Discovering Effective Technical Trading Rules with Genetic Programming: Towards Robustly Outperforming Buy-and-Hold

Dome Lohpetch
School of MACS, Heriot-Watt University
Edinburgh, UK
dl73@hw.ac.uk

David Corne
School of MACS, Heriot-Watt University
Edinburgh, UK
dwcorne@macs.hw.ac.uk

***Abstract*—Genetic programming is now a common research tool in financial applications. One classic line of exploration is their use to find effective trading rules for individual stocks or for groups of stocks (such as an *index*). The classic work in this area (Allen & Karjaleinen, 99) found profitable rules, but which did not outperform a straightforward "buy and hold" strategy. Several later works report similar outcomes, while a small number of works achieve out-performance of buy and hold, but prove difficult to replicate. We focus here on indicating clearly how the performance in one such study (Becker & Seshadri, 03) was replicated, and we carry out additional investigations which point towards guidelines for generating results that robustly outperform buy-and-hold. These guidelines relate to strategies for organizing the training dataset, and aspects of the fitness function.**

Keywords- stock trading, technical trading rules, genetic programming

I. INTRODUCTION

Financial markets are complex and ever-changing environments in which groups of individuals, companies and other investors are competing for profit. There are many opportunities in this area for advanced machine learning and optimization methods to be exploited, and it is now common to see applications of evolutionary computation applied to the financial markets [1]. Genetic Programming (GP) [2—4] is particularly prominent in terms of the degree to which it has recently been applied in finance [5—12].

In this paper we focus on the specific area in finance known as *technical analysis* [13—16]. Usually used to forecast future prices or price movements of specific stocks or collections of stocks, technical analysis is a financial markets technique that forecasts the future direction of security prices via the study of historical market data (primarily price data and trading volume data). Many different methods and tools are utilized in technical analysis, all of which rely on the principle that price patterns and trends exist in markets, and that these can be identified and exploited.

Common tools in technical analysis are the use of indicators such as moving averages (the mean value of the price for a given stock or index over a given recent time period), relative strength indicators (a function of the ratio of recent upward movements to recent downward movements), and others. There have been a number of attempts to use GP in technical analysis for acquiring technical trading rules, and a typical strategy is for such a GP-produced rule to be a combination of technical indicator 'primitives' with other mathematical operations. This leads to a 'signal'. E.g. a GP approach may involve finding both a good buy signal and a good sell signal – that is, one rule which, if its output is above 0, indicates that it is a good time to buy, and a different rule indicating when it is a good time to sell.

The classic first attempts to use GP in technical trading analysis were by Chen and Yeh [5] and Allen and Karjalainen [7]. However, although it was demonstrated that GP could produce profitable rules for the stock exchange markets, their performance did not show any excess returns when compared to the standard buy-and-hold approach. 'Buy-and-hold' simply means, for a given trading period, buy the stock at the beginning of the period, and sell at the end – hence, always a good strategy in a market that generally moves up during the period. More recent applications of GP in this context have been more encouraging [8,17,18]

In particular, GP-evolved technical trading rules, which can outperform buy-and-hold (at least if dividends are excluded from stock returns) have been found in studies by Becker and Seshadri [19—21]. They adopted the overall approach of Allen and Karjaleinen [7] (which we will refer to hereafter as "AK"), and introduced several changes, which we discuss later. In work by others, Potvin et al [12] illustrated that the trading rules explored by GP can be generally beneficial when the market is falling or when it is stable, while Fransworth et al [22] also demonstrated that GP can be used to identify predictable patterns in financial asset prices.

One line of work has attempted to gain on the buy-and-hold strategy by including risk metrics in the rules (or in their evaluation). Typically, a risk measure such as the Sharpe ratio [23] is used to normalize the estimate of financial return. Effectively, this downgrades the performance of rules that promote trading in volatile conditions, and therefore tends to promote the generation of rules more likely to be applied by investors. For example, in attempting to build on work by Fyfe et al. [6], which used GP to discover a successful buy rule, but which was not superior to buy and hold, Marney et al. [8, 17] modified this work by including the use of metrics for calculating risk; however, the rules found in their work still did not compare well with simple buy-and-hold strategy. Also, a study by Cheng and Khai [10] using a modified Stirling return measure as a risk adjustment, but this did not lead to a reasonable rate of investment profit after the transaction costs were taken into account. More recently, Marney et al [25] used the Sharpe ratio and found that the technical trading rule in their study easily outperformed simple buy and hold, in terms of unadjusted returns; however, when returns were risk adjusted, it could be seen that technical trading still underperformed simpler strategies.

Although we expect that the incorporation of risk measures (particularly in the context of an appropriate multiobjective approach) will eventually yield benefits in terms of robust trading rules, we are drawn first to the line of work pursued in [7,19—21], in which Becker and Seshadri were able to outperform buy-and-hold via modifying the AK approach. The remainder of this paper details the overall approach (section II), taking care to ensure enough information is given for replication (this is not really the case in [7,19—21], and summarizes the findings of several experiments that start to reveal how GP might be used to reliably produce technical trading rules that can outperform buy-and-hold (section III); we then have a concluding discussion in section IV, and point to where the reader may obtain our code for further experimentation.

