Developing Multi-Time Frame Trading Rules with a Trend Following Strategy, using GP

Jaime Machado

Abstract—This work presents a methodology to develop trading rules with a trend-following approach combining two time frames. Rules are based on technical indicators like the RVI and MACD and are individually optimized for 19 stock indices and 11 commodities in the period from 2006 to 2014. Tree structures are used to represent the trading rules which are gradually evolved through evolutionary algorithms. The best solutions to trade these assets are then simulated individually and in portfolios. The use of two time frames allows a reduction in risk since two different profiles of trades are combined in what can be described as trading system diversification. An innovative algorithm to delete similar rules is also presented, based on the Mean Squared Error of the generated trading signals. The trading rules obtained by this method are able to profit from upward and downward trends and react fast to sharp falls. In the bear market of 2008 the optimized rules not only prevented sharp drops in capital but managed to profit from the declining prices.


I. INTRODUCTION

MARKETS have an important role in our societies and they have been studied by academics for many decades, giving rise to different theories. Some argue markets are efficient [1] meaning that all relevant information is already incorporated in the prices and it’s impossible to predict its future movements. This supposedly leads to a market behavior known as Random Walk [2]. Others refute this view and defend markets are not efficient and it’s possible to predict their future behavior to some extent [3]. Increasing evidence point to markets not being efficient which means methods can be developed to predict market behavior increasing profits and/or reduce risks [3], [4].

When market participants try to predict the future behavior of a market or asset, two very different approaches have been used. One, called Fundamental Analysis [5] looks at the quality of the asset or its intrinsic value and determines if it’s undervalued or overvalued. The other is called Technical Analysis [5], [6], and studies past prices on the premise that all relevant information is reflected there and it’s possible to predict it’s future behavior by analyzing the formed patterns.

In the field of technical analysis, several techniques were developed in order to extract the useful information contained in past prices. Some try to identify graphical formations that correspond to a higher probability of a particular outcome and are called Chartists. Others use technical indicators calculated with past prices which filter relevant information that can then be used to decide what asset to trade and when.

Each market has its own characteristics and particular trading techniques used in one market may not work well in other markets. Also, each market may exhibit different behaviors in different phases, consequence of underlying factors, trader psychology of simply as a result of system dynamics. This leads to the necessity of adapting the trading approach or the trading rules as the market changes, to maximize profits and minimize risk. In the past decades much attention has been devoted by academics to the problem of optimizing trading rules and several optimization algorithms have been used, like Neural Networks, Fuzzy Logic, Monte Carlo, Particle Swarm, Hidden Markov Models, Support Vector Machine and Evolutionary Algorithms.

Evolutionary Algorithms are particularly suited to optimize trading rules because of their flexibility and robustness and is the optimization method used in this work. Trading rules exclusively based on technical indicators will be optimized with Genetic Programing, a form of Evolutionary Algorithm. Several chosen assets (stock indices and commodities) will be independently optimized and portfolios will be simulated using those optimized trading rules. In order to increase the robustness of the solutions, the optimization will be restricted in order to produce solutions which follow market trends, since that is one of the most structural and frequent market patterns [6]. Since markets can exhibit different behaviors in different time-scales, rules tuned to two different time frames will be combined in order to increase the robustness of the solutions.

II. RELATED WORK

Creating trading rules optimized to trade a particular asset in specific market conditions has been an active topic studied by academics. Fundamental Analysis (FA) [5] and Technical Analysis (TA) [5] have been the market analysis techniques used as the base for the creation of the trading rules. In most works only one is used but some authors combine rules based on these two techniques [7]. Several different optimization algorithms can be used to optimize the rules but Evolutionary Algorithms (EA) are particularly suited because of their

Jaime Machado is with the Instituto de Telecomunicações, Instituto Superior Técnico (email: jaime.machado@ist.utl.pt).
flexibility and robustness, in particular Genetic Algorithms (GA) [8] and Genetic Programming (GP) [9], [10]. In these algorithms solutions are represented by individuals that belong to a diversified population. The population is iteratively evolved towards better solutions by ranking the individuals according to a fitness function and reproducing the best.

