Expert Systems With Applications 46 (2016) 307–315

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Robust technical trading strategies using GP for algorithmic portfolio selection

José Manuel Berutich [a,*], Francisco López [a], Francisco Luna [a], David Quintana [b]

[a] Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Málaga, Spain
[b] Departamento de Informática, Universidad Carlos III de Madrid, Leganés, Madrid, Spain

ARTICLE INFO

Keywords:
Genetic programming
Algorithmic trading
Portfolio management
Trading rule
Finance

ABSTRACT

This paper presents a Robust Genetic Programming approach for discovering profitable trading rules which are used to manage a portfolio of stocks from the Spanish market. The investigated method is used to determine potential buy and sell conditions for stocks, aiming to yield robust solutions able to withstand extreme market conditions, while producing high returns at a minimal risk. One of the biggest challenges GP evolved solutions face is over-fitting. GP trading rules need to have similar performance when tested with new data in order to be deployed in a real situation. We explore a random sampling method (RSFGP) which instead of calculating the fitness over the whole dataset, calculates it on randomly selected segments. This method shows improved robustness and out-of-sample results compared to standard genetic programming (SGP) and a volatility adjusted fitness (VAFGP). Trading strategies (TS) are evolved using financial metrics like the volatility, CAPM alpha and beta, and the Sharpe ratio alongside other Technical Indicators (TI) to find the best investment strategy. These strategies are evaluated using 21 of the most liquid stocks of the Spanish market. The achieved results clearly outperform Buy&Hold, SGP and VAFGP. Additionally, the solutions obtained with the training data during the experiments clearly show during testing robustness to step market declines as seen during the European sovereign debt crisis experienced recently in Spain. In this paper the solutions learned were able to operate for prolonged periods, which demonstrated the validity and robustness of the rules learned, which are able to operate continuously and with minimal human intervention. To sum up, the developed method is able to evolve TSs suitable for all market conditions with promising results, which suggests great potential in the method generalization capabilities. The use of financial metrics alongside popular TI enables the system to increase the stock return while proving resilient through time. The RSFGP system is able to cope with different types of markets achieving a portfolio return of 31.81% for the testing period 2009–2013 in the Spanish market, having the IBEX35 index returned 2.67%.

1. Introduction

Algorithmic trading using evolutionary computation has been a hot topic of research in the recent years for academics from both finance and soft-computing domains with a large number of published research articles (Aguilar-Rivera, Valenzuela-Rendón, & Rodríguez-Ortiz, 2015; Hu et al., 2015). Normally, it is very hard for a simple investor to optimize his investments without requiring the skills of financial advisers. The main goal of this work is to provide an application which helps investors achieve a significant profit on buying and selling financial securities in an automatic way without requiring the help of portfolio managers.

Selecting the most promising securities is a very difficult problem for Genetic Programming (GP) due to the dynamic and stochastic nature of the markets and the vast quantities of data that needs to be analyzed.

In this paper we are interested in developing robust technical trading rules using GP which can replace the intervention of human money managers, and be applied systematically to manage a portfolio of stocks. One of the most important benefits of systematic trading is that it helps to remove emotional decision making from the investment process, as emotions can easily overwhelm rational decision making. This can be lessened to a large extent by having a system that automatically makes the decisions for you.

Another important benefit of systematic strategies is that they can be tested on historical data. This ability to simulate a strategy is one of the biggest benefits of systematic trading. Back-testing tells you how well the strategy would have done in the past. While back-tested performance does not guarantee future results, it can be very helpful

* Corresponding author. Tel.: +34619266536.
E-mail addresses: jmberutich@gmail.com (J.M. Berutich), valverde@lcc.uma.es (F. López), flv@lcc.uma.es (F. Luna), dquintan@inf.uc3m.es (D. Quintana).

http://dx.doi.org/10.1016/j.eswa.2015.10.040
0957-4174/

when evaluating potential strategies. Back-tested results can be used to filter strategies that either do not suit the required investment style or are not likely to meet risk/return performance goals.

One of the biggest difficulties GP evolved strategies face is over-fitting. While solutions perform well in the training dataset, once they are tested out-of-sample with new data, their performance is seriously degraded. We will explore a method for reducing over-fitting of GP solutions. Robust GP solutions should display similar behaviour during out-of-sample testing as during training. Moreover, GP investment strategies need to be robust in order to be deployable in a real portfolio management situation.

1.1. Robustness

The term *"robust"* has many definitions depending on the author. It can be broadly defined as the ability of a system to preserve its functionality despite internal *(genotypic robustness)* or external perturbations *(phenotypic robustness)* (Branke, 1998; Soule, 2003).

1.1.1. Genotypic robustness

Genotypic robustness aims at achieving insensitivity of fitness to perturbations from genetic operators. Soule (2003) finds that the *code bloat* phenomenon in GP, where an increase of the size of the trees does not result in fitness improvement, is a redundancy mechanism. Trees grow *introns* which safeguard valuable code and protect it against loss during crossover or mutation. Even though this approach favors broad plateaus instead of peaks of high fitness, it is of negligible use when the surface of the search space changes.

1.1.2. Phenotypic robustness

Phenotypic robustness deals with resilience to external changes and can be categorized into:

- *Generalization Robustness (GR):* Robustness as the generalization ability of evolved solutions. From a machine learning standpoint, it is the predictive accuracy of a learner for new unseen cases. The objective here is reducing over-fitting and producing solutions whose performance is similar for both in-sample and out-of-sample datasets (Kushchu, 2002).
- *Environmental Robustness (ER):* Robustness to external environmental perturbations. Financial markets suffer abrupt structural changes which tend to persist in time. (Granger & Hyung, 2004). Robust GP solutions should withstand periods of extreme volatility and trend change. Yan and Clack (2010) propose training on three distinct scenarios; a *bull* or rising market, a *bear* or falling market and a volatile *sideways* market.
- *Robustness to Noise (RN):* Robustness to noise inherent in the data or the readings produced by the system (Kitano, 2004).
- *Self Repair (SRS):* Robustness as the ability to self-repair after severe phonotypic damage (Bowers, 2006).

In this paper we are concerned with the generalization and environmental robustness of evolved solutions. We explore how we can design solutions that display similar performance for both in-sample and out-of-sample data, as well as solutions that can resist abrupt trend changes and extreme volatility periods.

Our approach is substantially different to previous work and is centered around how we calculate the fitness function. Our main contribution lies in evaluating the fitness using a random sampling method which will explain later in Section 3.4.

