Evolving Rule-Based Trading Systems

Christian Setzkorn[1], Laura Dipietro[2], and Robin Purshouse[3]

[1] Department of Computer Science, University of Liverpool, UK
[2] Advanced Robotics and Systems Lab, Scuola Superiore Sant'Anna, Italy
[3] Department of Automatic Control and Systems Engineering, University of Sheffield, UK

Abstract. In this study, a market trading rulebase is optimised using genetic programming (GP). The rulebase is comprised of simple relationships between technical indicators, and generates signals to buy, sell short, and remain inactive. The methodology is applied to prediction of the Standard & Poor's composite index (02-Jan-1990 to 18-Oct-2001). Two potential market systems are inferred: a simple system using few rules and nodes, and a more complex system. Results are compared with a benchmark buy-and-hold strategy. Neither trading system was found capable of consistently outperforming this benchmark. More complicated rulebases, in addition to being difficult to understand, are susceptible to overfitting. Simpler rulebases are more robust to changing market conditions, but cannot take advantage of high-profit-making opportunities. By increasing the richness of the available rulebase building-blocks and the variety of training data, it is anticipated that subsequent systems will surpass the benchmark strategy.

1 Introduction

This paper presents a study of market trading system development using evolutionary algorithms (EAs). An explicit aim of the research is to develop rulebases that are simple and easy to analyse, whilst also outperforming the benchmark buy-and-hold strategy. The resulting methodology is applied to rule generation for the Standard & Poor's composite index.

A brief background to stock market prediction is presented in Section 2, together with the motivation for the application of EAs to this severely difficult problem. In Section 3, the proposed technique is outlined. The selection of technical indicators and rulebase functionality are discussed. Section 4 describes GP technical implementation details; particular attention is paid to the choice of fitness function. Section 5 assesses the effectiveness of the new methodology via application to the past decade's closing prices of the Standard & Poor's 500 share index. The most profitable rulebase identified is validated on the test data sets and is subjected to scrutiny. In particular, the regions of the index corresponding to high and low performance are analysed. In Section 6 initial conclusions are offered, together with proposals for future developments.

2 Background

2.1 Prediction of Stocks

The prediction of stock market behaviour is a very difficult task [8]. A market is a time-varying, highly volatile process that largely resembles a random walk. Published investigations have been largely unsuccessful, failing to produce excess returns over a simple *buy-and-hold* strategy. Indeed, a controversial investment theory known as the *Efficient Market Hypothesis* exists which states that it is impossible to beat the market.

Stock market prediction analyses can generally be classified as either *fundamental* or *technical*. The former approach considers the *cause* of market behaviour, whilst the latter studies the *effect*. Thus, technical analysis is based only on quantifiable market data, whilst fundamental analysis includes data related to the market situation, time of year, company prospects and so forth [8]. Technical analysis has attracted a large following amongst trading practitioners but has been criticised in the past by theoreticians (see, for example, [6]). It should be noted, however, that more recent studies in the literature have given some support to the technical approach [3]. The technique developed in this paper has focused solely on technical analysis. However, it could easily be extended to cater for fundamental data types.

Technical stock analysis is based on three basic principles [11], namely:

1. Market action discounts everything;
2. Prices move in trends;
3. History repeats itself.

If these tenets are assumed to be true, then it should be possible to develop rules to predict market behaviour. Technical analysis often involves *technical indicators*, which are indices formed from combinations of current and past price data. Popular indicators include moving averages, break-out systems, and oscillators. Many different variations of each indicator have been developed [1].

Many attempts have been made to predict various financial markets, ranging from traditional time series approaches to artificial intelligence techniques, such as fuzzy systems [9] and, especially, artificial neural network (ANN) methodologies [15][9][8]. However, the main drawback with ANNs, and other largely *black-box* techniques, is the tremendous difficulty in interpreting the results. They do not provide an insight into the nature of the interactions between the technical indicators and the stock market fluctuations. Thus, there is a need to develop methodologies that facilitate an increased understanding of market processes, in addition to providing temporally accurate predictions [7].

2.2 Potential for Evolutionary Computing

EAs have recently been proposed as potential search and optimisation engines of a trading system. Allen and Karjalainen [2] used genetic algorithms (GAs) to derive trading rules for the Standard & Poor's composite index, whilst Neely and Weller [12] applied a GP approach. Recent work by O'Neill et al [13] adopted a grammar-based technique. This current work is somewhat in its infancy. Note in particular that candidate solutions in these studies were each comprised of only a single rule. Further studies are required in order to reap the full benefits of the evolutionary computing framework.