The academic research experimented on different aspects of optimizing trading rules and the most important are:

- Testing specific TA or FA rules to generate the trading signals, with different configurations and parameters;
- The possible advantages of different representations of the solutions (ex: tree or array);
- Testing different fitness functions, simple or complex, with several components (including for example: return, risk, complexity of solutions). Multi-objective optimization are also studied;
- Comparing the results of optimizations with different time frames (Ex: daily, weekly and monthly) and combining them as well;
- Testing different global methodologies for the process of obtaining the solutions.

Korczak and Roger used GA to develop trading solutions for 24 French stocks using more than 200 fixed rules based on TA [11]. The optimizer selects the combination of rules that produce the best solution and the final trading decision is given by averaging the outputs of all the selected rules. The obtained results beat the Buy-and-Hold (B&H).

Contreras et al. combined fixed rules based on TA and FA using GA on 100 stocks from S&P500 [7]. Each element in the population contains the parameters of 8 trading rules, 4 based on TA and 4 based on FA. Each rule contributes with a certain weight to the final trading decision and these weights are also optimizable. The obtained results beat the B&H.

Allen and Karjalainen used GP with tree representation to generate trading rules for American Stock Index S&P500, tested from 1936 to 1995 [12]. The operators used were arithmetic, logical, Simple Moving Average, maximum price in an interval, minimum price in an interval, >, <, If-Then-Else and Lag (offsets a vector received from a child node). The authors tried to limit curve fitting by training the population in a 2 year window and then validating the result in another 2 year window. When the best element has a better fitness score in the validation window, it is saved, and otherwise it’s discarded. Optimization stops after a predefined number of generations with no improvement in the validation window. S&P500 historical prices are normalized, dividing them by the 250 day MA. The results of this configuration did not beat the buy-and-hold strategy.

Becker and Seshadri used GP to generate trading rules for S&P500 [13]. They used two chromosomes, one to generate buy signals and another for sell signals. They opted to reproduce the two chromosomes independently, meaning that a buy and a sell chromosome will never exchange genes and evolve separately. But since a trading solution must have synchronized buy and sell signals to be coherent, the evolution of the two chromosomes is linked and interdependent - the authors named this process as cooperative coevolution. There are several options to manage the two types of chromosomes and some were tested in this work: 1) each buy chromosome is paired with a sell chromosome in a solution and they evolve together; 2) each chromosome of one type is paired with the best 5 chromosomes of the other type, from the previous generation; 3) each chromosome of one type is paired with 5 randomly chosen chromosomes of the other type. The results showed option 1) with paired chromosomes performed best and the authors argue it may mean that compatibility between the chromosomes is more important than diversity. Tests were performed in S&P500 from 1954 to 2002.

Lohpetch and Corne experimented with multi-objective GP with chromosome tree representation in order to trade S&P500 stocks [14]. Results showed that multi-objective strategies outperform the single-objective strategies. Simulations with monthly data produced the best results outperforming the B&H, weekly data gave worse results but still outperforms B&H. Using daily data the results are equivalent to B&H.

A summary of relevant works can be consulted in Table 1.

III. PROPOSED SOLUTION

This work proposes a method to develop optimized trading rules for individual assets, using GP to evolve solutions with a trend-following approach and combining two time-frames.

A. Genetic Programming

The optimization algorithm chosen was genetic programming since it combines flexibility and robustness. [9], [10]. A population of diversified solutions exist and each element/solution is designated as a genome. Each genome includes all the necessary information to generate the trading signals used to trade an asset. The internal structure of a genome includes several components which are designated as subsystems. A subsystem is an independent set or trading rules with generates its own trading signals and which contributes with a certain weight to the final trading signals. This organizations allow genomes to have several independent trading rules working in parallel, with different characteristics, see Figure 1. A subsystem can have two chromosomes, one dedicated to generating buy signals and another dedicated to generating short signals. But it also possible for a subsystem to have only one chromosome which generates both buy and short signals.
<table>
<thead>
<tr>
<th>Ref</th>
<th>Paper date</th>
<th>Optim. Method</th>
<th>Operators/Genes</th>
<th>Representation</th>
<th>Fitness function</th>
<th>Time Frame</th>
<th>Assets used</th>
<th>Test Period</th>
<th>Results</th>
</tr>
</thead>
</table>

Table 1 – Summary of related works

A chromosome has a tree structure composed by nodes and each node is defined as a gene, see Figure 2.