The main contributions of this study can be summarized as follows: (1) The use of a random sampling mechanism to divide the time series of a basket of stocks into segments without requiring user intervention, or any unsupervised machine learning method to group segments into the distinct market conditions (bull, bear and sideways markets). (2) The use of a robust fitness function that uses all sampled segments to calculate an overall fitness score across random market conditions reducing over-fitting of solutions. (3) The use of different financial metrics never used together with technical indicators in previous research such as the returns, the moving average of returns, the Capital Asset Pricing Method (CAPM) alpha and beta, the Sharpe ratio and the volatility of the stocks calculated over different period lengths. (See Table 2).

Consequently, in this study a robust GP evolutionary approach will be presented to automate buying and selling decisions in order to maximize the Sterling ratio (total return divided by maximum drawdown). The proposed method will be tested on a basket of 21 stocks from the Spanish market using 13 years of daily price data and compared to the IBEX35 market index, the results will be analyzed and some possible conclusions will be discussed.

The remainder of the paper will be organized continuing in the next section with the most relevant previous work followed by the algorithmic approach, experiments performed and discussion of the results obtained. We finalize with our conclusions and future work.

2. Related work

GP was first employed by Allen and Karjalainen (1999) for technical trading rule discovery. The dataset used in their experiments was the S&P 500 index using daily prices from 1928 to 1995. Their results demonstrated that although GP could find profitable trading rules, it failed to produce excess-returns over the passive strategy of *"Buy & Hold"* (B&H), which consists in buying on the first evaluation day and selling on the last.

Neely (2003) extends the previous work by Allen and Karjalainen (1999) using a risk adjustment selection criteria to generate rules with the hope of improving performance. However, the results show no evidence that the rules significantly outperform B&H on a risk-adjusted basis.

Becker and Seshadri (2003) present results of GP-evolved technical trading rules, which outperform a buy-and-hold strategy on the S&P 500 after taking into account transaction costs. They introduce several changes to the original work of Allen and Karjalainen (1999), which include a complexity-penalizing factor, a fitness function that considers consistency of performance, and co-evolution of separate buy and sell rules. Monthly data is used instead of daily.

Lohpetch and Corne (2009) replicate the work of Becker and Seshadri (2003) and the authors find that the results are sensitive to the data periods chosen for the experiments. Their results are improved by using a validation set, used for choosing the best rule found during training.

Mallick and Lee (2008) used GP to find trading rules on the thirty component stocks of the Dow Jones Industrial Average index. The authors find Statistical evidence of outperforming B&H in falling markets, and confirm that GP based trading rules generate a positive return under bull (rising) and bear (falling) markets.

Yan and Clack (2010) use GP for building a symbolic regression expression that measures the attractiveness of each stock; Each month a portfolio is constructed with the most attractive stocks according to the GP model. The portfolio is a market neutral long/short portfolio of Malaysian equities. The authors propose two approaches for evolving robust trading rules. First by splitting the training dataset into three extreme environment periods: up, down and sideways volatile. Secondly instead of using just one solution, a voting comity is used, formed by the three best solutions trained on each of the extreme environments. The authors show results that considerably beat the benchmark index, but the results have a significant caveat, i.e. they used a small out-of-sample period (July 1997–December 1998), which is before the training period (January 1999–December 2004). Monthly data was used to simulate portfolio, meaning at the beginning of a month the stocks which the system recommends are bought, and at the end of the month the position is reassessed.

In this paper, we use a similar approach to Yan and Clack (2010), as we think that *which* data and *how* it is presented is crucial for any machine learning to occur; after all you can only learn what's on the data. But our approach substantially differs as we use a random sampling method at the GP individual level instead of hand-picking different bull, bear and volatile scenarios for training. Secondly we treat the problem as classification problem instead of symbolic regression. Our GP expression returns a boolean value that we interpret as a trading signal. One of the main advantages of our proposed method over Yan and Clack (2010), is that our random sampling method does not require any user intervention in order to divide the stock price time series. Another important advantage of randomly sampling the time series is increasing the robustness of the solutions evolved. Lastly, our method uses daily data instead of monthly data, hence it is quicker to react to abrupt changes in market conditions.

Hsu (2011) use a hybrid Self-Organizing Map (SOM) GP system where the SOM unsupervised neural network is used to cluster the time series into similar segments. These segments are then used by the GP system to learn trading rules for each of the market conditions detected by the SOM. During testing the SOM is used to classify the unseen time series and select the best GP solution found during training on similar time series. Our method has some advantages over this method, as it does not require the use of any unsupervised method to group similar time series segments and thus is less computationally expensive. Another important advantage of our method is that it provides a single robust solution that works well in all market conditions.

Mousavi, Esfahanipour, and Zarandi (2014) use a dynamic GP portfolio trading system based on the technical indicators. The authors extend the classical GP algorithm to a multi-tree GP forest that is able to extract multiple trading rules, one for each of the assets considered. Since the traditional GP structure is not able to cope with this specific problem, the consequent parts of the rules are designed as a crisp function of the weights of the stocks in the portfolio. The fitness measure employed in this study is the conditional Sharpe ratio, a modification of the original Sharpe ratio, using CVAR (Conditional Value at Risk) as the divisor instead of the standard deviation of returns. The system was trained and tested on 15 stocks from the Iranian Stock Exchange and 15 stocks from the Toronto Stock Exchange with a sliding window approach using 4.5 years of daily stock prices. Our method has some advantages, firstly being simpler, as a single rule is evolved to trade all assets in the portfolio, it is less computationally expensive, and secondly, evolving a single trading rule that can be applied to all the stocks, increases the robustness of solutions. Thirdly our method has been trained on 8 years of data and tested during the next 5 years, proving its robustness over an extended period of time.

Gypteau, Otero, and Kampouridis (2015) use an intrinsic time scale based on directional changes (DC) combined with Genetic Programming to find an optimal trading strategy that forecasts future price moves. A DC event is identified by a change in the price of a given stock greater than a predefined threshold value, which was in advance decided by the user. The authors use total return as the fitness measure and use two stocks from the UK market, and the NASDAQ and NYSE indices to evolve their solutions over a period of 1000 days for training and 500 days for testing. Their results showed that the strategies evolved by GP are more profitable when using multiple threshold values than using a fixed threshold value, providing evidence that DC can be used for forecasting and that combining multiple thresholds is beneficial. In comparison to our proposed method, this method only uses DC and does not consider other technical or financial metrics on a very limited selection (two stocks and two Indices) of assets, and does not compare the results obtained with other traditional strategies such as Buy & Hold, making the results obtained in the study hard to interpret.