2.3 Background to the Study

In this paper, a technical stock analysis *rulebase* (as opposed to a single rule) is developed using GP to produce *buy long*, *sell short*, and *do nothing* signals at the end of each day's trading. All investments are closed out after a fixed period of 10 days, following the approach outlined in [13]. This differs from other GP schemes in which the trading rule identifies regions to be in or out of the market. A key aspect of the work is that the system is designed such that the rulebase is easily understandable. A further advantage is that the rules can be periodically tuned to increase their relevance in the presence of changing market dynamics.

3 The Trading System

3.1 The GP Search Engine

A minimalist approach has been taken regarding the choice of GP function set and terminal set. A central aim of this work is to obtain rules that are readily understandable and have a logical structure. Thus, the number of rules per candidate solution has been heavily constrained, as have the available operators in the functional set and the range of indicators available in the terminal set. Note that the search space of possible rulebase configurations is still considerable.

The expressions in the trading rules of a candidate solution operate on the outputs of financial indicators rather than the raw index data itself. The evolutionary process chooses which indicators to use, and how they should be combined to form a rule; thus, a set of candidate trading systems consisting of several rules is evolved.

3.2 Selection of Financial Technical Indicators

This paper focuses on one of the most fundamental, versatile, and popular financial indicators: the *moving average* (MA).

The MA technical indicator takes an average over a particular segment of data. The segment is defined with respect to the current day. Hence, the operator acts on a moving window of the time series. A common approach is to take the arithmetic mean of the last ten days' closing prices [11], although many variations exist. The MA is a smoothing device that identifies trends in the data. Note that the MA follows the market: the trend is identified *after* it has begun. The MA works best in trending periods of the market. In sideways moving periods, MAs have been shown to perform poorly. Other financial indicators, such as an oscillator, must be deployed in these latter conditions. However, some types of oscillator can actually be constructed from MA building blocks.

The length of a moving average crucially affects its performance as an indicator [11]. A shorter average will identify trends faster but carries a greater risk of providing a false signal. A trade-off exists between sensitivity to trends and insensitivity to noise. Thus, most traders use a combination of MAs to derive a trading signal. In the work presented here, the GP chooses the length of average to use (within a pre-defined maximum limit of 30 days), and also combines and interprets multiple MAs.

In addition to selecting the length of an MA, the GP is also free to select the region to average within the predefined window of past data. Thus, in this approach, the MA is not anchored to the current closing price. This provides an additional degree of flexibility. The standard anchored approach is represented as a subset of the total search space.

Since each candidate trading system consists of multiple rules, there is scope for the GP to identify and support good combinations of MAs for various market conditions within a single system. If the system is described by a single rule only, this rule may become very complicated if it is to simultaneously support substantially different market conditions.

MAs exist that do not rely on the arithmetic mean as the method of averaging. Linearly weighted and exponentially smoothed MAs have been proposed and investigated in the literature [11]. These variations are beyond the scope of the current pilot study.

3.3 Rulebase Functionality

Each candidate system is comprised of a pre-defined, constant, number of rules (although this strategy can be varied). Each rule is of the form IF A THEN B, where A is the antecedent and B is the consequent. The antecedent takes the form of a tree with a function set defined purely by the logical operator AND. The terminal set is a library of MA structures, each of which can be defined over a restricted region of past data. Two different terminals are used in this investigation, namely:

- $MA(a, b) < MA(c, d)$
- $MA(a, b) < MA(c, d) < MA(e, f)$

MA is the simple moving average, as defined in the previous section. The parameters [a,…,f] represent indices to the past data. Note that MA(a, a) returns the stock value a days into the past.

The terminal set of consequents is $\{BUY, SELL, DO\_NOTHING\}$. These are trading signals with the meanings 'buy shares', 'sell short', and 'remain outside the market' respectively.

When a rulebase is applied to a period of past data, each antecedent will produce an output, which can either be 1 or 0. If the value is 1, then the corresponding consequent is proposed as a candidate decision. If all candidates are identical then this becomes the rulebase decision, otherwise DO_NOTHING is chosen since the system is in a state of indecision.

4 Methodology

4.1 Problem Domain

This study considers the time series of closing prices of the Standard & Poor's 500 share index for the period 02-Jan-1990 to 18-Oct-2001 (see Figure 1). No pre-processing of the data was undertaken. The data was split into distinct training and test sets, following the approach adopted in [13]: training: 02-Jan-1990 to 28-Dec-1993, validation: 27-Oct-1993 to 10-Dec-1997, testing: 10-Oct-1997 to 18-Oct-2001.