Genes can be operators, market data or constant values. Some genes have parameters which can be optimized and other don’t have parameters. Genes can have inputs if they are operators or have no inputs if they are terminals. Both inputs and outputs can be of the type Double or Boolean and the chromosome tree must be organized in such a way that parent-child genes have compatible input-output data types. The outputs of a chromosome are the output of the root gene and must always be of the type Boolean.

The Boolean outputs of a chromosome can have different meanings depending if the chromosome is of type long, short or mixed, see Table 2.

The optimization uses a sliding window procedure which alternates an in-sample window for training/optimizing and a contiguous out-of-sample window for testing. These two windows advance in time until there is no more data, see Figure 3. A size of 3 years was chosen for the in-sample window and a size of 3 months was chosen for the out-of-sample window.

Table 2 – Meaning of chromosome outputs

A particular window is optimized in successive generations were the population evolves towards better solutions according to a fitness function. In each generation genomes produce vectors with trading signals which are used to simulate trading on an asset and obtain performance metrics used in the fitness function.

There are 5 different reproduction operators. Elitism and “Best Chromosome Merge” are applied first and the remaining genomes are reproduced with the other 3 operators. The reproduction operators are:
• **Elitism** – In each generation the genomes representing the 5% most scored are copied unchanged to the next generation.

• **Crossover** – A child chromosome is created from the chromosome trees of two randomly selected parents [9].

• **Hard Mutation** – A child chromosome is created from a randomly chosen parent by substituting a fraction of its chromosome tree by a randomly generated sub-tree [9].

• **Soft Mutation** – The parameters of chromosome genes are randomly mutated.

• **Best Chromosome Merge** – In each generation, a special genome is created by merging the best chromosomes of each position in the genome. Chromosomes are simulated individually and a ranked list created for each chromosome position. The best chromosome of each list is then picked and a new genome created.

Crossover, Hard Mutation and Soft Mutation are mutually exclusive and only one is chosen with probabilities 0.5, 0.3 and 0.2 respectively. In these 3 operators Genomes are reproduced chromosome by chromosome and a chromosome from a particular position in the genome never exchanges genes with a chromosome from a different position.

**B. Trend-Following solutions**

Markets tend to form repeating patterns due to trader behavior/psychology, underlying factors in the economy and system dynamics. The most frequent patterns are:

1. **Trends**: Which can be defined as a continued bias of the marked in a particular direction. In mathematical terms it corresponds to a high autocorrelation of price. Trends can form in different time frames and can last up to several years. A trend can be upwards, downwards and usually has successive swings which form a zigzag-like pattern.

2. **Mean-Reversal**: Which happens when prices overshoot a theoretical mean line representing the consensus of a fair price and then reverses back to this line. The overshooting can continue in an oscillatory behavior.

Trends exist since markets were created and is a pattern which can consistently be exploited for profits. Trend-following is a trading strategy that detects a trend, opens positions to profit from that trend and maintains the position open until the trend reverses or disappears. Trend-following is one of the most used trading strategies used by professionals and will be adopted in the current work. In practice this means that the optimized trading solutions must have a trend-following philosophy and several measures must be employed to enforce that outcome. The principal measure is to adopt genes which encapsulate trading rules with that trading approach. These kind of genes will be designated as “Technical Primitive genes”. Combinations of genes which allow solutions with different trading approaches should be prohibited.

**C. Combining two time frames**

Several authors observed that there is a tendency for results to improve as the time-scale of the used data increases. Using weekly data tends to produce better results than daily data and monthly data tends to produce better results than weekly data [17]. Following this observation, it was decided to combine two time frames in order to capture trends in two different time-scales. They will be referred as Medium Term (MT) and Long-Term (LT). Each of these time frames will produce a different pattern of trades and losses that can potentially reduce the risk and increase the robustness of the solutions. These two components will be implemented in different subsystem of the genomes with a weight of 0.5 each. Instead of using data from two different scales for the two subsystems, a different approach was implemented: only daily data was used but the rules of each time frame use different parameters. The technical indicators used by the rules of the LT time frame have a range of allowed parameters that emulates results using weekly or monthly data. As an example we can think of a moving average with a period multiplied by 5 (number of trading days in a week). To further enforce the differentiation of the two subsystems, they are individually interpreted/simulated and each one must generate an average annualized number of trades inside a pre-defined range. The LT subsystem must generate between 0.0 and 3.0 trades per year and the MT must generate between 3.0 and 8.0 trades per year in average. If the real number is outside this ranges, a progressive penalization is applied to the corresponding genome.