Pinto, Neves, and Horta (2015) use a dual-objective genetic algorithm to maximize the total return on investment (ROI) while minimizing the standard deviation of returns. In their study they introduce the VIX volatility index and other technical indicators to boost performance of trading strategies, using the most important world indices to optimize the parameters of the strategies. This approach is able to avoid serious market declines, while producing a Pareto front of non-dominated solutions that can cater from the most conservative types of investors looking for strategies with minimal risk to the most aggressive ones, who prefer higher returns at a higher risk. Our proposed method has some advantages, first our method learns trading rules, and not only optimizes the parameters of some predefined rules, but is able to construct rules and choose the best parameters as we employ a GP system instead of a GA. Secondly our GP system learns the trading rules from a basket of stocks, instead of a single market index, exposing the evolutionary process to more data and different market conditions, as some stocks can be in a bullish trend while others are bearish or range bound (sideways), thus producing solutions that are more robust, and able to cope with extreme market conditions.

Luengo, Winkler, Barrero, and Castao (2015) manually divide the stock price time series into three segments of 4 years which sometimes overlap, and then this previous segments are again divided into three different periods, 1 for pre calculating the technical indices, 2 for GP training and 3 for testing. In comparison our approach has some advantages as we can use the best evolved ruleunder all market conditions proving to be resilient to regime switches in the data. It also has the advantage of not requiring human intervention in manually dividing the time series into distinct market regimes and providing a robust solution that produces good results in all market environments.

One possible weakness of our method in comparison with the references previously discussed, is that it is more computationally expensive, as the fitness has to be calculated over more data than other methods who only use the whole time series for rule discovery, or methods that divide the time series into bullish, bearish and sideways trends, as randomly sampling the time series might produce some overlap between segments.

Lastly, readers are keenly directed towards two recent surveys dealing with evolutionary algorithms and trading strategies. Hu et al. (2015), Aguilar-Rivera et al. (2015)

3. Algorithmic approach

We are interested in exploring how to evolve robust GP solutions. Which techniques can we employ to steer away solutions from being over-fitted. Ideally solutions should present similar performance for both in-sample and out-of-sample datasets.

3.1. Genetic programming

Genetic Programming (GP) pioneered by Koza (1992) is an extension of the original Genetic Algorithm (GA) introduced by Holland (1975). GP automatically generates expressions that are executable and have a variable length representation in the shape of a tree structure. Expressions are formed by combining *functions* and *terminals*. The function set contains the primitives which are used to form the expressions, and sit in the branches of the tree. These can be arithmetic, boolean, or any user-defined functions. The terminal set contains the values that rest in the final nodes hanging from the branches. These terminal values can be the independent variables of the problem, *ephemeral* constants (randomly generated constants generated at the initialization of the tree expression) or functions that lack arguments. Terminals might also serve as the parameters to these functions.

GP follows the same evolutionary approach as GAs. A Random population is initialized. GP has different random approaches for initialization, the *full* and *grow* methods, and a combination of both,

Ramped half-and-half, which is the most widely used method because is able to generate a wider variety of sizes and shapes than the afore mentioned (Poli et al., 2008).

The performance of GP solutions is measured by the *Fitness* function. Parents are probabilistically selected based on their fitness, hence better parents have a higher chance of reproducing. The crossover operator mixes the genetic content of both parents and creates offspring solutions. Mutation is another typical operator employed in GA/GP to introduce small genetic variations to a minor fraction of the population.

3.1.1. Description of the GP system

We approach the problem as a binary classification problem. A GP rule is evolved using the functions and terminals provided and evaluates to a boolean that we interpret as a buy or sell signal.

We employ a $\mu + \lambda$ evolutionary process where from μ parents we generate λ offspring and the best individuals from both μ and λ form the population of the next generation. $\mu + \lambda$ has the advantage of not losing the best solutions during evolution as they are never replaced by inferior individuals.

We utilize strongly-typed GP which allows for the declaration of data types of functions and terminals, and offers the advantage of limiting the search-space to syntactically valid expressions only.

3.2. Portfolio simulation

To evaluate the GP evolved rules we simulate a long only portfolio of the 21 largest and most liquid Spanish stocks. We choose the Spanish market as the testing period from 2009 to 2013 has been particularly volatile, especially in the summer of 2012 with the outbreak of the sovereign debt crisis in Europe. We use as the reference benchmark the IBEX35 index.

Portfolio returns are calculated in the following manner; at the initial evaluation period, the portfolio starts with W_0 in cash. At every day the GP rule is evaluated and the stocks which are classified as True are bought (or maintained if already in the portfolio), and those classified as False are sold if owned in the portfolio. The total portfolio value W_t is calculated daily by valuing the shares at the closing price of each day plus the value of the cash account. We do not use leverage or reinvest profits and each purchase is allocated a fixed amount of cash of 10,000.00 EUR. Transaction costs of 0.3% are included in the calculations.

3.3. Fitness function

There are a wide variety of metrics for assessing the performance of trading strategies, being the most widely used the Sharpe ratio, the Sterling ratio and total return (Iba & Aranha, 2012). The Sharpe ratio provides a risk-adjusted measure of the performance of a portfolio, and has been used in Adamu and Phelps (2010), Yan and Clack (2010), Lohpetch and Corne (2009), Becker and Seshadri (2003), Allen and Karjalainen (1999).

$$\textbf{Sharpe Ratio} = \frac{\mu - RF}{\sigma}\sqrt{n} \tag{1}$$

Where μ is the mean of the portfolio returns, RF is the risk-free rate, σ is the standard deviation of portfolio returns and n the number of observations. The Sharpe ratio even though has been widely used in previous literature is not the ideal fitness measure as we will explain next.

3.3.1. Sharpe ratio is not the ideal fitness measure

The main drawback of the Sharpe Ratio is that it was not designed to handle negative portfolio values. During the course of evolutionary search, rules that are not very good are generated and when evaluated generate negative portfolio values. This distorts the evolutionary search by masking poor solutions as better individuals. Let us suppose we have two strategies, one year of observations (255 trading days) and a risk-free rate of 0%:

$$\textbf{Strategy A}\ ; \frac{\mu = -0.0003}{\sigma = 0.01}\sqrt{255} = -0.4790 \tag{2}$$

$$\textbf{Strategy B}\ ; \frac{\mu = -0.0001}{\sigma = 0.0005}\sqrt{255} = -3.1937 \tag{3}$$

Clearly strategy B is better than A as it has a higher mean return μ and lower volatility σ, but it has a worse Sharpe ratio. This problem negatively affects the evolutionary search process, as better solutions are replaced with worse ones.

3.3.2. Sterling ratio

We employ a more suitable risk-adjusted metric, the Sterling ratio to measure the fitness of individuals. The Sterling ratio is not as widely used in the literature for evolving trading strategies as the Sharpe ratio, but it does not have the problem previously mentioned. Dempster and Jones (2001) and Zhang and Ren (2010) used it as the fitness measure in their systems.

$$\textbf{Sterling Ratio} = \frac{\text{Total Return}}{\text{Maximum Drawdown}} \tag{4}$$

Total return is calculated as the percentage difference between the final and initial portfolio value. Maximum drawdown is the maximum decline in portfolio value from peak to nadir measured as return.

