Evolving Market Index Trading Rules using Grammatical Evolution

Michael O'Neill[2], Anthony Brabazon[1], Conor Ryan[2], and J.J. Collins[2]

[1] Dept. Of Accountancy, University College Dublin, Ireland.
Anthony.Brabazon@ucd.ie
[2] Dept. Of Computer Science And Information Systems,
University of Limerick, Ireland.
{Michael.ONeill|Conor.Ryan|J.J.Collins}@ul.ie

Abstract. This study examines the potential of an evolutionary automatic programming methodology to uncover a series of useful technical trading rules for the UK FTSE 100 stock index. Index values for the period 26/4/1984 to 4/12/1997 are used to train and test the model. The preliminary findings indicate that the methodology has much potential, outperforming the benchmark strategy adopted.

1 Introduction

The objective of this study is to determine whether an evolutionary automatic programming methodology, Grammatical Evolution, is capable of uncovering useful technical trading rules for the UK FTSE 100 index.

The paper is organised as follows. Section two discusses the background to the technical indicators utilised in this study. Section three describes the evolutionary algorithm adopted, Grammatical Evolution [16] [18]. Section four outlines the data and function sets used. The following sections provide the results of the study followed by a discussion of these results and finally a number of conclusions are derived.

1.1 Technical analysis

A market index is comprised of a weighted average measure of the price of individual shares which make up that market. The value of the index represents an aggregation of the balance of supply and demand for these shares. Some market traders, known as technical analysts, believe that prices move in trends and that price patterns repeat themselves [14]. If we accept this premise, that there are rules, although not necessarily static rules, underlying price behaviour it follows that trading decisions could be enhanced through use of an appropriate rule induction methodology such as Grammatical Evolution (GE). Although controversy exists amongst financial theorists regarding the veracity of the claim of technical analysts, recent evidence has suggested that it may indeed be possible to uncover patterns of predictability in price behaviour. Brock, Lakonishok and LeBaron [3] found that simple technical trading rules had predictive power and suggested that the conclusions of earlier studies that technical trading rules did not have such power were "premature". Other studies which indicated that there may be predictable patterns in share price movements include those which suggest that markets do not always impound new information instantaneously [11] [5], that stock markets can overreact as a result of excessive investor optimism or pessimism [10], that returns on the market are related to the day of the week [7] or the month of the year [9]. The continued existence of large technical analysis departments in international finance houses is consistent with the hypothesis that technical analysis has proven empirically useful.

1.2 Potential for application of evolutionary automatic programming

As noted by Iba and Nikolaev [12] there are a number of reasons to suppose that the use of an evolutionary automatic programming (EAP) approach can prove fruitful in the financial prediction domain. EAP can conduct an efficient exploration of the search space and can uncover dependencies between input variables, leading to the selection of a good subset for inclusion in the final model. Additionally, use of EAP facilitates the utilisation of complex fitness functions including discontinuous, non-differentiable functions. This is of particular importance in the financial domain as the fitness criterion may be complex, usually requiring a balancing of return and risk. EAP, unlike for example basic neural net approaches to financial prediction, does not require the ex-ante determination of optimal model inputs and their related transformations. Another useful feature of EAP is that it produces human-readable rules that have the potential to enhance understanding of the problem domain.

1.3 Motivation for study

This study was motivated by a number of factors. Much of the existing literature concerning the application of genetic algorithms (GA) or GP to the generation of technical trading rules [1] [6] [2] [15] [8] concentrates on the US and to a lesser extent the Japanese stock markets. Published research on this area is both incomplete and scarce. To date, only a limited number of GA / GP methodologies and a limited range of technical indicators have been considered. This study addresses these limitations by examining index data drawn from the UK stock market and by adopting a novel evolutionary automatic programming approach.

2 Background

As with any modelling methodology, issues of data pre-processing need to be considered. Rather than attempting to uncover useful technical trading rules for the FTSE 100 index using raw current and historical price information, this information is initially pre-processed into technical indicators. The objective of these pre-processing techniques is to uncover possible useful trends and other information in the time series of the raw index data whilst simultaneously reducing the noise inherent in the series.