**D. Fitness Function**

A central element in evolutionary algorithms is the scoring and ranking of the solutions in the population. This allows a bias in the selection of solutions for breeding towards the best ones. In the context of optimizing trading rules, the desired characteristics are high returns, low risk and eventually low complexity of the solutions. High returns and low risks are sometimes contradictory and a good fitness function should balance these two elements. In the present work returns are measured as annualized average returns in percentage and risk is measured as the percentage maximum loss (maximum drawdown of returns in percentage of the fixed capital invested).

The fitness score is a modified Risk Return Ratio [18] with the difference that the denominator is never lower than 5 (1). This prevents the optimizer from choosing solutions which almost don’t trade and have a very low maximum loss, producing a high fitness score. This fitness function favors solutions that produce higher returns but do not compromise account capital in the process.

\[
Fitness \text{ score} = \frac{\text{annualized Return \%}}{5 + \text{Maximum Loss \%}} \tag{1}
\]
E. Genes

The genes available to the optimizer is vast but depending on the setup, only a fraction of options is used. The operator genes can be of type Arithmetic (+, -, *, /, Maximum, Minimum), Mathematical (Power, Square Root, Natural Logarithm, Log10, Round, Floor, Ceiling, Absolute Value), Comparison (> , <, >=, <=, ==, !=), Boolean (AND, OR, NAND, NOR, XOR, XNOR, NOT), Technical (Simple Moving Average, Exponential Moving Average, Weighted Moving Average), Majority Vote, If (Double and Boolean). The terminal genes can be of type Constant (e.g. 1.0, 3.14), Fixed (e.g. Open, High, Low, Close), Technical Indicators with fixed parameters (e.g. Stochastic Oscillator, Relative Strength Index, Relative Volatility Index) and Technical Primitive (mini-systems encapsulated in a single gene) which are discussed next.

F. Elimination of similar solutions

The elimination of similar solutions (genomes) is very important because otherwise the most scored ones would gradually create similar versions that would dominate the population. This would strongly decrease the diversity of solutions and increase the likelihood of the optimization being stuck in local maxima’s.

The elimination of similar genomes is not based on the tree structure of the chromosomes since different chromosome trees can produce similar or equivalent results. The chosen method is to look at the Mean Squared Error (MSE) of the vectors with trade signals produced by the genomes, see Figure 4. The algorithm starts at the most scored genome and compares its signals with the ones from the less scored genomes until a distance of N genomes. If the MSE is less than an arbitrary threshold, the less scored are eliminated. After having compared the first genome with all the genomes below up to a distance of N, the second genome is compared to the less scored ones in a similar fashion. The process is repeated with all genomes until the end of the list.

![Figure 4 – Example of algorithm to eliminate similar genomes](image)

The reason why genomes are not compared with all other genomes in the list is that it would be computationally expensive because of the large number of Mean Squared Error calculations. The presented solution seems to be a good compromise between efficacy in removing similar genomes and the computational cost.

G. Technical Indicators and Rules

Technical Primitive genes assume an important role in the solutions found, since they allow for more complex rules to be encapsulated in a single gene which outputs Boolean trading signals. They allow control over the structure and behavior of the rules the optimizer can chose and is a way to control the search space.

1) Moving Average Convergence/Divergence

This technical indicator known as MACD is very popular and can be used in several ways. It is composed by 3 lines but only one is used in the proposed configuration. The main line of this indicator is called the MACD line and is the difference between two exponential moving averages (EMA). A buy signal is generated when the MACD line crosses the zero line from below and a short signal is generated when the MACD line crosses the zero line from above. The periods used to calculate both EMAs are optimizable parameters.

2) Moving Averages

Three different moving averages are used to generate trading rules whose mathematical definition can be consulted in [6]: Simple Moving Average (SMA), Exponential Moving Average (EMA) and Weighted Moving Average (WMA). When generating trading signals from these moving averages (MA), a buy signal is triggered when the price closes above the MA and a short signal is triggered when the price closes below the MA, see blue circles in Figure 5.

