis Module uses Technical indicators to interpret the stock market, inspired on Gorgulhos work[5]. In the technical analysis module, a set of multiple technical indicators is calculated. Then, a score (for each observation) will be extracted from each of the indicators, corresponding to how much the stock is under or overvalued, according to a set of predefined rules. This scores calculated will be used as features. Basically, the Technical Analysis Module receives as input a dataset with the stock price history, outputted from the Data Module, and outputs a set of scores (to each observation).

The Investment Simulator Module is responsible for the simulation of the trades made by the trading system developed, in the stock market, during a pre-defined time period. Executing the market entry and exit decisions, based on the predictions calculated through the weighted sum of the scores calculated in the Technical Analysis Module. Basically the Investment Simulator Module receives as input a solution (vector with weights correspondent to each feature) from the Optimization Module and a set of scores outputted from the Technical Analysis Module and outputs a predictive signal.

The Performance Evaluation Module is responsible for evaluating the performance of the Trading System. Calculating the fitness values, the accuracy and returns obtained by the trading system, according to the trades made in the Investment Simulator Module, during a specific trading time period. Basically the Performance Evaluation Module Receives as input the predictive signal vector outputted from the Investment Simulator Module, and the dataset with the stock price history outputted from the Data Module (or Optimization Module). Outputting a fitness value according to the performance obtained. The Optimization Module Basically the Optimization Module Receives as input the scores, outputted from the Technical Analysis Module. After simulating the system multiple times the system, using the Investment Simulator Module and the Performance Evaluation Module, outputs the best solution (set of parameters) that maximize the performance of the trading system. Defining a weight for each feature, equivalent to the responsibility it will have in the prediction of the stock price movement, during a specific trading trading period. The Trading System will be performed once for each of the 4 different optimization algorithms used in this work, so that the results are compared.

B. Overview of the Trading System

In order to facilitate the understanding of the general operation of the developed system, a flowchart of the trading system is presented in figure 1, illustrating the sequence of the main steps involved in the developed trading system, whose steps that make up the general flowchart of the system are described in more detail below.

(Step 1) Data Acquisition - The system starts by downloading the excel files with the datasets containing the financial data. Corresponding to the share price history, per minute, for 6 years, of the 104 companies that make up the SP100 index.

(Step 2) Pre-processing - After importing the datasets with the stock price history, due to the large size of the data set (almost 6 years of observations per minute), it was necessary to resample the data set, discarding some unnecessary observations. With this the dataset will keep the rows that represent the significant fluctuations in the stock price, discarding the observations that practically don’t show fluctuation in relation to the previous observation.

(Step 3) Technical Analysis - The Technical Analysis process begins with the application of a set of Technical Indicators to the financial data of a given stock, previously pre-processed, to then employ a set of rules for each of the calculated technical indicators. Where each set of rules applied to an indicator is responsible for analyzing each observation, in order to determine at every moment whether the share price will rise, fall or remain, assigning a score to each observation, for each indicator used.

(Step 4) Data Splitting - The developed system uses the sliding window approach, where each sliding window is composed of a dataset corresponding to the training phase and another to the test phase. Each training phase covers a period of approximately 8 months and the test phase covers the next 4 months. The dataset, containing total financial data and scores, was divided into multiple datasets.

(Step 5) Window Selection - The developed system use the sliding window approach, therefore the system have multiple
training phases always followed by the test phase. In this step, one of the sliding windows was selected. Moving on to the next sliding window, when all observations from the actual sliding window are used.

**Step 6: Calculation of the Predictive signals** - Using the data set corresponding to the training phase of the selected sliding window (in the previous step) and the population solutions, a value was calculated for each observation in the data set. Each value was calculated using a weighted sum of the score of each technical indicator, where each weight (normalized) is given by one of the values that makes up the solution. Creating a vector called predictive signal, with a dimension equal to the number of observations present in the data set, for each individual in the population. Each value saved in the vector predictive signal corresponds to the prediction made for the corresponding observation.

**Step 7: Performance Evaluation** - In this step, the fitness value of each individual in the population, using the dataset corresponding to the training phase of the selected sliding window and the predictive signal for the same period. The return of each prediction was recorded to calculate the fitness value (accuracy) obtained by each individual (solution) in the population.

**Step 8: Population Orientation** - The Optimization algorithm used maintains and successively improves a collection of potential solutions, being responsible for finding the ideal combination of values that make up the solution that maximize the performance metric (fitness value) in the training phase. Responsible for simulating the trading system several times during the chosen training period, saving the best solution when the stopping criterion is reached, to later be used in the Test phase. The solutions were started randomly in the search space, and were directed to the best areas through the interaction between solutions, according to some rules.