2.1 Technical Indicators

The development of trading rules based on current and historic market price information has a long history [4]. The process entails the selection of one or more technical indicators and the development of a trading system based on these indicators. These indicators are formed from various combinations of current and historic price information. Although there are potentially an infinite number of such indicators, the financial literature suggests that certain indicators are widely used by investors [3][14][17].

Four groupings of indicators are given prominence in prior literature:

i. Moving average indicators
ii. Momentum indicators
iii. Trading range indicators
iv. Oscillators

Given the large search space, an evolutionary automatic programming methodology has promise to determine both a good quality combination of, and relevant parameters for, trading rules drawn from individual technical indicators.

We intend to use of each of these groupings as our model is developed, but in our preliminary investigation, we have limited our attention to moving average indicators.

Moving Average Indicators The simplest moving average systems compare the current share price or index value with a moving average of the share price or index value over a lagged period, to determine how far the current price has moved from an underlying price trend. As they smooth out daily price fluctuations, moving averages can heighten the visibility of an underlying trend. A variation on simple moving average systems is to use a moving average convergence divergence (MACD) oscillator. This is calculated by taking the difference of a short run and a long run moving average. In a recursive fashion, more complex combinations of moving averages of values calculated from a MACD oscillator can themselves be used to generate trading rules. For example, a nine day moving average of a MACD oscillator could be plotted against the raw value of that indicator. A trading signal may be generated when the two plotted moving averages cross. Moving average indicators are trend following devices and work best in trending markets. They can have a slow response to changes in trends in markets, missing the beginning and end of each move. They tend to be unstable in sideways moving markets, generating repeated buy and sell signals (whipsaw) leading to unprofitable trading. Trading systems using moving averages trade-off volatility (risk of loss due to whipsaw) against sensitivity. The objective is to select the lag period which is sensitive enough to generate a useful early trading signal but which is insensitive to random noise.

A description of the evolutionary automatic programming system used to evolve trading rules now follows.

3 Grammatical Evolution

Grammatical Evolution (GE) is an evolutionary algorithm that can evolve computer programs in any language. Rather than representing the programs as parse trees, as in traditional GP [13], a linear genome representation is adopted. A genotype-phenotype mapping process is used to generate the output program for each individual in the population. Each individual, a variable length binary string, contains in its codons (groups of 8 bits) the information to select production rules from a Backus Naur Form (BNF) grammar. The BNF is a plug-in component to the genotype-phenotype mapping process, that represents the output language in the form of production rules. It is comprised of a set of non-terminals that can be mapped to elements of the set of terminals, according to the production rules. An example excerpt from a BNF grammar is given below. These productions state that S can be replaced with either one of the non-terminals `expr`, `if-stmt`, or `loop`.

```
S ::= expr      (0)
    | if-stmt   (1)
    | loop      (2)
```

The grammar is used in a generative process to construct a program by applying production rules, selected by the genome, beginning from the start symbol of the grammar.

In order to select a rule in GE, the next codon value on the genome is generated and placed in the following formula:

$$Rule = Codon\ Value\ MOD\ Num.\ Rules$$

If the next codon integer value was 4, given that we have 3 rules to select from as in the above example, we get $4\ MOD\ 3\ =\ 1$. S will therefore be replaced with the non-terminal `if-stmt`.

Beginning from the left hand side of the genome codon integer values are generated and used to select rules from the BNF grammar, until one of the following situations arise:

i. A complete program is generated. This occurs when all the non-terminals in the expression being mapped, are transformed into elements from the terminal set of the BNF grammar.

ii. The end of the genome is reached, in which case the *wrapping* operator is invoked. This results in the return of the genome reading frame to the left hand side of the genome once again. The reading of codons will then continue unless an upper threshold representing the maximum number of wrapping events has occurred during this individual's mapping process. This threshold is currently set to ten events.
iii. In the event that a threshold on the number of wrapping events is exceeded and the individual is still incompletely mapped, the mapping process is halted, and the individual assigned the lowest possible fitness value.