![Figure 5 – 100-day Exponential Moving Average applied to the close of the S&P500 index](image)

This method has the drawback of producing successive contradictory trades in consequence of price noise, which decreases profits. An implemented alternative is to use an upper and lower bands around the MA and use those levels to trigger the signals: a buy signal is generated when the price crosses the upper band from below (green circles in Figure 5) and a short signal is generated when the price crosses the lower band from above (red circles in Figure 5). These rules are encapsulated in technical primitive genes and the MA periods are optimizable parameters, as is the distance of the bands in the latter method.
3) Channel Breakout

A popular method of generating trading signals with a trend-following approach is to buy/short when the price closes above/below the maximum/minimum price of the past n days. This method can be consulted in [6] and is implemented in this work in technical primitive genes. The number of trading days used to calculate the channels is an optimizable parameter.

4) Relative Volatility Index

The Relative Volatility Index (RVI) was created by Donald Dorsey in 1993 and refined in 1995 [19]. It’s calculated in a similar way to the Relative Strength Index (RSI) but measures the Standard Deviation of High and Low Prices. It oscillates between 0 and 100, being 50 a neutral value. A buy signal is generated when the RVI line crosses an upper threshold level from below and a short signal is generated when the RVI line crosses a lower threshold line from above. The RVI period and the upper and lower threshold levels are optimizable parameters.

5) Stop-and-Reverse

Rules based on Stop and Reverse indicators (SAR) are usually always in the market, alternating long and short positions. A buy signal is generated when the close price crosses the SAR line from below and a short signal is generated when the close crosses the SAR line from above. When a signal is generated, a new cycle begins and the SAR line is recalculated making a sudden jump from its previous value. For example, when a buy signal is generated, the new SAR line will restart some distance below the close, calculated with a particular algorithm. After that, the SAR line tracks the close price according to another particular algorithm. The two algorithms previously mentioned may change but the logic of the trading rule remains the same. The specific rules of the SAR used in this work can be consulted in [20].

6) Composite Indicators

So far all the mentioned rules are based on a single indicator but it is also possible to combine several indicators into a composite indicator. Each of the single indicator rules focus on a particular characteristic or behavior of the price, like momentum, volatility, price action and when a trend forms, all give a signal, some sooner and some later. The advantage of combining several indicators into a single rule is to filter some of the individual false signals by waiting for the consensus of at least the majority of the single indicators. A practical way to achieve this result is to use majority vote on the outputs of the single indicator rules. This work uses two composite genes. The first has the following components:

- Relative Volatility Index in the configuration previously presented. The RVI period is 52 and the thresholds are 42 and 58.
- Simple Moving Average with a band, crossed by the close, as previously presented (see Figure 5). The SMA period is 290 and the bands are separated by a distance equal to 2.5 times the Average True Range [6] of period 60.
- Based on the Aroon indicator [6], Two 190-day SMAs are calculated from the Aroon up line and from Aroon down line. The Aroon period is 16. Signals are generated when the two SMAs cross.

The second composite indicator uses the following internal components:

- Relative Volatility Index in the configuration previously presented. The RVI period is 35 and the thresholds are 48 and 52.
- Simple Moving Average with a band, crossed by the close, as previously presented (see Figure 5). The SMA period is 60 and the bands are separated by a distance equal to 2.5 times the Average True Range [6] of period 60.
- Based on the Dynamic Momentum Index (DMI) [6] in an equivalent configuration used for the RVI. The DMI uses a period of 14 and the result is smoothed by a SMA of period 30. The threshold levels used to generate signals are 48 and 52.

IV. IMPLEMENTATION

The trading rules were optimized using genetic programming coded in C++. All code was developed with the help of the Integrated Development Environment (IDE) “Qt Creator” version 3.3.0 (based on Qt version 5.5.0). Library MathGL version 2.3 and library FLTK 1.3.3 were used to produce charts with results and the Boost C++ library was used for several auxiliary tasks. The code was developed and run on a Linux machine with a core i3 2350M CPU, 4GB of RAM and running Ubuntu Desktop 14.10 LTS 64 bits.