**Step 9: Calculation of the Predictive signal** - This step is very similar to step 6, but the predictions were calculated using only one solution, the individual with the best fitness (performance) value from the training phase, for the test phase that follows the training phase in question.

**Step 10: Merging Datasets** - Basically, in this step, the different datasets (composed by the financial data) and predictive signal vectors, from each of the test phases, were grouped into a single data set, to be subsequently calculated several performance metrics for the total test period.

**Step 11: Final Performance Evaluation** - Finally the performance of the trading system for the total test period was calculated, using the dataset and the predictive signal of all the grouped test phases. The Rate of Return (ROR) obtained for each investment was recorded in order to calculate multiple performance metrics obtained by the investment strategy developed, including metrics such as accuracy, accumulated rate of return, maximum drawdown and others. The results of the various performance metrics obtained by the trading system were recorded in an excel file.

C. Data Module

1) **Data Acquisition Sub-Module**: Several datasets (files) were imported by the Data Module, one for each company, to later be used by the trading system. The multiple datasets of raw data supplied correspond to the historical data of stock prices, by minute, from 104 companies that make up the NASDAQ 100 Index (NDX) covering the time period from 25 October 2012 until 23 November 2018. As imported, the raw financial data, is composed by the following attributes (columns):

- **Date** - The Date attribute gives the trading date and hour corresponding to each observation recorded in the dataset, used as index of the Data structure.
- **Open** - The stock opening price corresponds to the price at which the stock was first traded on the exchange, in a one minute trading period.
- **High** - This attribute represents the highest price the stock was traded, during a one minute trading period.
- **Low** - This attribute represents the lowest price the stock was traded, during a one minute trading period.
- **Close** - The stock’s closing price corresponds to the price at which the stock is last traded, in a one minute trading period.
- **Volume** - The volume attribute corresponds to the number of stocks that are exchanged during a one minute trading period.

2) **Data Pre-Processing Sub-Module**: After being imported, the data will be pre processed, discarding the observations (lines) of the dataset where any of the attributes are missing. The resampling process will discard all the rows that the Close Price do not change more than a certain (0.024%) percentage in relation to the previous row. With this the dataset will keep the rows that represent the significant fluctuations in the stock price, discarding the observations that practically don’t show fluctuation, in relation to the previous observation. Thus reducing the amount of redundant and noisy features, since very similar observations do not provide any additional information, given the large scale of the data set analyzed. So, basically, with the resampling process, the dataset will keep the observations practically per hour, more or less time, when the stock price is rapidly fluctuating. However, when the stock price is stagnant, only observations per week will be kept in the dataset.

D. Technical Analysis Module

The Technical Analysis starts with the application of a set of technical Indicators to the financial data of a specific stock, to subsequently employ a set of rules to each of the calculated technical indicators, with the aim of predicting whether the stock price will increase or decrease. The Technical Analysis procedure applied on this Module is inspired by Gorgulho approach.[5]

1) **Indicators Calculation Sub-Module**: The following technical indicators were used in this work: Bollinger Bands (BB) Indicator, Exponential Moving Average (EMA) Indicator, Hull Moving Average (HMA) Indicator, On Balance Volume (OBV) Indicator, Double Crossover (DB) Indicator, Rate of Change (ROC) Indicator, True Strength Index (TSI) Indicator, Relative Strength Index (RSI) Indicator, Moving Average Convergence Divergence (MACD) Indicator.

2) **Scores Calculation Sub-Module**: This second part of the Technical Analysis Module is responsible for the characterization of each of the technical indicators, used to determine at
every moment whether the share price will rise, fall or hold. Upon receiving the dataset, consisting of preprocessed financial data and technical indicators, a score is assigned to each indicator, for each observation, according to a set of predefined rules. The data set, including all the calculated scores, was saved in an Excel file for later use in the Investment Simulator Module. The classification given covers five possible grades:

- **Very Low Grade**: Assigns 1.0 points, indicates a strong sell/short signal. Announcing, with some certainty, that the stock price will fall.

- **Low Grade**: Assigns 0.5 points, indicates a potential(reasonable) sell/short signal. Announcing a potential fall in the stock price.

- **High Grade**: Assigns 0.5 points, indicates a reasonable(potential) buy/long signal. Announcing a potential rise in the stock price.

- **Very High Grade**: Assigns 1.0 points, indicates a strong buy/long signal. Announcing, with some certainty, that the stock price will rise.

