

Gene Expression Programming and Trading Strategies

Georgios Sermpinis, Anastasia Fountouli, Konstantinos Theofilatos, Andreas Karathanasopoulos

► **To cite this version:**

Georgios Sermpinis, Anastasia Fountouli, Konstantinos Theofilatos, Andreas Karathanasopoulos. Gene Expression Programming and Trading Strategies. Harris Papadopoulos; Andreas S. Andreou; Lazaros Iliadis; Ilias Maglogiannis. 9th Artificial Intelligence Applications and Innovations (AIAI), Sep 2013, Paphos, Greece. Springer, IFIP Advances in Information and Communication Technology, AICT-412, pp.497-505, 2013, Artificial Intelligence Applications and Innovations. <10.1007/978-3-642-41142-7_50>. <hal-01459683>

HAL Id: hal-01459683

<https://hal.inria.fr/hal-01459683>

Submitted on 7 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Gene Expression Programming and Trading Strategies

Georgios Sermpinis¹, Anastasia Fountouli¹, Konstantinos Theofilatos² Andreas Karathanasopoulos³

¹University of Glasgow Business School, University of Glasgow, Adam Smith Building, Glasgow, G12 8QQ, UK (georgios.sermpinis@glasgow.ac.uk, natfou86@hotmail.com)

²Pattern Recognition Laboratory, Dept. of Computer Engineering & Informatics, University of Patras, 26500, Patras, Greece (theofilk@ceid.upatras.gr)

³Business School, University of East London, United Kingdom (a.karathanasopoulos@uel.ac.uk)

Abstract. This paper applies a Gene Expression Programming (GEP) algorithm to the task of forecasting and trading the SPDR Down Jones Industrial Average (DIA), the SPDR S&P 500 (SPY) and the Powershares Qqq Trust Series 1 (QQQ) exchange traded funds (ETFs). The performance of the algorithm is benchmarked with a simple random walk model (RW), a Moving Average Convergence Divergence (MACD) model, a Genetic Programming (GP) algorithm, a Multi-Layer Perceptron (MLP), a Recurrent Neural Network (RNN) and a Gaussian Mixture Neural Network (GM). The forecasting performance of the models is evaluated in terms of statistical and trading efficiency. Three trading strategies are introduced to further improve the trading performance of the GEP algorithm. This paper finds that the GEP model outperforms all other models under consideration. The trading performance of GEP is further enhanced when the trading strategies are applied.

Keywords: Genetic Programming, Gene Expression Programming, Daily Trading, DJIA, S&P500

1 Introduction

Evolutionary and Genetic Programming are becoming popular forecasting tools with an increasing number of Finance applications ([1] and [2]). This paper applies a promising evolutionary algorithm, the GEP, to the task of forecasting and trading the DIA, SPY and QQQ ETFs. Its performance is benchmarked with a simple random walk model (RW), a Moving Average Convergence Divergence (MACD) model, a Genetic Programming (GP) algorithm a Multi-Layer Perceptron (MLP), a Recurrent Neural Network (RNN) and a Higher Order Neural Network (HONN). All models are evaluated in terms of statistical accuracy and trading profitability.

The models under study will forecast the one day ahead return of the DIA, SPY and QQQ ETF. Based on the sign of the forecasted return, a trading signal will be generated. In order to further improve the trading performance of the GEP algorithm, three trading strategies based on one-day ahead volatilities forecasts of the series under study and the size of the forecasts will be introduced.

The five non linear forecasting models will attempt to capture the non linear, non stationary and complex behaviour that dominate financial trading series. The RW and MACD models will act as linear benchmarks. Concerning the trading strategies, the economic rationale is twofold. To improve the trading performance of the best performing model (the GEP) and to reduce the risk from the trading signals generated. Complicated trading strategies can be a profitable tool to investors as they consider additional factors that might improve the overall profitability of their portfolio.

The GEP algorithm is a domain-independent technique that runs in various environments. These environments are structured in a manner which approximates problems in order to produce accurate forecasts. GEP is based on the Darwinian principle of reproduction and survival of the fittest. It applies biological genetic operations such as crossover recombination and mutation to identify complex non-linear and non-stationary patterns. [3] and [4] underline that Evolutionary Algorithms address and quantify complex issues as an automated process via programming, which enable computers to process and solve problems.