A. Assets

Stock indices and commodities were the two classes of assets chosen to implement this methodology. A pre-selection of the major world indices and commodities was made, excluding the ones without quality data. From this pre-selection, 19 indices and 11 commodities were chosen. The criteria for indices was to choose the major ones from each continent/country to have a global representation. For the commodities, the criteria was to choose the 4 most traded ones from each of the following categories: Energy, Metals and Agriculture. (Only 3 options were available from the Energy category) [21]. Data from indices was obtained in Yahoo Finance and data from commodities was obtained from the Quandl website. Commodity data consists of continuous futures contracts created from individual contracts. The creation method used was Backwards Ratio Adjusted, with Rollover on the first month. Data begins in 2006 and ends in 2014.
B. Optimization parameters

The optimization is done using in-sample windows of 3 years and out-of-sample windows of 3 months. The initial population has 300 individuals which drops to 100 in subsequent generations. The optimization is stopped if at least 150 generations have passed and if in the last 150 generations the fitness score of the best genome increased less than 5%.

C. Simulation parameters

All simulations of the trading rules are done using a fixed capital of 100 000 monetary units meaning there is no reinvestment of profits. This prevents exponential return curves and helps in the analysis of results. The number of shares/contracts of a trade is calculated dividing the capital by the close price of the asset in the previous day. If the ideal trade size changes by more than 10% the trade size is adjusted. Commissions of 0.1% per trade are considered (open + close = 0.2%). No slippage is considered.

V. RESULTS

This section presents results obtained with different setups. Averages of the individual asset results are presented and also results from portfolios constructed with the individual trading signals obtained. Several performance metrics are calculated and presented in the results tables, namely “Lake Ratio2” which is the inverse of the Lake Ratio [22], “RRR” which stands for Risk Return Ratio [18] and “Max Loss %” which is the Maximum Drawdown in percentage of the fixed invested capital.

A. Setup with a large search space (3 levels)

As a first approach, a setup with a large search space was tested including all available genes. The chromosomes can have a maximum tree depth of 3 but the fitness score is penalized if the depth is greater than 1 to promote simpler solutions. Penalizations increase arithmetically from 0% for 1 level to 10% for 3 levels. The genomes have two subsystems, one focusing on the MT time frame and the other focusing on the LT time frame. Both subsystems have two chromosomes, one to generate long trading signals and the other to generate short trading signals. Both subsystems contribute with a weight of 0.5 to the global output.

The results are presented in Table 3 and show an average annualized percentage return of 1.39% which can be considered poor and suggests the trading signals may be near random. The interpretation we suggest is that the large search space available to the optimizer allows it to find solutions that focus on spurious patterns present in the in-sample window and which don’t repeat in the out-of-sample window. The solutions produced are not robust because they don’t focus on structural patterns of the market. In summary, there is curve-fitting due to the large search space. A different problem with this setup is that it allows for trading rules that don’t use a trend-following approach. Looking in detail at the generated rules, we can see operations between incompatible data which means some rules don’t make sense. It is therefore desirable to explore setups with a smaller search space and restrict the genes used.

B. Setup with only Boolean genes (3 levels)

In order to decrease the search space and restrict the possible gene combinations, all genes with Double inputs/outputs were removed. The allowed genes are AND, OR, Majority Vote, Boolean IF and the Technical Primitive genes (mini trading systems encapsulated in genes which generate Boolean outputs). This allows enforcing of trend-following solutions since these genes already have that philosophy. The search space is therefore reduced to certain areas of interest.

The results are presented in Table 4 and the average annualized percentage return is considerably higher than in the previous setup, 3.91%. Our interpretation for this is that the search space was strongly reduced resulting in less curve fitting and more robust solutions that generalize.

C. Setup with best Technical Primitive genes

So far the results point to the advantage in reducing the search space by limiting the pool of genes and the allowed combinations. In preliminary tests it was possible to confirm that further reducing the available optimizer options, increases returns and the robustness of the solutions. Removing some technical primitive genes which individually produce worse performing solutions also helps increase returns. This path leads to the best setup found and which is presented next. This setup uses maximum simplicity since the maximum allowed depth of the chromosomes is 1, meaning there is only one gene and that gene must be a technical primitive gene.

The allowed genes in this setup are the best performing individually:

- Based on the MACD and RVI technical indicators in configurations previously described in sections III.G.1) and III.G.4) respectively.
- The two composite indicators previously described in section III.G.6).
- A MT gene which has some internal complexity and was previously created by the authors. It uses a volatility calculation based on the difference between an upper and lower price bands. The upper band is calculated with the past high peak values and the lower band is calculated with past low peak values. A buy/short signal is generated when the present volatility line crosses a past volatility reference.