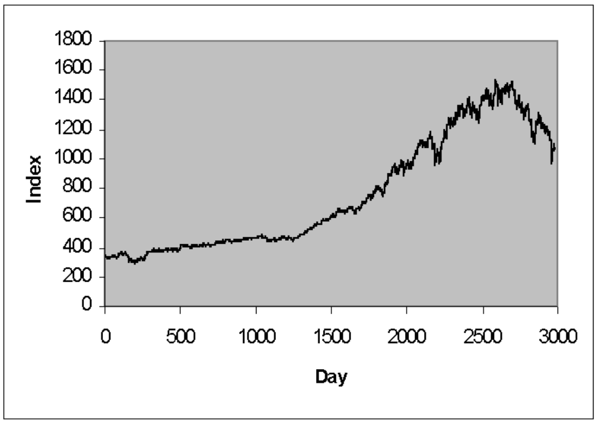

Fig. 1. Standard and Poor's 500 index for the period 02-Jan-1990 to 18-Oct-2001

A population of trading systems is inferred from the training set. The candidate models that results from this process are then validated on a further data set to check for overfitting. For each decision-point in the data set, the rulebase takes raw data as input and returns a BUY, SELL, or DO_NOTHING trading signal. A constant close-out period of 10 days for BUY and SELL strategies is adopted in this work. The size of investment corresponding to a BUY or SELL signal is assumed to be an arbitrary constant, $1,000.

The profit made by each transaction is adjusted to account for trading costs and slippage. The trading cost is defined as 0.2%, whilst slippage is set at 0.3% [13]. Note that trading costs for an individual speculator would be somewhat higher than those assumed here. Slippage is a catch-all for other factors that might deleteriously effect the profit made during a transaction. For example, the execution of a trade will be subject to delay and, in particular, it may not be possible to begin the transaction at the previous day's closing price.

The total profit for the system is the sum of all adjusted transaction profits, together with the risk free rate of return generated on uncommitted funds (those times when a DO_NOTHING signal was given). However, this profit is not used as the *fitness* of the candidate solution. This is discussed in the following sub-section.

4.2 Fitness Function

The GP search engine requires knowledge of the performance of each candidate solution within the current population in order to perform a directed stochastic search. Proposed fitness functions in this area of research tend to consider two criteria: profit made and associated risk. This latter term, whilst important, is largely dependent on the psychology of the investor. It is omitted from this study (thus performance is assessed purely on a profit measure, regardless of risk). Alternatively, if the preference of the investor is unknown, a multi-objective genetic programming (MOGP) technique could be used to generate a representation of Pareto optimal trading systems [14].

The fitness function utilised in this study is simply the summation of all adjusted transaction profits. Note that DO_NOTHING receives zero profit rather than the risk free rate of return, in order to promote the evolution of an active system. Since many solutions will correspond to negative fitnesses, this approach should somewhat attenuate the relatively positive effect of a DO_NOTHING signal. Whilst this approach may be regarded by some as incautious, any resulting high risk - high reward systems are likely to be exposed on the dual test sets.

4.3 Implementation details

Initialisation. A pre-defined number of rules are built for each candidate solution of the initial population. Each antecedent tree is restricted to within a pre-defined maximum number of nodes. Repetition of terminals is prevented. The tree is built in a top-down manner using the ramped half-and-half strategy [4]. Each consequent is generated by rotating through the terminal set.

Selection. Tournament selection, with a tournament size of 13, was used in this work. Selected individuals were then subjected to the genetic operators described below in order to generate new candidate solutions.

Genetic operators. All operations act on the antecedent trees only. Single-point binary crossover has been implemented (probability = 0.7 per pair of solutions). A check is made to ensure that the results of crossover are valid. A mutation operator that reinitialises branches of the antecedent tree has also been implemented (probability = 0.01 per solution node).

5 Results

In this initial investigation, the performance of two trading systems developed using the methodology described herein is compared to that of a benchmark buy-and-hold strategy.

The first trading system, labelled as the *complex system*, consists of a rulebase of 9 rules with antecedents of up to 7 nodes. This rulebase cannot be presented here due to size restrictions and is complicated to analyse. The second trading system, known as the *simple system*, has a rulebase of only 3 rules, where the number of nodes in the antecedent is limited to 1. Analysis of this system is somewhat more tractable.