3.4. Randomly sampled fitness (RSFGP)

We are concerned with endowing generalization and environmental robustness to the solutions evolved by GP. We achieve this by exposing the individuals to random market situations sampled from the original dataset.

Instead of evaluating the fitness of the individual in the whole training dataset, our approach consists in selecting n random segments $\{s_1, s_2, \ldots, s_n\} \in S$ from the whole training dataset S and calculating the fitness on each of the segments. Assuming I_i is an individual in the population of solutions, and $f_{I_i}^{s_j}$ is the fitness of the individual I_i on segment s_j The final fitness $\overline{F}_i$ is calculated as the mean fitness obtained in the n randomly sampled segments of S. The random sampling is done with replacement at the individual level, i.e. each individual is always evaluated on different randomly selected segments of the original dataset.

$$\overline{F}_i = \frac{1}{n}\sum_{j=1}^{n} f_{I_i}^{s_j} \tag{5}$$

We have chosen $n = 100$ for the number of segments while the length is set at one trading year or 255 days. If the trading rule does not generate any buy or sell signals for that section, a penalty is used which sets the fitness value to −9.99. This is done in order to penalize solutions do not trade and would have a fitness value of 0 and forces evolution to choose poor solutions with low fitness values (but with some trading activity) over solutions that did not trade.

3.5. Volatility adjusted fitness (VAFGP)

Yan and Clack (2010) use a volatility adjusted fitness that is quite similar to our random sampling method but using the standard deviation of the fitness as the divisor. See Eqs. 6 and 7. In the experimental results of Section 4.4.3 we also study the effects of using a volatility

Table 1
Stocks used in this study.

Symbol	Company name
ABE.MC	Abertis Infraestructuras
ACS.MC	Actividades de Construcción y Servicios
ANA.MC	Acciona
BBVA.MC	Banco Bilbao Vizcaya Argentaria
BKT.MC	Bankinter
EBRO.MC	Ebro Foods
FCC.MC	Fomento de Construcciones y Contratas
GAM.MC	Gamesa Corporación Tecnológica
GAS.MC	Gas Natural SDG
IBE.MC	Iberdrola
IDR.MC	Indra Sistemas
JAZ.MC	Jazztel
MAP.MC	Mapfre
OHL.MC	Obrascon Huarte Lain
POP.MC	Banco Popular
REE.MC	Red Electrica Española
REP.MC	Repsol
SAN.MC	Banco Santander
SCYR.MC	Sacyr
TEF.MC	Telefónica
VIS.MC	Viscofán

adjusted fitness and compare it to our proposed random sampling method.

$$\sigma = \sqrt{\sum_{j=1}^{n} \frac{(f_{I_i}^{s_j} - \overline{F}_i)^2}{n}} \tag{6}$$

$$\textbf{Volatility adjusted fitness} = \frac{\overline{F}_i}{\sigma} \tag{7}$$

4. Experimental results

We compare the effects of using randomly sampled subsets from the original dataset, to a standard GP method where the whole training dataset is used. We also study if including the the standard deviation in the fitness as Yan and Clack (2010) do using a volatility adjusted fitness is beneficial.

4.1. Dataset used

Our dataset consisted on 14 years of daily prices (Open, High, Low, Close, Volume) adjusted for splits and dividends obtained from Reuters. Table 1 shows the Reuters symbol and company name of the stocks used. We used 21 of the largest and most liquid stocks for the Spanish Market for which we had data for the 14 years of our study. The period from January 2000 to December 2008 is used for searching for the optimal GP rule while the out-of-sample testing period used starts on January 2009 and ends on December 2013. We use the IBEX35 index as the reference benchmark.

From the daily prices dataset we compute 264 company specific features, see Table 2. These features are used during the evolution as terminals in the GP tree. The first 200 days of the year 2000 are used to compute the initial technical indicators, hence they are not included in the training results.

Table 3
GP parameter settings.

Algorithm type	Strongly-typed $\mu + \lambda$
Population size	1000
Initialization	Ramped half and half
Function set	$+, -, *, /, <, >$, and, or, if-then-else, isBetween
Terminal set	264 Company specific features (See Table 2)
Crossover operator	Single point crossover
Crossover fraction	80%
Mutation operator	Uniform mutation
Mutation fraction	10%
Max. initial tree depth	6
Termination	100 generations

4.2. GP parameter settings

Table 3 shows summary of the GP parameters and operator functions used in our experimentation. These parameters, which were decided upon after some preliminary testing, are held constant during all our experiments.

4.3. Computational environment

Our test machine consisted of a dual Intel Xeon E5-2687W @ 3.10 GHz workstation running Ubuntu 12.04.04 Linux with 64Gb of RAM. We implemented our GP system and portfolio simulation using Python 2.7.3 and Distributed evolutionary algorithms in Python (DEAP) (Fortin, De Rainville, Gardner, Parizeau, & Gagné, 2012).

4.4. Experiments

We execute 30 independent runs in all of our experiments, and measure the robustness of solutions using shrinkage (Berutich, Luna, & López, 2014; Mehta & Bhattacharyya, 2004). Shrinkage is calculated as the percentage change in performance between training and testing data. In order to better assess the performance of the evolved strategies we include together with the sterling ratio fitness metric, the total return and the Sharpe ratio. Likewise, we analyze the mean daily returns and volatility of the portfolios generated by the various methods. Statistical analysis has been conducted on the results at the 5% significance level.

4.4.1. Standard GP

We use a standard genetic programming (SGP) approach as the basis of comparison between the random sampling fitness (RSFGP) and the volatility adjusted fitness methods (VAFGP). In SGP the fitness of the individual is calculated on the whole training dataset.

Table 2
Description of features in the terminal set.