GE uses a steady state replacement mechanism, such that, two parents produce two children the best of which replaces the worst individual in the current population if the child has a greater fitness. The standard genetic operators of point mutation, and crossover (one point) are adopted. It also employs a duplication operator that duplicates a random number of codons and inserts these into the penultimate codon position on the genome. A full description of GE can be found in [16] [18].

4 Problem Domain & Experimental Approach

We describe an approach to evolving trading rules using GE. This study uses daily data for the UK FTSE 100 stock index drawn from the period 26/4/1984 to 4/12/1997. The training data set was comprised of the first 440 trading days of the data set. The remaining data was divided into five hold out samples totaling 2125 trading days. The division of the hold out period into five segments was undertaken to allow comparison of the out of sample results across different market conditions in order to assess the stability and degradation characteristics of the developed model's predictions. The extensive hold out sample period helps reduce the possibility of training data overfit. The rules evolved by GE are used to generate one of three signals for each day of the training or test periods. The possible signals are *Buy*, *Sell*, or *Do Nothing*. Permitting the model to output a Do Nothing signal reduces the hard threshold problem associated with production of a binary output. This issue has not been considered in a number of prior studies. A variant on the trading methodology developed in Brock et al. [3] is then applied. If a buy signal is indicated, a fixed investment of \$1,000 (arbitrary) is made in the market index. This position is closed at the end of a ten day (arbitrary) period. On the production of a sell signal, an investment of \$1,000 is sold short and again this position is closed out after a ten day period. This gives rise to a maximum potential investment of \$10,000 at any point in time (the potential loss on individual short sales is in theory infinite but in practice is unlikely to exceed \$1,000). The profit (or loss) on each transaction is calculated taking into account a one-way trading cost of 0.2% and allowing a further 0.3% for slippage. The total return generated by the developed trading system is a combination of its trading return and its risk free rate of return generated on uncommitted funds.

The rate adopted in this calculation is simplified to be the average interest rate over the entire data set (8.5%).

The only technical indicator that we adopt for these experiments is the moving average, where the period is determined by evolution. We choose to do this for the sake of simplicity in these preliminary experiments.

As well as the moving average the grammar also allows the use of the binary operators f_and, f_or, and the standard arithmetic operators, and the unary operator f_not. The operations f_and, f_or, and f_not are fuzzy logic operators returning the minimum, maximum, of the arguments, and 1 - the argument, respectively. We are therefore getting a mix of types for free, through the grammar and the genotype-phenotype mapping process of GE.

The signals generated for each day, Buy, Sell, or Do Nothing, are post-processed using fuzzy logic. The trading rule, a fuzzy trading rule, returns values in the range 0 to 1. We use pre-determined membership functions, in this case, to determine what the meaning of this value is. The membership functions adopted were as follows:

$$Buy = 0.0 >= Value < .33$$

$$Sell = .33 >= Value < .66$$

$$DoNothing = .66 >= Value <= 1.0$$

4.1 Data Preprocessing

The value of the FTSE 100 index increased substantially over the training and testing period, rising from 1130.9 to 5082.3. Before the trading rules were constructed, these values were normalised using a two phase preprocessing. Initially the daily values were transformed by dividing them by a 75 day lagged moving average. These transformed values are then normalised using linear scaling into the range 0 to 1. This procedure is a variant on that adopted by Allen and Karjalainen [1]and Iba and Nikolaev [12].

4.2 Selection of Fitness Function

A key decision in applying a GP methodology to construct a technical trading system is to determine what fitness measure should be adopted. A simple fitness measure such as the profitability of the system both in and out of sample is inadequate as it fails to consider the risk associated with the developed trading system. The risk of the system can be estimated in a variety of ways. One possibility is to consider market risk, defined here as the risk of loss of funds due to a market movement. A measure of this risk is provided by the maximum drawdown (maximum cumulative loss) of the system during a training or test period. This measure of risk can be incorporated into the fitness function in a variety of formats including: (return / maximum drawdown) or return - 'x'(maximum drawdown), where 'x' is a pre-determined constant dependent on an investor's psychological risk profile. For a given rate of return, the system generating the lowest maximum drawdown is preferred.