- **Neutral Grade**: Assigns 0 points, indicates a neutral signal. Announcing, that the stock price will hold.

The rules developed to classify the share prices, based on the indicator EMA (12), are shown in Table I.

<table>
<thead>
<tr>
<th>EMA (12) Indicator</th>
<th>Scoring Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very Low Score</strong></td>
<td>If the Close price line crosses below the EMA line.</td>
</tr>
<tr>
<td><strong>Low Score</strong></td>
<td>While the EMA line is going down (negative slope).</td>
</tr>
<tr>
<td><strong>High Score</strong></td>
<td>While the EMA line is going up (positive slope).</td>
</tr>
<tr>
<td><strong>Very High Score</strong></td>
<td>If the Close price line crosses above the EMA line.</td>
</tr>
<tr>
<td><strong>Neutral Score</strong></td>
<td>Otherwise</td>
</tr>
</tbody>
</table>

**TABLE I**

**Scoring rules for the EMA (12) Indicator**

---

### E. Predictive signal Calculation

The vector called predictive signal has the dimension equal to the number of observations present in the dataset, where each position of the vector correspond to the prediction made for an observation.

The vector S represents a solution, with the same dimension as the search space defined as eighteen (18). This vector includes the weights assigned to each one of the sixteen (16) attributes, used to calculate the prediction signal. It also includes two extra positions, corresponding to two limit values.
that define the minimum and maximum amount of the sum, needed to obtain a Strong Rise and Strong Fall prediction respectively. In equation III-E, a representation of the solution is presented.

\[ S = (W_{feature1}, W_{feature2}, W_{feature3}, \ldots, W_{feature16}, Min_{sum}, Max_{sum}) \]  
(1)

The formula used to calculate the weighted sum vector is presented in equation 2, and the equation 2 presents the formula used to calculate the predictive signal.

\[ Sum_{vector}(n) = \sum_{i=1}^{16}(W_{feature_i} \times feature_i(n)) \]  
(2)

Predsignal(n) =
\[
\begin{cases} 
1, & \text{for } Sum_{vector}(n) > Min_{sum} \\
-1, & \text{for } Sum_{vector}(n) < Max_{sum} \\
0, & \text{c.c.} \end{cases}
\]  
(3)

The Sum vector and the Pred signal, called predictive signal, are initialized to zero. Where the feature correspond to an attribute of the data set, i defines the number of the feature considered and n correspond to the position in the vector correspondent. Therefore the value 1 means a Strong Rise (SR) prediction, the value -1 means a Strong Fall(SF) prediction and the value 0 means a neutral prediction.

F. Investment Strategy Sub-Module

Whenever a long position was opened, it was imposed that should remain open at least for the next 5 observations, even if the forecast for those positions was a price drop (Strong Fall prediction). The investment strategy Sub-Module receives the vector predictive signal and it’s responsible for deciding when to open and close the long and short positions, based on the predictions saved in vector received. To execute these decisions, start by creating 2 vectors with the same size as the vector predictive signal received. Called x-long and x-short Both initialized to zero. Using one of the vectors (x-short) to indicate when the trading system decided to open or close a short position and the second vector (x-long) has the same application but to point out long positions. Traversing the received vector predictive signal, whenever a position is found where a Strong Fall (value = -1 in the predictive signal) is predicted and in the previous position the same does not occur (value != -1 in the predictive signal), the value 1 is stored in the position of the x-short vector corresponding to the observation under analysis. And whenever a position is found where a Strong Fall is not predicted (value != -1 in the predictive signal) but is predicted in the previous position (value = -1 in the predictive signal) the value saved in the X-short vector is -1, in the position of the x-short vector corresponding to the observation under analysis. The X-long vector is filled in almost the same way only for Strong Rise prediction, but the value -1 is saved in the X-long vector only if the current position is at least five positions after the position where the value 1 was saved. Thus, each vector created (X-long and X-short) indicates when the trading system decided to open (positions in the vector with a value of 1) and close (positions in the vector with a value of -1) a position in the Stock Market, where the X-short vector refers to the short positions and the X-long vector refers to the long positions. The figure 2 illustrates an example of how the values are stored in the three vectors, Predictive Signal, X-long, X-short.

G. Fitness function

For the calculation of fitness value, a score is associated with each prediction according to the return obtained by that forecast, assigning a score of 1 for predictions with a positive return and a score of -1 for predictions with a negative return. The total accuracy of each solution is calculated by adding up all the scores obtained during the training period. This metric was the fitness function of the system used in the Training phase. For each trade carried out, a transaction cost of 0.1% of the value of the transaction was considered.