In financial forecasting although there are several applications of NNs, the empirical evidence of GEP is quite limited with the notable exceptions of [5] and [6]. This can be explained by the complexity of the algorithm compared to NNs (see [7]). Nevertheless, GEP has been applied successfully in other fields of science, such as mining and computing ([8] and [9]). A primitive form of the GEP algorithm, the Genetic Programming (GP) has several financial applications. [10] applied successfully GP in predicting the daily highest and lowest stock prices of six US stocks while [11] applied successfully the same algorithm to the task of forecasting two exchange rates.

NNs are popular forecasting tools in Finance with numerous applications. [12] used NNs to combine volatility forecasts of the US, Canadian, Japanese and UK stock markets and [13] use NNs to forecast and trade successfully the general index of the Madrid Stock Market.

Although there are limited empirical evidence around the forecasting superiority of GEP compared to NNs and GP, their theoretical advantages are great. Genetic Programming (GP) classifies its individuals as non-linear comprising of different shapes and sizes (tree like structures). On the other hand,

GEP it also classifies individuals as symbolic strings of fixed size (i.e. chromosomes). GEP clearly distinguishes the differences between the genotype and the phenotype of individuals within a population. In [7] the authors argue that GEP represents not only an individual's genotype, in the form of chromosomes, but also its phenotype as a tree like structure of expressions in order to establish fitness. Compared with NNs, GEP has no risk of getting trapped in local optima and is able to reach the optimal solution faster. The findings of this paper support these arguments as the results show that the GEP algorithm outperforms all other models in terms of statistical accuracy and trading efficiency.

The rest of the paper is organized as follows. Section 2 describes the dataset while section 3 describes the GEP algorithm and the benchmark models while Section 4 displays the empirical results. Section 5 provides some concluding remarks.

2 Dataset

In this study, seven forecasting models are employed to the task of forecasting and trading the one day ahead logarithmic return of the DIA, SPY and QQQ ETFs. ETFs are investment funds that are designed to replicate stock market indices. The DIA, the SPY and the QQQ ETF are designed to mimic the Down Jones Industrial Average, the S&P 500 and NASDAQ 100 stock market indices respectively. ETFs offer investors the opportunity to trade stock market indices with extremely low transaction costs.

The period from 03/09/2002 to 31/12/2008 will act as initial in-sample and the period from 02/01/2009 to 31/12/2012 as out-of-sample period. The parameters of the forecasting models will be optimized during the in-sample period and their performance will be validated to the unknown out-of-sample dataset. This estimation will be rolled forward each year. For example in the start, the models will be trained from 03/09/2002 to 31/12/2008 and validated from 02/01/2009 to 31/12/2009. Then the in-sample period would be rolled forward one year (02/01/2003 to 31/12/2010) and the forecasting models would be validated from 03/01/2011 to 30/12/2011. This rolling forward estimation is conducted three times.

3 Forecasting Models

In this section, the forecasting models under study are discussed.

3.1 The Gene Expression Programming

GEP models are symbolic strings of fixed length that represent the chromosome/genotype (genome) of an organism. They are encoded as non linear entities of different sizes and shapes determining an organism's fitness. GEP chromosomes are consisted by multiple genes with equal lengths across the structure of the chromosome. Each gene is comprised of a head (detailing symbols specific to functions and terminals) and a tail (only includes terminals). These can be represented mathematically by equation [1]:

$$t = (n-1)h + 1 \quad [1]$$

where:

h = the head length of the gene.

t = the tail length of the gene.

n = total number of arguments within the function (maximum arity)

The set of terminals included within both the heads and tails of the chromosomes is consisted by constants and specific variables. Each gene is equal and fixed in size and they hold the capacity to code for multiple and varied expression trees (ET). The structure of GEP is able to cope in circumstances when the first element of a gene is terminal producing a single node as well as when multiple nodes ('sub-trees' reproduced by functions) are produced in search for eventual terminality. In GEP valid ETs are always generated while in GP this is not guaranteed. Each gene encodes an ET and in situations where multiple generations arise, GEP codes for sub ETs with interlinking functions to enable reproduction of generations. The parameters of our GEP algorithm are based on the guidelines of [5] and [7]. The different steps of the algorithm are explained below:

In the beginning GEP randomly generates an initial population from populations of individuals and all succeeding populations are spawned from this initial population. The size of the initial population in this application is set to 1000. In the spawning of new generations genetic operators evolve each of the individuals by 'mating' them with other individuals in the population. These genetic operators are deciphered by the nature of the problem. In the next step we develop expression trees from our chromosomes. The structure of each ET is in such a way that the root or the first node corresponds with beginning of each gene. The resulting offspring evolved from the first node is dependent on the number of arguments. In this process of evolution the functions may have numerous arguments however the terminals have a parity of zero. Each of the resulting offspring's characteristics is populated in nodes ordered from left to right. This process is concluded once terminal nodes are established. Later the initial population is generated and the resulting ETs are developed. This is explained in detail by [7]. In order to create

an accurate model suited to our forecasting requirements it is imperative that a function which minimizes error and improves accuracy is used. In our application, the fitness value is defined as the MSE with the lowest MSE being targeted as the best. The main principal during the process of evolution is the generation of offspring from two superior individuals to achieve 'elitism'. As a consequence the best individuals from the parent generation produce offsprings with the most desirable traits whilst the individuals with less desirable traits are removed. On this basis our model minimizes error and maintains superior forecasting abilities. As explained in greater detail by [7], elitism is the cloning of the best chromosome(s)/individual(s) to the next population (also called generation). The role of 'elitism' (via suited genetic operators) enables the selection of fitter individuals without eliminating the entire population. The selection of individuals based on their 'fitness' is carried out during the 'tournament' selection for reproduction and modification. This process selects the individuals at random with the superior ones being chosen for genetic modification in order to create new generations. The size of each tournament is set to 20. In the reproduction of future generations, we apply the mutation and recombination genetic operators. Then the tournament losers are replaced with the new individuals created by the reproduction in the population. A check is made to determine whether the termination criterion is fulfilled, if it is not we return to the second step. As a termination criterion we used a maximum number of 100,000 generations during which the GEP was left to run. As a result the best individual found during the evolution process is presented.

A RW and a MACD model will act as linear benchmarks to the GEP algorithm while three Neural Network (NN) models and a GP algorithm will act as non linear benchmarks.

3.2 Non linear Benchmarks

Three Neural Network (NN) models and a GP algorithm will act as non linear benchmarks. All four models are well documented in the literature.

NNs are usually consisted by three or more layers. The first layer is called the input layer (the number of its nodes corresponds to the number of independent variables). In this study the inputs were selected among the first 12 autoregressive lags of the forecasting series. The specific choice of each set of inputs was based on a sensitivity analysis in the in-sample period. The last layer is called the output layer (the number of its nodes corresponds to the number of response variables). An intermediary layer of nodes, the hidden layer, separates the input from the output layer. Its number of nodes defines

the amount of complexity the model is capable of fitting. Normally, each node of one layer has connections to all the other nodes of the next layer. The training of the network (which is the adjustment of its weights in the way that the network maps the input value of the training data to the corresponding output value) starts with randomly chosen weights and proceeds by applying a learning algorithm called backpropagation of errors ([14]). The iteration length is optimised by minimising the MSE in a subset of in-sample dataset (the last year of the in-sample period each time). The most popular architecture NN model is the MLP. RNNs have the ability to embody short-term memory and past errors while HONNs are able to capture higher order correlations (up to the order three or four) within the dataset.

GP are algorithms that evolve algebraic expressions which enable the analysis / optimization of results in a 'tree like structure'. A complete description of GEP is provided by [4].

4 Empirical Results

4.1 Statistical Performance

In the table below, the statistical performance in the out-of-sample period of all models is presented. For the the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Theil-U statistics, the lower the output the better the forecasting accuracy of the model concerned. The Pesaran-Timmermann (PT) test (1992) examines whether the directional movements of the real and forecast values are in step with one another. It checks how well rises and falls in the forecasted value follow the actual rises and falls of the time series. The null hypothesis is that the model under study has no power on forecasting the ETF return series. The Diebold-Mariano (1995) DM statistic for predictive accuracy statistic tests the null hypothesis of equal predictive accuracy. Both the DM and the PT tests follow the standard normal distribution.