The genomes use two subsystems, one for the LT and the other for the MT and both contribute with a weight of 0.5 to the final outcome. Each subsystem has a single chromosome which generates both long and short trading signals. The average results from all assets is presented in Table 5.

Results show an average annualized percentage return much higher in this setup, 8.08%, and an increase in the other...
performance ratios. A negative evolution in this setup is a higher average maximum percentage loss.

To graphically exemplify the results produced by this setup the graphs with simulations in two assets are presented in Figure 6 and Figure 7. The horizontal scales shows the dates and the vertical scales shows the percentage return. Each figure has a black line representing the asset price percentage variation and a red line representing the percentage return. On the top, three sub-windows represent the trading signals: Medium Term (MT), Long Term (LT) and Final (F). Each of these trading signals range from -1.0 to 1.0 corresponding to being 100% short or 100% long, respectively. The Final trading signal is calculated by averaging the MT and LT signals with equal weights of 0.5.

Figure 6 shows the results of one optimization with this setup in the S&P500 index from 2006 to the end of 2014. We can observe the returns follow the price variation closely until mid-2008, when the price starts to fall violently and the trading system profits by being short. This results in a very positive performance in the 2008 bear market. Between 2010 and 2012 S&P500 had an upward bias but with some corrections which translated into the trading system having difficulties, alternating periods of profits and losses. After 2012, the index entered a period of steady gains and the trading system was able to consistently profit again. In summary, the trading system was able to profit when the market had a clear direction and was able to, not only prevent big losses in the 2008 bear market, but strongly profit in that period. The average annual return of the trading system in the 9 year period was around 9%.

As an example, a genome is presented from an optimization of S&P500. It is the best genome of generation 150 from the last optimization window (35), corresponding to the period between 2011/10/04 and 2014/10/06.

**Subsystem 0**

**Chromosome 1**
MACD>0 (186, 400)

**Subsystem 1**

**Chromosome 3**
RVI>X (37, 48.7321, 53.7321)

Chromosome indices are unique and can have any order depending on what is defined in the configuration files (in this case chromosomes with indices 1 and 3 are used). Each chromosome has a tree with a single technical primitive gene in the root node. In the LT chromosome, signals are generated by a MACD Line when it crosses the 0 level. The MACD line is calculated from Exponential Moving Averages (EMA) with periods 186 and 400. In the MT chromosome, signals are generated by a Relative Volatility Index (RVI) with period 37 when its value crosses the upper threshold line of 53.7321 or the lower threshold line of 48.7321.

Figure 7 shows the graphical results of one optimization in Brent, which is an asset that produces very good results with the trading rules generated with this method and setup. We can observe the returns follow the price change until mid-2008 where Brent initiates a very sharp fall. The system quickly reverses to a short position and strongly profits from the fall that ends in the beginning of 2009. After that, Brent has a more erratic behavior despite having an upward bias, and the trading system only manages a small gain in that period (until mid-2014). After mid-2014 Brent starts a very strong fall and the system is again able to strongly profit from the movement. In average, the trading system was able to gain 27% per year. In this case it becomes very clear that the system specializes in following strong trends and earns most of its profits from those movements, resulting in a return curve that can be almost flat for several years and then have sudden jumps. This means it’s wise to use these type of systems in a portfolio in order to have a smoother return curve and less risk.

**D. Portfolio with best setup**

This section will analyze the results of a portfolio simulation using the individual trading signals generated with the setup described in section V.C. All the trading signals are synchronized and each asset receives an equal share of the capital to trade those signals. The capital is fixed and equal to 100 million monetary units. Commissions and slippage are equal to those used in the individual simulations.

Table 6 shows the summary of results and Table 7 presents the annualized percentage returns by year. As can be observed, this portfolio has a much better performance in the 2008 bear market than the reference (all-bought equal-weight portfolio). Not only it didn’t lose money but was able to strongly profit from the fall observed in the majority of assets. The Maximum
loss was 16.69% of the invested capital and happened in the 4th of November, 2008. Analyzing the results by year, we can see that the results were very good until the end of 2011. The next year, 2012 was negative and the following years were positive but with lower returns. In fact, the last 3 years were not able to recover from the 2011 loss, which may be explained by a different phase of the market with more erratic movements from the majority of the assets.