This study incorporates drawdown in the fitness function by subtracting the maximum cumulative loss during the training period from the profit generated during that period. This is a conservative approach which will encourage the evolution of trading systems with good return to risk characteristics. This will provide a more stringent test of trading rule performance as high risk / high reward trading rules will be discriminated against. The adoption of a risk conservative approach will facilitate the comparison of the final results with those of a benchmark buy and hold trading strategy.

5 Results

The results from our preliminary experiments are now given. Runs were conducted with a population size of 500 for 100 generations. Trading rules were evolved with a performance superior to that of a benchmark buy and hold strategy. Under this benchmark, an amount of $10,000 is invested in the market at the beginning of each of the test periods. The gain on this investment to the end of each period is then calculated. The best individual (set of trading rules) found to date made a profit of US$2491 over the training period.

When tested on the 5 out of sample periods following the training data set we find that this individual was consistently profitable, with the exception of a small loss in test period 4. It is noteworthy that the performance of this individual showed no significant evidence of degradation in succeeding out of sample test periods. In some cases the individual performed better out of sample than in the training period. This individual demonstrated robust performance, showing an ability to adapt to a period of crisis in the market in the second test period caused by the market collapse in Oct 1987. Plots of the index over each of the test periods and the training period can be seen in Fig. 1.

To facilitate assessment of these results, they are compared with those of the benchmark buy and hold strategy. The results of this buy and hold strategy can be seen in table 1.

Trading Period (Days)	**Buy & Hold** Profit (US$)	**Best-of-run** Profit(US$)	**Best-of-run** Avg. Daily Investment
Test 1 (440 to 805)	5244	1190	7959
Test 2 (805 to 1170)	-1376	5459	4356
Test 3 (1170 to 1535)	1979	2122	6973
Test 4 (1535 to 1900)	1568	-595	7109
Test 5 (3196 to 3552)	3852	10143	6315
Total	11267	18319	

Table 1. A comparison of benchmarks with the best of run individual.

In assessing these results, the market risk profile of each trading strategy should be considered. The buy and hold strategy maintains an investment of

$10,000 in the market at all times whereas the maximum investment of the developed trading system, ignoring drawdown, is $10,000. Looking at table 1 we can see the average daily investment made by the best of run individual for each test period. Averaged over all 5 test periods the developed system has an investment of $6542 in the market.
There is no clear evidence that the trading system has higher market risk than the buy and hold strategy.

6 Discussion

In evaluating the performance of any market predictive system, a number of caveats must be borne in mind. Any trading model constructed and tested using historic data will tend to perform less well in a live environment than in a test period for a number of reasons. Live markets have attendant problems of delay in executing trades, illiquidity, interrupted / corrupted data and interrupted markets. The impact of these issues is to raise trading costs and consequently to reduce the profitability of trades generated by any system. An allowance for these costs ("slippage") has been included in this study but it is impossible to determine the scale of these costs ex-ante with complete accuracy. In addition to these costs, it must be remembered that the market is competitive. As new computational technologies spread, opportunities to utilise these technologies to earn excess risk-adjusted profits are eroded. As a result of this technological "arms-race", estimates of trading performance based on historical data may not be replicated in live trading as other market participants will apply similar technology. This study ignores impact of dividends. Although a buy-and-hold strategy will generate higher levels of dividend income than an active trading strategy, the precise impact of this factor is not determinable ex-ante. It is notable that the dividend yield on most stock exchanges has fallen sharply in recent years and that the potential impact of this factor has lessened.