An investment of $1,000 is associated with each decision. All transactions instigated by the rulebase are closed out after a fixed period of 10 days. Thus, the maximum investment at any one time is $10,000. When the decision is taken to DO_NOTHING, interest is earned at a risk free rate of 0.14% (the average 10 day rate derived from US Federal Funds historical data). Under the buy-and-hold strategy, the full investment of $10,000 is made at the beginning of the period and is closed out at the end of the period.

Results for the two systems, together with results for the buy-and-hold strategy, are presented in Table 1. The total profit produced by the strategies is decomposed into performance on the training, validation, and testing data segments.
Graphical results, depicting how the performance of each system varied over the total

Trading period	*Complex system*	*Simple system*	*Buy and hold*
Training	$4046	$3274	$4109
Validation	$1948	$2219	$10533
Testing	$1112	$1588	$1240
Total	$7106	$7081	$15882

Table 1. Profits made by various strategies

time series, are shown for the complex system in Figure 2 and for the simple system in Figure 3.
It is evident that neither of the evolved trading systems is capable of consistently outperforming the benchmark strategy. More complicated rulebases are able to make substantial profits on the training data (excess returns over buy-and-hold) but this is associated with poor performance on the validation and testing data sets. This is an indication of overfitting. The complex system shown above represents a trade-off solution across the training and validation data sets. The simple system is unable to match the complex system on the training data but is significantly more robust to changing market conditions. Indeed, the simple system beats the buy-and-hold strategy on the test set. Neither trading system is capable of capitalising on the major bullish period contained in the validation set (conditions under which a buy-and-hold strategy would be most successful).The mixed performance of the trading systems across the data sets may have arisen

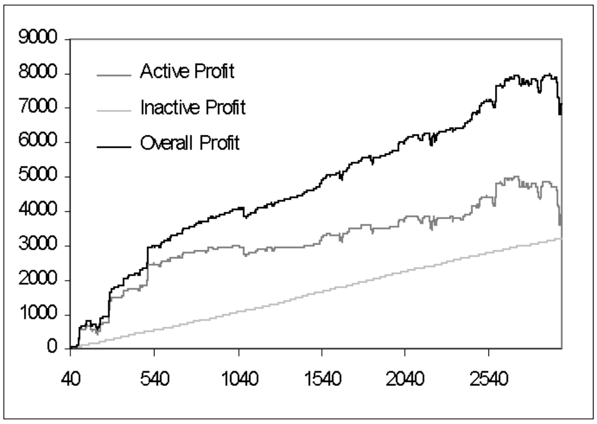

Fig. 2. Performance dynamics for the complex system: active and inactive profits arise from BUY/SELL and DO_NOTHING actions respectively

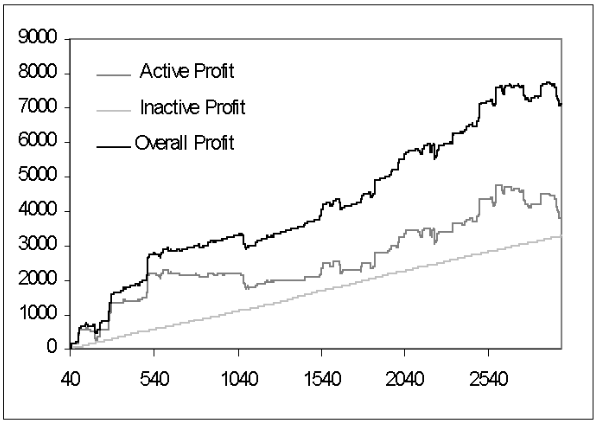

Fig. 3. Performance dynamics for the simple system

from insufficient market features in the training data. The bullish period in the validation data and the major bearish region of the test data are not visible in the training data. The GP-optimisation process may require a richer, or more representative, variety of training material (or rulebase building-blocks). This certainly warrants further investigation.

In order to perform a complete assessment of the results, the market risk profile of each trading strategy should also be accounted for. Since the buy-and-hold strategy makes a fixed investment of \$10,000 during the investment period, the maximum investment that can be lost is \$10,000. An average daily investment can be calculated for the trading systems developed in this study, assuming that selling short will never lead to a loss greater than the stake itself, in order to provide a measure of risk for the proposed systems. These results are shown in Table 2. The average daily investments for the

Data set	*Complex system*	*Simple system*
Training	$2351	$2134
Validation	$1580	$1680
Testing	$2755	$2208

Table 2. Average daily investment of trading systems

systems do not indicate any excess risk over the benchmark strategy.