IV. RESULTS

A. Financial Data Division

In the case studies under analysis, a rolling window mechanism will be used, which means that the financial data from 25 October 2012 until 23 November 2018, will be divided into eighteen different datasets to be used in the training phase and another eighteen datasets to be used in the test phase. Where each dataset used in the train phase contain 10% of the total financial data and each dataset used in the test phase contains 5% of the total financial data. Each rolling window consists of a dataset corresponding to the training phase followed by the dataset corresponding to the test phase. The Figure 3 illustrates how the total dataset was divided into several sliding windows, where each sliding window consists of a training and test period. After the rolling window passes through all the financial data, the results are grouped together in a single dataset to calculate the accumulated Return on Investment (ROI) over the entire test phase. The Trading system will be tested for 90 percent of the total financial data from 2014-04-08 14:43:00 to 2018-12-11 20:34:00, for the example of the Intel shares.

B. Case Studies

Three case studies were developed to validate the proposed trading system and the optimization algorithms used. The first case study focuses on analyzing the search capacity and
time complexity of the four algorithms, using three known optimization problems in the area of computer science. The following two case studies used the trading system optimized by each of the four algorithms separately. The difference is that in case 2, two different investment strategies were used, and in case 3, only one was used. The only difference in the strategies used is the Stop Loss and Take profit component. For the second case study, one of the strategies does not use Stop Loss or Take Profit and the second strategy uses fixed Stop Loss and Take Profit, setting the take profit to 20% and the stop loss to -10%. To finish the 3 case study, use a strategy with variable Stop Loss and Take Profit, the Stop Loss varies between -5% to -15% and the Take Profit varies between 20% to 25%, depending on the standard deviation of the share price at the moment. The parameters used by the implemented system are described in Tables II and III.

Since the optimization algorithms are stochastic, in order to guarantee robustness and avoid discrepancies in the results, all simulations were performed 10 times for each algorithm, presenting only the average of the 10 results.

### Table II
**Definition of the parameters used by the 4 algorithms**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Generations</td>
<td>100</td>
</tr>
<tr>
<td>Population size</td>
<td>50</td>
</tr>
<tr>
<td>Dimension of Search Space</td>
<td>18</td>
</tr>
<tr>
<td>Lower Bound</td>
<td>1</td>
</tr>
<tr>
<td>Upper Bound</td>
<td>1</td>
</tr>
<tr>
<td>Transaction Costs</td>
<td>0.1%</td>
</tr>
</tbody>
</table>

### Table III
**Definition of the parameters used by the Genetic Algorithm**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value / Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of parents mating</td>
<td>25</td>
</tr>
<tr>
<td>Offspring size</td>
<td>45</td>
</tr>
<tr>
<td>Parent selection type</td>
<td>Steady State Selection</td>
</tr>
<tr>
<td>Keep parents</td>
<td>5</td>
</tr>
<tr>
<td>Crossover type</td>
<td>Uniform</td>
</tr>
<tr>
<td>Mutation type</td>
<td>Random</td>
</tr>
<tr>
<td>Mutation percent genes</td>
<td>40%</td>
</tr>
</tbody>
</table>

### Table IV
**Table with the performance metrics, obtained by the trading system, for each algorithm, using the first investment strategy**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Case 2 First experiment Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accumulated total ROI (in %)</td>
<td>GWO</td>
</tr>
<tr>
<td>N° of Investments</td>
<td>35</td>
</tr>
<tr>
<td>Nº of Investments Long</td>
<td>25</td>
</tr>
<tr>
<td>Nº of Investments Short</td>
<td>8</td>
</tr>
<tr>
<td>Nº of investments with profit</td>
<td>25</td>
</tr>
<tr>
<td>Total Accuracy (in %)</td>
<td>58.39</td>
</tr>
<tr>
<td>Standard Deviation of the Returns</td>
<td>7.036</td>
</tr>
</tbody>
</table>

1) **First Investment Strategy**: The average results obtained by the Trading System, using the first investment strategy, for each algorithm, are presented in the table IV.

The Fig 4 illustrates an example of the Return on Investment (ROI) accumulated in the Trading System simulations for the four algorithms, without using stop loss or take profit, according to the investments made during the simulation of one of the companies that make up the Nasdaq 100 stock market index (NDX).

2) **Third Investment Strategy**: The average results obtained by the Trading System, for each algorithm, using the third investment strategy, are presented in table V.