7 Conclusions & Future Work

GE was shown to successfully evolve trading rules with a performance superior to the benchmark buy and hold strategy. These preliminary results, with regard to the potential utility of technical analysis, are more positive than those reported in some earlier studies. Allen and Karjalainen [1] found that after transaction costs, the technical trading rules developed in their study, using a more traditional GP methodology, did not produce excess returns. However, the scope of their finding is limited as the methodology adopted in the study did not compare returns with a similar risk profile. The risk of the benchmark buy-and-hold portfolio exceeded that of the portfolio generated by the technical trading rules because an investor following the technical trading system was only invested in the market 57% of the time.

There is notable scope for further research utilising GE in this problem domain. Our preliminary methodology has included a number of simplifications, for example, we only considered moving averages, a primitive technical indicator. The incorporation of additional technical may further improve the performance of our approach.

References

1. Allen, F., Karjalainen, R. (1999) Using genetic algorithms to find technical trading rules. *Journal of Financial Economics*, **51**, pp. 245-271, 1999.
2. Bauer R. (1994). Genetic Algorithms and Investment Strategies, New York: John Wiley & Sons Inc.
3. Brock, W., Lakonishok, J. and LeBaron B. (1992). 'Simple Technical Trading Rules and the Stochastic Properties of Stock Returns', Journal of Finance, 47(5):1731-1764.
4. Brown, S., Goetzmann W. and Kumar A. (1998). 'The Dow Theory: William Peter Hamilton's Track Record Reconsidered', Journal of Finance, 53(4):1311-1333.
5. Chan, L. K. C., Jegadeesh, N. and Lakonishok, J. (1996). 'Momentum strategies', Journal of Finance, Vol. 51, No. 5, pp. 1681 - 1714.
6. Colin, A. (1994). 'Genetic Algorithms for Financial Modelling', in Guido Deboeck (Editor) (1994). Trading on the edge: neural, genetic and fuzzy systems for chaotic and financial markets, New York: John Wiley & Sons Inc.
7. Cross, F. (1973). 'The Behaviour of Stock prices on Friday and Monday', Financial Analysts' Journal, Vol. 29(6), pp.67-74.
8. Deboeck G. (1994). 'Using GAs to optimise a trading system', in Guido Deboeck (Editor) (1994). Trading on the edge: neural, genetic and fuzzy systems for chaotic and financial markets, New York: John Wiley & Sons Inc.
9. DeBondt, W. and Thaler, R. (1987). 'Further Evidence on Investor Overreaction and Stock Market Seasonality', Journal of Finance, Vol. 42(3):pp.557-581.
10. Dissanaike, G. (1997). 'Do stock market investors overreact?', Journal of Business Finance & Accounting (UK), Vol. 24, No.1, pp. 27-50.
11. Hong, H., Lim, T. and Stein, J. (1999). 'Bad News Travels Slowly: Size, Analyst Coverage and the Profitability of Momentum Strategies', Research Paper No. 1490, Graduate School of Business, Stanford University.
12. Iba H. and Nikolaev N. (2000). 'Genetic Programming Polynomial Models of Financial Data Series', In *Proc. of CEC 2000*, pp. 1459-1466, IEEE Press.
13. Koza, J. (1992). *Genetic Programming.* MIT Press.
14. Murphy, John J. (1999). Technical Analysis of the Financial Markets, New York: New York Institute of Finance.
15. Neely, C., Weller P. and Dittmar, R. (1997). 'Is technical analysis in the foreign exchange market profitable? A genetic programming approach", Journal of Financial and Quantitative Analysis, Vol. 32, No. 4, pp. 405 - 428.
16. O'Neill M., Ryan C. (2001) Grammatical Evolution. *IEEE Trans. Evolutionary Computation.* 2001.
17. Pring, M. (1991). Technical analysis explained: the successful investor's guide to spotting investment trends and turning points, New York: Mc Graw-Hill Inc.
18. Ryan C., Collins J.J., O'Neill M. (1998). Grammatical Evolution: Evolving Programs for an Arbitrary Language. *Lecture Notes in Computer Science 1391, Proceedings of the First European Workshop on Genetic Programming*, pages 83-95. Springer-Verlag.