Table 1. Out-of-sample statistical performance

		RW	MACD	GP	MLP	RNN	HONN	GEP
<i>DIA</i>	<i>RMSE</i>	1.027	0.239	0.073	0.136	0.085	0.069	0.057
	<i>MAE</i>	0.833	0.162	0.059	0.097	0.077	0.054	0.046
	<i>Theil-U</i>	0.989	0.756	0.634	0.673	0.641	0.620	0.601

	<i>PT</i>	0.02	0.35	5.89	5.02	5.68	6.44	6.99
	<i>DM</i>	-14.85	-10.47	-4.62	-5.91	-5.73	-4.45	-
<i>SPY</i>	<i>RMSE</i>	1.021	0.287	0.068	0.143	0.072	0.064	0.053
	<i>MAE</i>	0.831	0.195	0.055	0.103	0.061	0.052	0.042
	<i>Theil-U</i>	0.977	0.792	0.621	0.686	0.672	0.613	0.597
	<i>PT</i>	0.05	0.30	6.48	4.85	5.79	6.52	7.11
	<i>DM</i>	-14.56	-11.41	-5.59	-6.65	-5.97	-5.08	-
<i>QQQ</i>	<i>RMSE</i>	1.022	0.295	0.069	0.127	0.071	0.067	0.055
	<i>MAE</i>	0.834	0.204	0.058	0.088	0.059	0.055	0.044
	<i>Theil-U</i>	0.987	0.799	0.632	0.658	0.644	0.616	0.599
	<i>PT</i>	0.08	0.25	6.14	5.19	5.73	6.50	7.08
	<i>DM</i>	-15.31	-12.04	-5.42	-5.81	5.70	-4.77	-

From the table above we note that GEP outperforms all benchmarks for all the statistical measures retained. The HONN model presents the second best performance while the GP algorithm produces the third more statistically accurate forecasts. The PT statistics indicate that all non-linear models under study are able to forecast accurately the directional movements of the three ETF return series while the DM statistics confirm the statistical superiority of the GEP forecasts. On the other hand, the two statistical benchmarks seem unable to provide statistically accurate forecasts for the series and period under study.

4.2 Trading Performance

The trading performance of our models in the out-of-sample period is presented in the table below.

Table 2. Out-of-sample trading performance

		MACD	GP	MLP	RNN	HONN	GEP
<i>DJIA</i>	<i>Information Ratio</i>	-0.17	0.76	0.63	0.67	0.79	0.94
	<i>Annualised Return</i>	2.16%	15.22	10.01	12.94	15.92	23.55

	<i>(including costs)</i>	-	-	-	-	-	-
	Maximum Draw-down	23.24	22.04	20.91	19.95	21.42	17.87
		%	%	%	%	%	%
S&P 500	Information Ratio	-0.24	0.85	0.51	0.74	0.92	1.28
	Annualised Return (including costs)	-	19.57	7.65%	16.79	19.67	26.68
		2.12%	%		%	%	%
	Maximum Draw-down	-	-	-	-	-	-
		36.16	30.12	35.92	34.16	32.54	26.56
		%	%	%	%	%	%
NASDAQ 100	Information Ratio	-0.38	0.78	0.70	0.76	0.86	1.21
	Annualised Return (including costs)	-	17.57	14.51	16.42	19.54	24.02
		3.65%	%	%	%	%	%
	Maximum Draw-down	-	-	-	-	-	-
		32.85	28.19	30.81	31.91	27.91	-24.
		%	%	%	%	%	41%

From the table above we note that GEP clearly outperforms its benchmarks are all the trading criteria retained. In the next section, two trading strategies are introduced to further increase the trading performance of GEP.

4.2.1 Time Varying Volatility Leverage

In order to further improve the trading performance of our models we introduce a leverage based on one day ahead volatility forecasts. The intuition of the strategy is to exploit the changes in the volatility. As a first step we forecast the one day ahead exchange rate volatility with a GARCH, GJR, RiskMetrics and EGARCH model in the test and validation sub-periods. Then, we split these two periods into six sub-periods, ranging from periods with extremely low volatility to periods experiencing extremely high volatility. Periods with different volatility levels are classified in the following way: first the average (μ) difference between the actual volatility in day t and the forecast for day $t+1$ and its 'volatility' (measured in terms of standard deviation (σ)) are calculated; those periods where the difference is between μ plus one σ are classified as 'Lower High Vol. Periods'. Similarly, 'Medium High Vol.' (between $\mu + \sigma$ and $\mu + 2\sigma$) and 'Extremely High Vol.' (above $\mu + 2\sigma$) periods can be defined. Periods with low volatility are also defined following the same 1σ and 2σ approaches, but with a minus sign. For each sub-period leverage is assigned starting with 0 for periods of extremely high volatility to a leverage of 2.5 for periods of extremely low volatility.