1. Return from N periods	$N = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200$
2. Simple moving average of returns (N periods)	$N = 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200$
3. Exponential moving average close (N periods)	$N = 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200$
4. Bollinger bands (N periods)	$N = 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200$
5. Internal bar strength (N periods)	$N = 0, 1, 2, 3 \ldots, 30$
6. Relative strength index (N periods)	$N = 5, 6, 7, \ldots, 30$
7. Volatility (N periods)	$N = 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200$
8. CAPM α (N periods)	$N = 10, 11, 12, \ldots, 60$
9. CAPM β (N periods)	$N = 10, 11, 12, \ldots, 60$
10. Sharpe ratio (N periods)	$N = 10, 13, 16, 19, \ldots, 60$

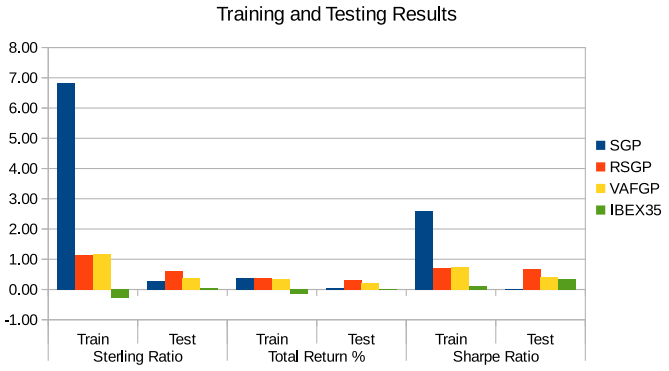

Fig. 1. Mean results obtained for training and testing datasets.

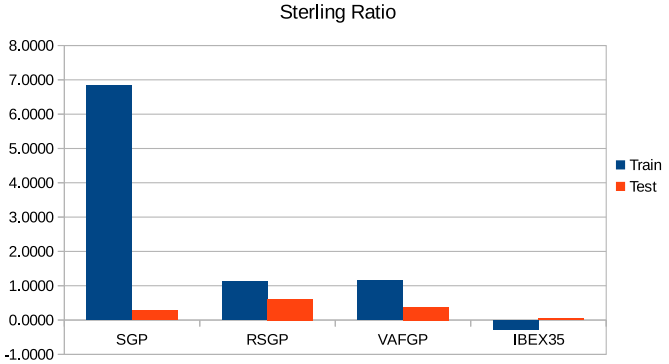

Fig. 2. Mean sterling ratio obtained in all executions.

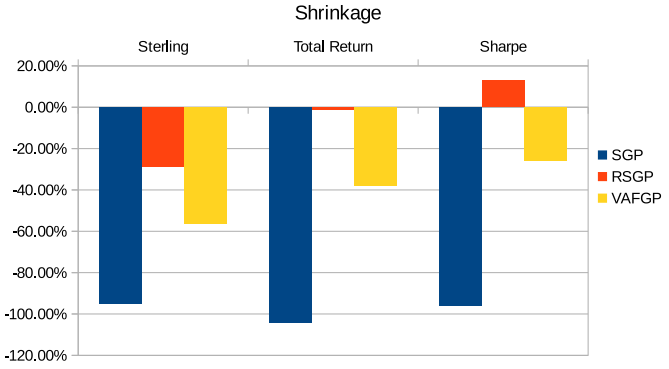

Fig. 3. Mean shrinkage between training and testing datasets.

Table 4
Shrinkage between training and testing results.

	Sterling (%)	Total return	Sharpe (%)
SGP	−94.94	−104.02	−96.07
RSFGP	−28.84	−1.29	13.03
VAFGP	−56.10	−38.12	−25.69

Table 5
Mean results obtained in the study.

	Sterling ratio		Total return		Sharpe ratio	
	Train	Test	Train (%)	Test (%)	Train	Test
SGP	6.8398	0.2723	37.08	5.79	2.5877	−0.0175
RSFGP	1.1235	0.6124	38.83	31.81	0.7103	0.6933
VAFGP	1.1584	0.3809	35.76	19.85	0.7284	0.4154
IBEX35	−0.2708	0.0521	−13.65	2.67	0.1223	0.3389

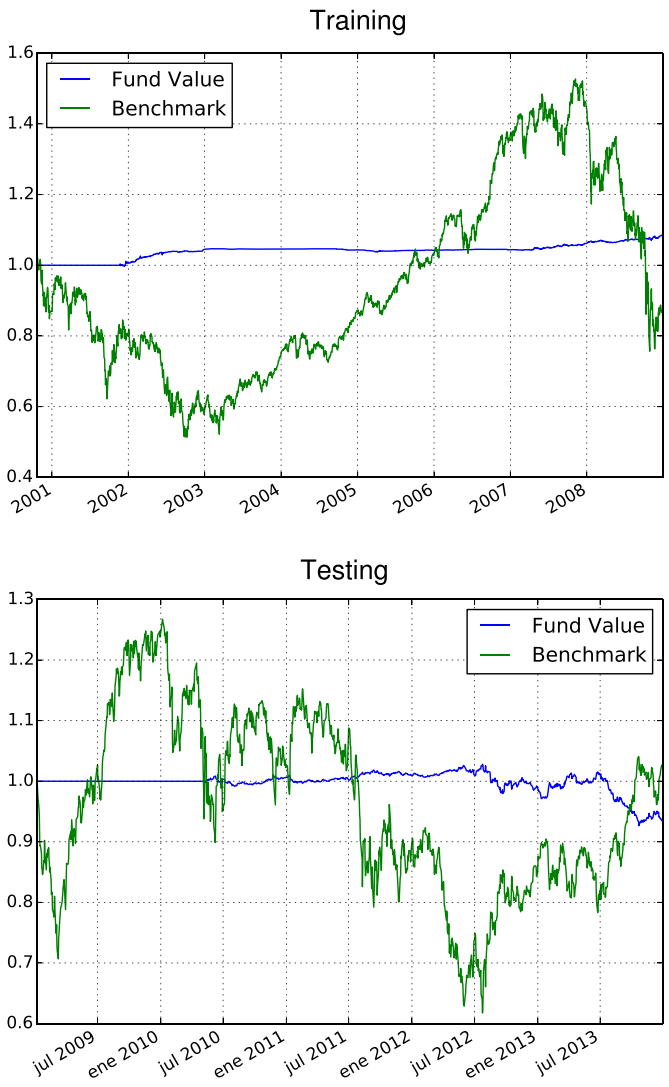

Fig. 4. Training (top) and out-of-sample testing (bottom) performance of an over-fitted SGP portfolio.

As we can see in Figs. 1 and 2, SGP achieves a very high sterling ratio during training, with a mean value of 6.8398, but very poor out-of-sample results during testing with a mean sterling ratio of 0.2723. Fig. 3 shows the shrinkage between training and testing results. SGP has the highest shrinkage evidencing over-fitting of the solutions. The testing performance of SGP solutions is degraded between −94.94% and −104.02% on average as Table 4 shows. We also provide a summary of the results in Table 5.

SGP tends to over-fit solutions. 8 out of the 30 executions (26.66%) where so over-fitted that no results were produced during out-of-sample testing as the rules where never triggered. Only 7 solutions out of the 30 executions (23.33%) had acceptable results when tested out-of-sample.

As an illustration we show in Fig. 4 one of the solutions generated by SGP.