Table 6 also presents the results of the best portfolio but using only the MT or LT subsystems and what can be observed is that the annualized average percentage return is similar in the two time frames but the maximum average percentage loss is higher. This indicates that combining the two time frames reduces risk by combining two different profiles of trades and losses.

We can conclude that this portfolio has an interesting annualized average return in the 9 years of the simulation with a relatively low risk, but the returns were heterogeneously distributed. There is no disadvantage in having very high performing years but there is a disadvantage in having negative or poorly performing years.

Figure 8 shows the returns obtained with this portfolio (red line) and the black line corresponds to an all-bought reference portfolio. This reference portfolio uses equal weights for all assets but is not exactly a Buy-and-Hold portfolio because the trade sizes are periodically adjusted to reflect the ideal sizes (all simulations use fixed capital). The adjustments are made if the difference between the real size and ideal size is greater than 10%.

<table>
<thead>
<tr>
<th>Portfolio</th>
<th>An. Ret %</th>
<th>Win %</th>
<th>Avg Loss %</th>
<th>Max Loss %</th>
<th>ML Date</th>
<th>Std. Dev.</th>
<th>LR2</th>
<th>RRR</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>All-bought (reference)</td>
<td>5.21</td>
<td>76.67</td>
<td>-9.27</td>
<td>-59.94</td>
<td>21/11/2008</td>
<td>0.00936</td>
<td>0.035</td>
<td>0.087</td>
<td>5.614</td>
</tr>
<tr>
<td>Best setup (MT+ LT)</td>
<td>7.81</td>
<td>40.04</td>
<td>-5.96</td>
<td>-16.69</td>
<td>04/11/2008</td>
<td>0.00631</td>
<td>0.088</td>
<td>0.468</td>
<td>8.705</td>
</tr>
<tr>
<td>Best setup (only MT)</td>
<td>6.28</td>
<td>33.68</td>
<td>-8.44</td>
<td>-19.49</td>
<td>31/12/2014</td>
<td>0.00678</td>
<td>0.058</td>
<td>0.322</td>
<td>6.816</td>
</tr>
<tr>
<td>Best setup (only LT)</td>
<td>7.85</td>
<td>36.33</td>
<td>-7.41</td>
<td>-21.21</td>
<td>18/05/2009</td>
<td>0.00689</td>
<td>0.070</td>
<td>0.370</td>
<td>8.592</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS

The results obtained confirm that trend patterns in markets can be successfully identified and explored in order to obtain profits. The portfolio simulation using the best setup produced an annualized percentage return of around 8% with a maximum loss percentage of around 16% in the period 2006 – 2014 which must be considered as positive. This results also confirm that technical analysis can be used to predict the
future behavior of the markets to some extent, in particular using technical indicators.

This work combined rules with two time frames, MT and LT and demonstrated the improvement that can be obtained by such approach. The returns of the combined rules are similar to the returns obtained by the single time frame components but, the maximum loss is significantly lower. This is the result of having two sets of rules with different trade profiles and different loss profiles, which tend to balance each other and reduce risk. It’s a form of diversification: trading strategy diversification.

Several measures were successfully employed to reduce curve-fitting of the solutions and improve robustness. This was done by reducing the search space, eliminating solutions which didn’t use a trend following philosophy. In practice this was achieved by only using genes that produced trend-following rules. The complexity of the solutions was also reduced by limiting the chromosome trees maximum depth.

A novel algorithm of deleting similar solutions was proposed which doesn’t look at the chromosome tree similarities but compares the Mean Squared Error (MSE) of the vectors with trading signals generated by the solutions.

A special type of complex gene was proposed (technical primitive) which combines output simplicity with internal complexity. The signals generated by these genes are simple in the sense that they are Boolean and follow a very simple logic of detecting and following a trend. But the way to achieve that end is more complex than common trading rules since they may include several sub-rules combined by majority vote. This approach has the advantage of not increasing the search space since the gene doesn’t necessarily have many optimizable parameters and, allows for more sophisticated rules. Majority vote also tends to suppress false signals generated by the sub-rules and the output signals tend to be more stable, with some immunity to noisy price action.

REFERENCES