4.2.2 The strongest signal

The second trading strategy is based on the absolute values of our forecasts. The GEP algorithm is forecasting the one-day ahead return of the three indices. In this trading strategy, each day we invest to the index that our GEP algorithm is giving the highest in absolute value return or else the strongest signal. Instead of investing in the three indices based on our GEP forecasts, we invest only to the one that the algorithm indicates that will be the most profitable. These trades are held for one day unless in the following day, our GEP signals indicate the same index with the same sign.

4.3 Empirical Results

In the table below, we present the performance of our trading strategies.

Table 1. Trading Strategies

	GEP-DJIA	GEP-S&P500	GEP-NASDAQ 100	GEP- Level of Confidence
<i>Information Ratio</i>	1.03	1.32	1.29	1.96
<i>Annualised Return (including costs)</i>	26.85%	28.14%	26.98%	34.54%
<i>Maximum Draw-down</i>	-18.95%	-28.95%	-22.95%	-25.81%

We note that all trading strategies were successful.

4. Conclusions

In this study, a GEP algorithm was applied to the task of forecasting and trading the DJIA, S&P 500 and NASDAQ 100 indices. It was benchmarked against several non linear models. The GEP forecasts outperformed its benchmarks in terms of annualised return and information ratio. This trading performance was further enhanced with the introduction of two trading strategies.

These results should go a step towards convincing a growing number of quantitative fund managers and academics to experiment beyond the bounds of traditional statistical and neural network models. They also validate the importance of trading strategies in improving the profitability of trading signals.

5 References

1. Chen, S. (2002), *Genetic Algorithms and Genetic Programming in Computational Finance*, Kluwer Academic Publishers, Amsterdam.
2. Lisboa, P., Edisbury, B. and Vellido, A. (2000), 'Business Applications of Neural Networks', *Business Applications of Neural Networks: The State-of-the-Art of Real-World Applications*, World Scientific, Singapore
3. Koza, J.R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge MIT Press
4. Koza J.R. (1998), 'Genetic Programming', In Williams, J. G. and Kent, A., (eds.), *Encyclopedia of Computer Science and Technology*. New York, NY: Marcel-Dekker. 39, (Supplement 24), 29–43.
5. Sermpinis G., Laws, J., Karathanasopoulos, A, and Dunis, C., (2012) 'Forecasting and Trading the EUR/USD Exchange Rate with Gene Expression and Psi Sigma Neural Networks', *Expert Systems with Applications*, 8865-8877.
6. Divsalar et al . (2012). A Robust Data-Mining Approach to Bankruptcy Prediction. *Journal of Forecasting*, 31, 504–523.
7. Ferreira C., (2001), 'Gene Expression Programming: A New Adaptive Algorithm for Solving Problems', *Complex Systems*, 13, 87-129.
8. Lopez, H. S. and Weinert, W. R. (2004), 'An Enhanced Gene Expression Programming Approach for Symbolic Regression Problems', *International Journal of Applied Mathematics in Computer Science*, 14, 375-384.
9. Dehuri, S. and Cho S. B., (2008), 'Multi-Objective Classification Rule Mining Using Gene Expression', *Third International Conference on Convergence and Hybrid Information*.
10. Kaboudan M. A., (2000) 'Genetic Programming Prediction of Stock Prices', *Computational Economics*, 16, 207-236.
11. Alvarez-Díaz, M. and Alvarez, A. (2005), 'Genetic multi-model composite forecast for non-linear prediction of exchange rates', *Empirical Economics*, 30, 3, 643-663.
12. Donaldson, R. G. and Kamstra, M. (1996), 'Forecast Combining with Neural Networks', *Journal of Forecasting*, 15, 49–61
13. Fernandez-Rodríguez, F., González-Martel, C. and Sosvilla-Rivero, S. (2000) 'On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid stock market', *Economics Letters*, 69, 1, 89-94
14. Shapiro, A. F. (2000), 'A Hitchhiker's Guide to the Techniques of Adaptive Nonlinear Models', *Insurance, Mathematics and Economics*, 26, 2, 119-132