We also analyze the returns of the simulated portfolios in Fig. 8. We can clearly see that SGP delivers the worst performance in terms of inferior mean daily returns. There is a cluster of SGP portfolios whose standard deviation of returns is lower than the rest. This is due to a high percentage of solutions that traded rarely and had a very low market exposition during out-of-sample testing. This can also be seen by looking at Fig. 4 as we can see a very low volatility of the fund value during testing.

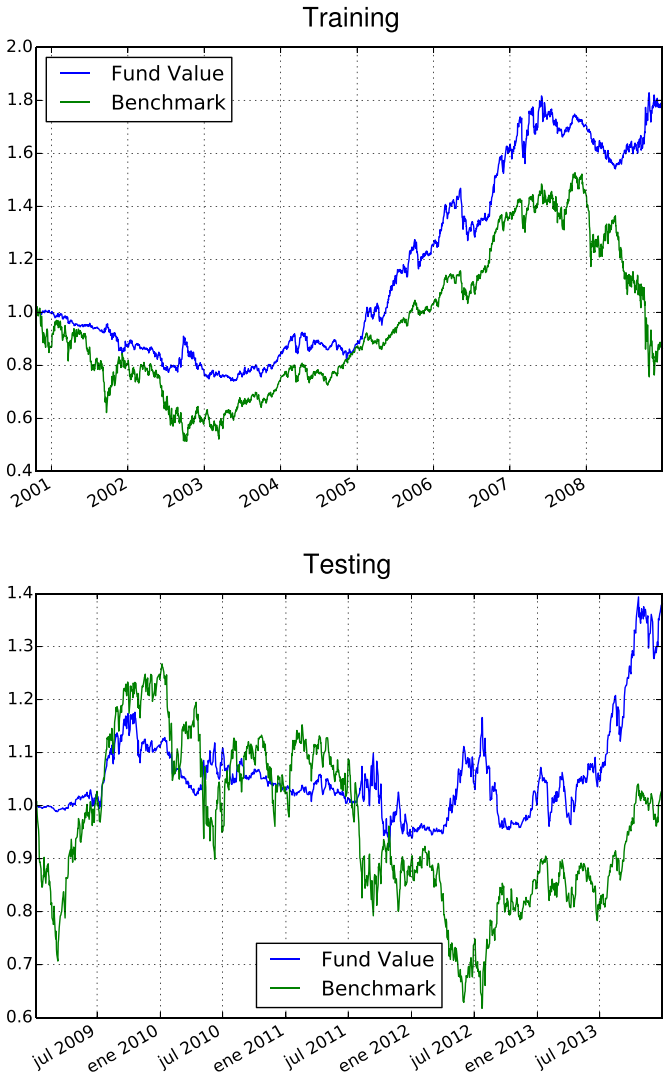

Fig. 5. Training (top) and out-of-sample testing (bottom) performance of a RSFGP strategy.

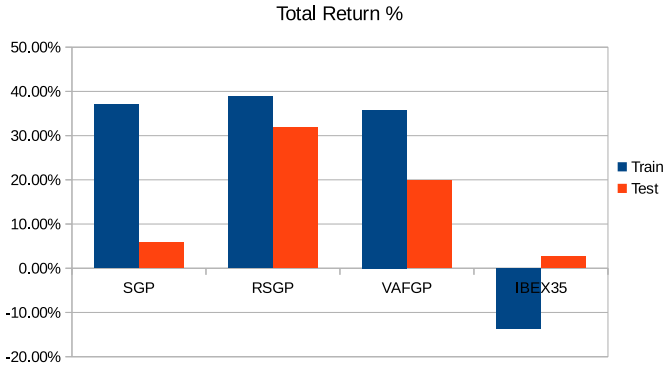

Fig. 6. Mean total return obtained in all executions.

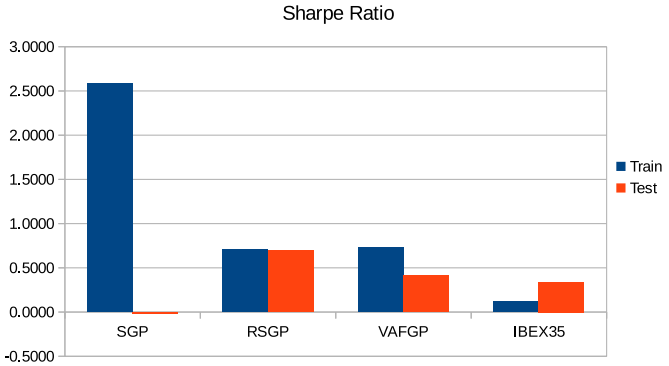

Fig. 7. Mean sharpe ratio obtained in all executions.

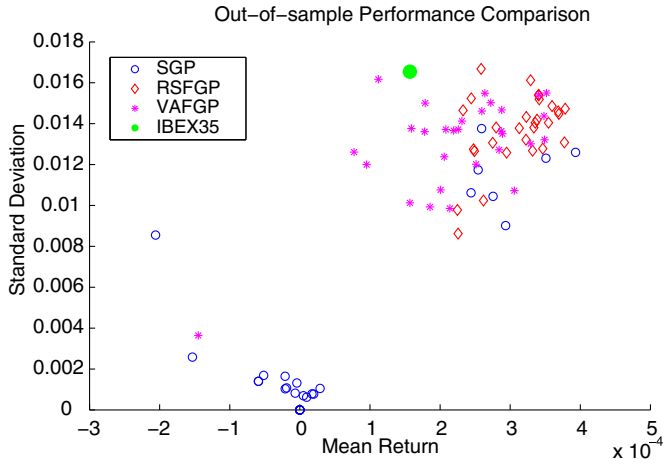

Fig. 8. Comparison of out-of-sample mean daily portfolio returns and standard deviation (volatility).

4.4.2. *Randomly sampled fitness*

Our proposed method RSFGP achieves the highest out-of-sample performance in all the metrics as seen in Figs. 1, 2, 6 and 7 with a mean sterling ratio of 0.6124, a mean total return of 31.81% and a mean Sharpe ratio of 0.6933, substantially beating the IBEX35 benchmark portfolio.

RSFGP also delivers the least shrinkage as can be seen in Fig. 3 and Table 4 with a −28.84% shrinkage in sterling ratio and −1.29% in total return. We have to note that the performance measured in Sharpe ratio did not only experience shrinkage, but quite the opposite, with an average increase of 13.03% during out-of-sample testing as compared to the training dataset.

Fig. 5 shows a robust strategy evolved with RFSGP. The out-of-sample results show higher returns and substantially lower volatility in out-of-sample testing when compared with the IBEX35 benchmark portfolio.