6 Conclusions and Future Work

A GP-optimised rule-based trading system has been presented in this paper. The aims of this work have been two-fold: (1) to develop an effective system (that is able to outperform a benchmark strategy) with (2) a high degree of simplicity and transparency. Initial results indicate the importance of the rulebase complexity issue. A degree of complexity is required in order to generate excess returns, whilst extravagant complexity will lead to overfitting. A balance must be sought, which also embraces the requirement for transparency.

This study has considered only a single technical indicator, the MA. As stated earlier, this indicator has known strengths and weaknesses in its prediction capabilities. Also, only the simplest form of MA was considered here. Future extensions will seek to make other technical indicators available, such that the rulebase can utilise complementary indicators if this is deemed desirable. The library of possible indicators can be increased without bounds, given that the rulebase complexity remains unchanged, so long as the combination of these indicators remains tractable. Indeed, increasing the choice of indicators may help to prevent rulebase bloat.

The structure of the rulebase used in this study is somewhat restrictive. Extensions to the rulebase could be developed to provide extra flexibility whilst maintaining tractability. Fuzzification of the rulebase could be one such improvement, where the first step would be to replace the crisp equality relation within the MA terminals by fuzzy counterparts.

The application of the methodology to other market sources will form an interesting next step. It will be possible to see if the conclusions derived from the Standard & Poor's 500 share index concerning the methodology remain valid across other financial sectors. Furthermore, the inclusion of multiple market sources in the prediction of a single index may prove rewarding.

A promising system has been yielded through the use of highly limited technical indicators. From this foundation, it is hoped that simple enhancements will produce a system capable of consistent success over benchmark strategies.

Acknowledgments

This paper is the first product of an international collaboration that was initiated at the EvoNet Summer School 2001 held between 27-Aug and 01-Sep in Thessaloniki, Greece (http://evonet.dcs.napier.ac.uk/summerschool2001/). The authors' would like to express their thanks to everyone involved in the summer school, with special thanks to Conor Ryan who set the original problem from which this work evolved.
For the opportunity to attend the school, the authors would like to thank the Dept. of Computer Science at the University of Liverpool and Dr R. C. Paton (C. Setzkorn); Scuola Superiore S. Anna of Pisa, Prof. P. Dario, and Prof. A. M. Sabatini (L. Dipietro); Prof. P. J. Fleming (R. Purshouse).

References

1. Achelis, S.: Technical analysis from A to Z. 2nd edition. Irwin Professional Publishing, Chicago (1997)
2. Allen, F., Karjalainen, R.: Using genetic algorithms to find technical trading rules. Journal of Financial Economics **51** (1995) 245–271
3. Brock, W., Lakonishok, J., LeBaron, B.: Simple technical trading rules and the stochastic properties of stock returns. Journal of Finance **47** (1992) 1731–1764
4. Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D.: Genetic programming: an introduction-On the automatic evolution of computer programs and its applications. Morgan Kaufman Publishers (1998)
5. Deb, K.: Multi-objective optimization using evolutionary algorithms. John Wiley and Sons Ltd, Chichester (2001)
6. Fama, E.F.: Efficient capital markets: a review of theory and empirical work. Journal of Finance **25** (1970) 383–417
7. Golan, R.H., Ziarko, W.: A methodology for stock market analysis utilizing rough set theory. Proceedings of the IEEE/IAFE 1996 Conference on Computational Intelligence for Financial Engineering (1995) 32–40
8. Hellstrm, T., Holmstrm, K.: Predicting the stock market. Technical report series IMa-TOM-1997-07, Malardalen University, Sweden (1998)
9. Hiemstra, Y.: A stock market forecasting support system based on fuzzy logic. Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences (1994)
10. Koza, J.R.: Genetic Programming. MIT Press (1992)
11. Murphy, J.J.: Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. New York Institute of Finance (1986)
12. Neely, C.J., Weller, P.A.: Technical trading rules in the European Monetary System. Journal of International Money and Finance **18** (1999) 29–58
13. O'Neill, M., Brabazon, T., Ryan, C., Collins, J.J.: Developing a market timing system using grammatical evolution. Proceedings of GECCO (2001) 1375–1381
14. Rodriguez-Vazquez, K., Fonseca, C.M., Fleming, P.J.: Multiobjective genetic programming : a nonlinear system identification application. Late Breaking Papers at the 1997 Genetic Programming Conference (1997) 207–212
15. Schierholt, K., Dagli, C. H.: Stock market prediction using different neural network classification architectures. Proceedings of the IEEE/IAFE 1996 Conference on Computatational Intelligence for Financial Engineering (1996) 72–78