The Fig 5 illustrates the average Return on Investment (ROI) accumulated in the final simulations of the complete Trading System for the four algorithms, with the stop loss and take profit assuming variable values, according to the investments made during the simulation of one of the companies that make up the Nasdaq 100 stock market index (NDX).

3) **Conclusions**: Although none of the 4 algorithms was absolute superior in all experiments, in general the algorithm that obtained the best performance was the Genetic Algorithm optimizer. Emphasizing that the GWO algorithm presented the best performance in the first case study, referring to the exploration and exploitation capacity of the algorithm, although it is also the algorithm that needs more computational resources. However, in the following two case studies, the GWO algorithm demonstrated that was too overfitted to the training phase, as it was the one that got the worst performance in general when used as an optimizer for the developed Trading system. In contrast, the CS algorithm presents the worst performance of the four algorithms in the first case study, but in general it was the algorithm with the second best performance when used as an optimizer for the developed trading system, obtained very similar results to the GA algorithm. The
MFO algorithm is the algorithm that requires less computational resources, obtaining a very acceptable performance in case studies two and three, in fact it was the algorithm that accumulated the highest average ROI in a period of 6 years, when used as an optimizer of the trading system without stop loss or take profit, accumulating an average ROR of 60.3% over a six year period, with an accuracy of 58.4%. The results obtained show that the trading system developed presented a good performance, for the four algorithms used, obtaining values of accumulated ROI, accuracy and maximum drawdown.
TABLE V
TABLE WITH THE PERFORMANCE METRICS, OBTAINED BY THE TRADING SYSTEM, FOR EACH ALGORITHM, USING THE THIRD INVESTMENT STRATEGY

<table>
<thead>
<tr>
<th>Metric</th>
<th>GWO</th>
<th>GA</th>
<th>MFO</th>
<th>CS</th>
<th>B&amp;H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accumulated total ROI (in %)</td>
<td>57.2742</td>
<td>39.7916</td>
<td>39.0973</td>
<td>39.3864</td>
<td>108.7305</td>
</tr>
<tr>
<td>Nº of Investments</td>
<td>42</td>
<td>5</td>
<td>41</td>
<td>42</td>
<td>1</td>
</tr>
<tr>
<td>Nº of Investments Long</td>
<td>35</td>
<td>34</td>
<td>36</td>
<td>37</td>
<td>1</td>
</tr>
<tr>
<td>Nº of Investments Short</td>
<td>7</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Nº of investments with profit</td>
<td>24</td>
<td>20</td>
<td>23</td>
<td>24</td>
<td>1</td>
</tr>
<tr>
<td>Total Accuracy (in %)</td>
<td>56.59</td>
<td>58.68</td>
<td>57.30</td>
<td>57.08</td>
<td>-</td>
</tr>
<tr>
<td>Maximum Drawdown (in %)</td>
<td>-32.52</td>
<td>-30.17</td>
<td>-31.64</td>
<td>-32.16</td>
<td>-39.12</td>
</tr>
<tr>
<td>Standard Deviation of the Returns</td>
<td>7.050</td>
<td>7.270</td>
<td>7.047</td>
<td>7.056</td>
<td>2.90111</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

We conclude that the trading system presented in this thesis is capable of investing autonomously in the stock market, with profits on a regular basis, minimizing the associated risks. Demonstrating that there are recent optimization algorithms that are still little used in solving optimization problems in the recent literature, but that they are capable of obtaining quite good results when compared to the most recognized algorithms. One of the great innovations of this thesis is the implementation of a trading system optimized by four algorithms separately, three of which were rarely (or never) used to optimize an intelligent trading system in the available literature, and never together. This document allows an easy analysis and comparison of the performance of 4 optimizing algorithms, since they were simulated using the same parameters, using the same training and test date, being evaluated by the same performance metrics (financial and computational). Showing that the intelligent trading system developed despite not generating as much return on average, has a lower associated risk when compared to leaving the money invested in the stocks that make up the NASDAQ 100 index for 6 years. To finish the analysis of the algorithms compared in this thesis, we conclude that the four algorithms showed a high capacity to solve complex real world optimization problems, as in the forecast of stock price fluctuations. More specifically, the three recent optimization algorithms used showed that they are at a level similar, or even higher in some cases, to that of the GA algorithm, which is already more recognized and widely referenced in the available literature.

VII. ACKNOWLEDGMENT

I would like to thank Professor Rui Neves for all the guidance, help and constructive criticism, of fundamental importance, enabling the publication of this thesis, reaching the intended objectives. I would also like to thank my family and friends for all the motivation and support given, very important to finish this work.

REFERENCES

Fig. 6. Example of ROI obtained for 4 investment strategies