Fig. 8 compares the mean daily returns and standard deviation of returns (volatility) between all the experiments. We can clearly see that a significant part of the solutions generated by RSFGP have a higher return and similar volatility to VAFGP. RSFGP produces portfolios that generate higher returns but at similar risk as VAFGP.

4.4.3. *Volatility adjusted fitness (VAFGP)*

The VAFGP draws its inspiration from the Sharpe ratio. The Sharpe ratio as explained earlier in Section 3.3.1 might not be the best fitness measure.

VAFGP offered the second best out-of-sample testing results after RSFGP with a mean sterling ratio of 0.3809, only slightly better than SGP which had a mean sterling ratio of 0.2723 while the IBEX35 benchmark portfolio had a 0.0521 sterling ratio. In terms of mean total return, VAFGP had a 19.85% return compared with 2.67% for the IBEX35.

The mean daily return and volatility scatter plot in Fig. 8 shows that VAFGP had a worse mean daily return compared to RSFGP at a similar level of volatility. One of the VAFGP solutions had a terrible out-of-sample performance as can be seen in the bottom left

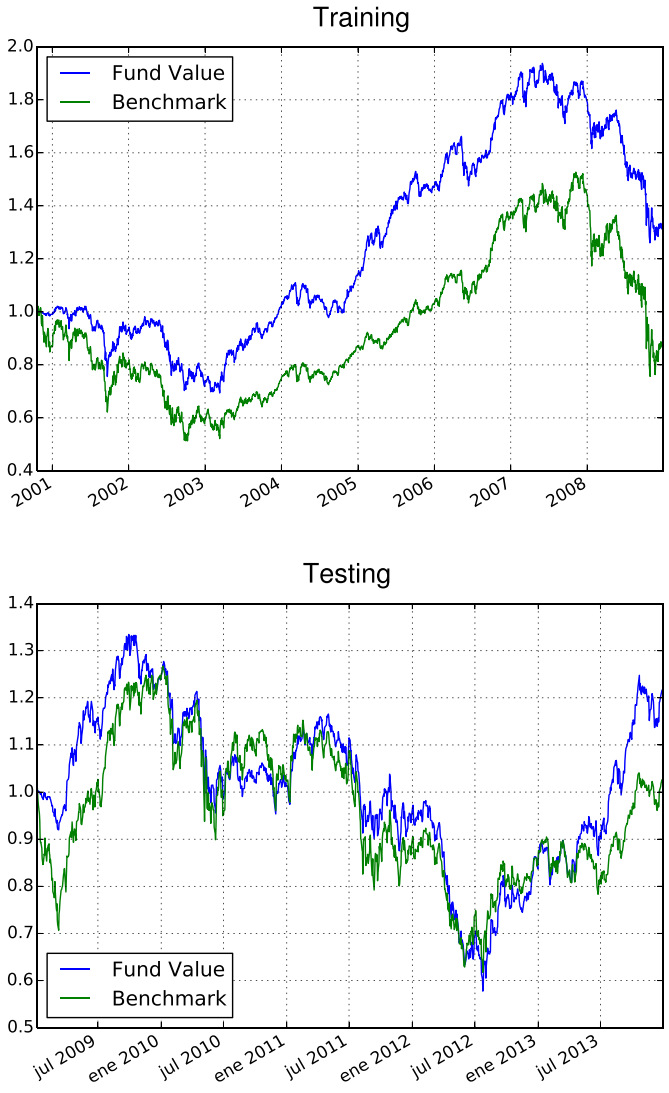

Fig. 9. Training (top) and out-of-sample testing (bottom) performance of a VAFGP strategy.

quadrant of the graph. VAFGP solutions have a higher dispersion in performance, whereas RSFGP solutions are more concentrated in the same area of the plot.

Our results show that including the standard deviation in the fitness (see Eqs. 6 and 7) offers reduced performance and higher shrinkage compared to RSFGP. This might be due to the same problem the Sharpe ratio experiences, as introducing a divisor in the fitness distorts it. Fig. 9 shows the training and testing performance of a VAFGP portfolio.

4.5. Statistical significance of results

We analyzed the statistical significance of the out-of-sample results by performing a Wilcoxon rank-sum test on the sterling ratios obtained from evaluating the solutions out-of-sample. The Wilcoxon rank-sum test is a non-parametric test of the null hypothesis that two populations are the same (have the same median) against an alternative hypothesis that one population has larger values than the other. This test is very practical as it does not assume a normal-distribution and has greater efficiency than the *t-test* on non-normal distributions.

Table 6 shows the *P-values* obtained in the Wilcoxon rank-sum test. All tests have a low *P*-value thus rejecting the null hypothesis that results between SGP, RSFGP and VAFGP are the same at the 5% significance level. The best results are given by RSFGP, followed by VAFGP and lastly SGP as can be seen in Fig. 1.

Table 6
Wilcoxon rank-sum test *P-values* for out-of-sample-testing.

	SGP	RSFGP	VAFGP
SGP	1.00000	0.00490	0.09730
RSFGP	0.00490	1.00000	0.00047
VAFGP	0.09730	0.00047	1.00000

5. Conclusions and future works

This paper presents a novel GP method for learning robust trading strategies using a random sampling method (RSFGP) which improves the performance of solutions when tested out-of-sample and reduces over-fitting. RSFGP calculates the fitness across the randomly sampled segments from the time series and produces solutions that perform in a similar fashion during testing and training, and that can be used for a prolonged duration across different market conditions. In this study we have also included along some popular technical indicators common in the literature, some novel financial metrics never used together previously on other GP studies such as the returns, the moving average of returns, the Capital Asset Pricing Method (CAPM) alpha and beta, the Sharpe ratio and the volatility of the stocks calculated over different period lengths.

The proposed approach is able to cope well with extreme drops in the market, reducing the possible loses of capital, and was validated using real and publicly available market data on 21 of the most liquid stocks from the Spanish stock market. The results show a return of slightly higher than 30% for the testing period 2009–2013, having this period experienced the sovereign debt crisis that affected Spain, which brought the market down to level to the worst days of the 2008 sub-prime crisis with the collapse of Lehman Brothers. In the same period our benchmark, the Spanish IBEX35 market index gained less than 3%.

We have also studied the effects of including the standard deviation as a divisor in the fitness as proposed in Yan and Clack (2010) with their *Volatility Adjusted Fitness* (VAFGP). The results of this study show worse performance for VAFGP compared to RSFGP. We demonstrate that for our dataset, including the standard deviation of the fitness as a divisor degrades the performance of the GP evolved solutions.

Both VAFGP and RSFGP were able to produce solutions that on average beat the IBEX35 benchmark portfolio during out-of-sample testing in terms of risk and return. The RSFGP is the method that offered the best results with an mean return of 31.81% compared to 2.67% for the IBEX35 reference benchmark. RSFGP is also the method that experienced the least shrinkage from training to out-of-sample testing, clearly demonstrating that our method increases the robustness of solutions and reduces over-fitting.