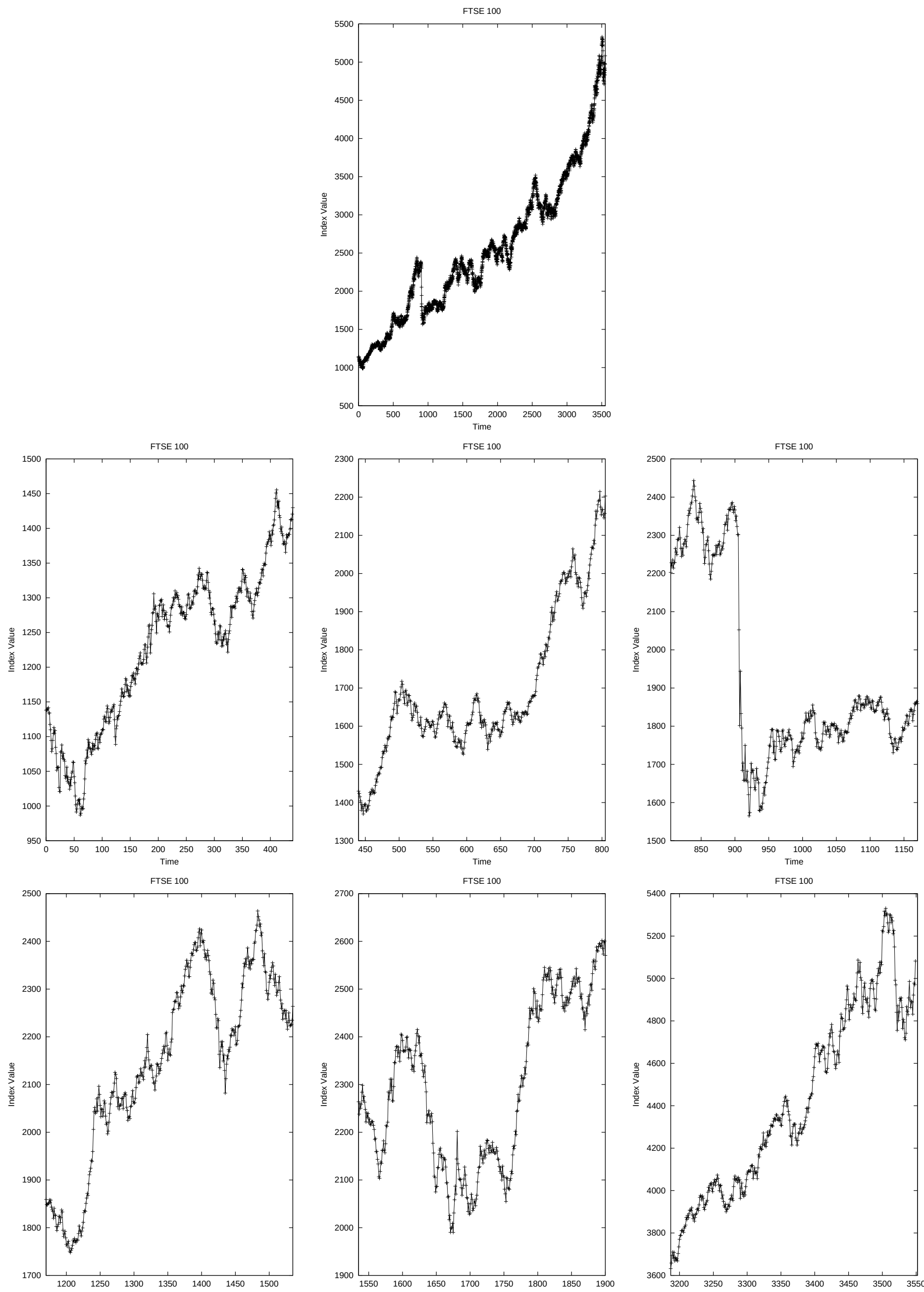

Fig. 1. A plot of the FTSE 100 over the entire data set (top), over the training period (middle-left), over the first two test periods. Days 365 to 730 (middle-center), and days 730 to 1095 (middle-right), and the third, fourth & fifth test periods (bottom row, from left to right).