Some key advantages of our method are that it learns a trading rule which can be applied to managing a portfolio of stocks in an automatic manner without requiring the help of financial market experts. These solutions are robust, as they can be used for prolonged periods without needing the system to be retrained, are able to cope with extreme market environments, and provide similar performance during training as when tested out-of-sample. Our random sampling method had the least shrinkage when compared to SGP and VAFGP.

Some limitations of this research were the consideration of only long positions for the portfolio, as sometimes market regulators place bans on short selling, as was the case for bank stocks during the European sovereign debt crisis, and bank bailouts by the government after the sub-prime crisis. Another research limitation of this work

was the number stocks considered, this was due to choosing only the most liquid stocks that had trading data for the years 2000–2013, resulting in the 21 stocks of Table 1.

As future works, we plan to extend this work to other markets and datasets in order to continue studying the beneficial effects of our proposed random sampling method. We also plan to broaden the scope of the problem where a long/short portfolio is simulated and the buy and sell rules are co-evolved. We would also like to include in future research different financial metrics such as autocorrelation of returns, and test different risk metrics as VaR (Value at Risk) and CVaR (Conditional Value at Risk) instead of maximum draw-down.

Acknowlegments

We would like to thank the anonymous reviewers for their insightful comments which have greatly improved the quality of this publication.

References

Adamu, K., & Phelps, S. (2010). Coevolution of technical trading rules for high frequency trading. *Proceedings of the World Congress on Engineering, I*, 96–101.

Aguilar-Rivera, R., Valenzuela-Rendón, M., & Rodríguez-Ortiz, J. (2015). Genetic algorithms and darwinian approaches in financial applications: a survey. *Expert Systems with Applications, 42*(21), 7684–7697.

Allen, F., & Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules. *Journal of financial Economics, 51*, 245–271.

Becker, L., & Seshadri, M. (2003). GP-evolved technical trading rules can outperform buy and hold. *Proceedings of the Sixth International Conference on Computational Intelligence and Natural Computing, Embassy Suites Hotel and Conference Center, Cary, North Carolina USA, September 26–30 2003*.

Berutich, J. M., Luna, F., & López, F. (2014). On the quest for robust technical trading strategies using multi-objective optimization. *AI Communications, 27*(4), 453–471.

Bowers, C. (2006). *Simulating evolution with a computational model of embryogeny* Doctral Thesis. The University of Birmingham. November.

Branke, J. (1998). Creating robust solutions by means of evolutionary algorithms. In *Proceedings of the 5th international conference on parallel problem solving from nature PPSN V* (pp. 119–128). London, UK, UK: Springer-Verlag.

Dempster, M., & Jones, C. (2001). A real-time adaptive trading system using genetic programming. *Quantitative Finance, 1*(4), 397–413.

Fortin, F. A., De Rainville, F. M., Gardner, M. A., Parizeau, M., & Gagné, C. (2012). DEAP: evolutionary algorithms made easy. *Journal of Machine Learning Research, 13*, 2171–2175.

Granger, C. W., & Hyung, N. (2004). Occasional structural breaks and long memory with an application to the S&P 500 absolute stock returns. *Journal of Empirical Finance, 11*(3), 399–421.

Gypteau, J., Otero, F., & Kampouridis, M. (2015). Generating directional change based trading strategies with genetic programming. In A. M. Mora, & G. Squillero (Eds.), *Applications of evolutionary computation* (pp. 267–278). Springer International Publishing. (vol. 9028). Lecture Notes in Computer Science.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI, USA: University of Michigan Press.

Hsu, C.-M. (2011). A hybrid procedure for stock price prediction by integrating self-organizing map and genetic programming. *Expert Systems with Applications, 38*(11), 14026–14036.

Hu, Y., Liu, K., Zhang, X., Su, L., Ngai, E., & Liu, M. (2015). Application of evolutionary computation for rule discovery in stock algorithmic trading. *Applied Soft Computing, 36*(C), 534–551.

Iba, H., & Aranha, C. (2012). *Practical applications of evolutionary computation to financial engineering*: 11. Berlin Heidelberg: Springer.

Kitano, H. (2004). Biological robustness.. *Nature reviews. Genetics, 5*(vol. 11), 826–837.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*. Cambridge, MA, USA: MIT Press.

Kushchu, I. (2002). Genetic programming and evolutionary generalization. *IEEE Transactions on Evolutionary Computation, 6*(5), 431–442.

Lohpetch, D., & Corne, D. (2009). Discovering effective technical trading rules with genetic programming: towards robustly outperforming buy-and-hold. *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 439–444.

Luengo, S., Winkler, S., Barrero, D., & Castao, B. (2015). Optimization of trading rules for the spanish stock market by genetic programming. In M. Ali, Y. S. Kwon, C.-H. Lee, J. Kim, & Y. Kim (Eds.), *Current approaches in applied artificial intelligence* (pp. 623–634). Springer International Publishing. (vol. 9101). Lecture Notes in Computer Science.

Mallick, D., & Lee, V. C. (2008). An empirical study of genetic programming generated trading rules in computerized stock trading service system. *2008 International Conference on Service Systems and Service Management*, 1–6.

Mehta, K., & Bhattacharyya, S. (2004). Adequacy of training data for evolutionary mining of trading rules. *Decision Support Systems, 37*(4), 461–474.

Mousavi, S., Esfahanipour, A., & Zarandi, M. H. F. (2014). A novel approach to dynamic portfolio trading system using multitree genetic programming. *Knowledge-Based Systems, 66*, 68–81.

Neely, C. J. (2003). Risk-adjusted, ex ante, optimal technical trading rules in equity markets. *International Review of Economics & Finance, 12*(1), 69–87.

Pinto, J. M., Neves, R. F., & Horta, N. (2015). Boosting trading strategies performance using VIX indicator together with a dual-objective evolutionary computation optimizer. *Expert Systems with Applications, 42*(19), 6699–6716.

Poli, R., Systems, E., Langdon, W. B., Sciences, M., Mcphee, N. F., & Koza, J. R. (March 2008). *A field guide to genetic programming*. Lulu Enterprises, UK Ltd.

Soule, T. (2003). Operator choice and the evolution of robust solutions. *Genetic Programming Theory and Practice*, 257–269.

Yan, W., & Clack, C. D. (2010). Evolving robust GP solutions for hedge fund stock selection in emerging markets. *Soft Computing, 15*(1), 37–50.

Zhang, H., & Ren, R. (2010). High frequency foreign exchange trading strategies based on genetic algorithms. *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, 426–429.