II. THE AK APPROACH AS MODIFIED BY BECKER AND SESHADRI

A. Overview

The approach we experiment with in this paper is based on that in [19,21] which in turn modified AK [7]. Standard GP is used, with a function set comprising a common set of arithmetic, Boolean and relational operators, while the terminal set comprises a collection of basic technical indicators, along with real and Boolean constants, and real-valued variables (such as *stock price*). An example of a specific rule found by [19] is in Fig. 1.

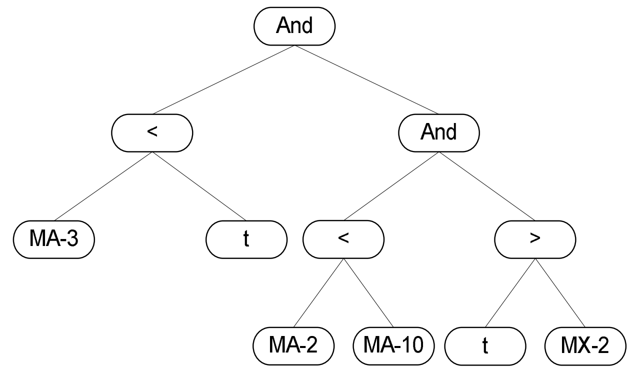

Figure 1. Example of a trading rule found by GP.

The interpretation of the rule in Fig. 1 is as follows. **“**the 3-month moving average (MA-3) is less than the lower trend line (t) and the 2-month moving average (MA-2) is less than the 10-month moving average (MA-10) and the lower trend line (t) is greater than the second previous 3-month moving average maxima (MX-2)**”**. It therefore has a Boolean result, either true or false. A rule leads to trading behaviour as follows: “If you are currently out of the market and the rule yields true, then *buy*; if you are currently in the market and the rule becomes *false*, then sell.”. This procedure assumes a fixed amount to invest (e.g. $1,000) whenever there is a buy signal.

We explain the approach in more detail in the remainder of this section, paying particular attention to the fitness function; this description includes several key details that were not explicit in [7,19,21], but which we have discovered essential in order to achieve appropriate replication and good performance. The approach we explain is that of Becker and Seshadri, and we note here the modifications they made to the AK approach. These were: the use of monthly data rather than daily data; a reduced function set, but a larger terminal set, with more indicators; the use of a complexity-penalising factor in the fitness function to avoid over-fitting; modifying the fitness function to consider the number of periods with well-performing returns, rather than just the total return over the test period.

Finally, the date we use (same as [7,19—21] is the Standard and Poors 500 (S & P 500) index (see Fig. 2) – that is, our ‘portfolio’ is simply the fixed set of 500 stocks, which, aggregating over all of the included stocks, give us a daily price indicator, such as the opening price. E.g. the value for a given day is the sum of the opening prices of all S&P500 stocks for that day. Actually, there are four such indicators per day: opening price, closing price, daily low, and daily high. As indicated, we work with monthly data – that is, 12 sets of values per year, where open, close, low and high for a particular month are taken to be, respectively: the opening price on the first day of the month, the closing price on the last day of the month, the lowest price reached during the month, and the highest price reached during the month.

B. *Function and Terminal Sets*

The function set comprises simply the Boolean operators and, or and not, and the relational operators > and <. The terminal set comprises the following (explained in more detail below):

- opening, closing, high and low prices for the current month;
- 2,3,5 and 10-month moving averages;
- Rate of change indicator: 3-month and 12-month;
- Price Resistance indicators: the two previous 3-month moving average minima, and the two previous 3-month moving average maxima;
- Trend Line Indicators: a lower resistance line based on the slope of the two previous minima; an upper resistance line based on the slope of the two previous maxima.

The *n*-month moving average at month *m* is the mean of the closing prices of the *n* months from *m* back to month $m-(n-1)$. The *n*-month rate of change indicator measured at month *m* is: $(c(m) - c(m-(n-1))\times 100)/c(m-(n-1))$, where $c(x)$ indicates the closing price for month *x*. Previous maxima MX1 and MX2 are obtained by considering the 3-month moving averages at each point in the previous 12 months. Of the two highest values, the one closest in time to the current month is MX1, and the other is MX2. the two previous minima are similarly defined. Finally, to identify trend line indicators, the two previous maxima are used to define a line in the obvious way, and the extrapolated value of that line from the current month becomes the upper trend line indicator; the lower trend line indicator is defined similarly by using the two previous minima.

C. *The Fitness Function*

The fitness function has three main aspects. First is the 'excess return'. This indicates how much would have been earned by using the trading rule, over and above the return from a buy-and-hold strategy. The other two aspects of the fitness function were introduced by Becker and Seshadri to avoid overfitting. These were a modification to fitness that preferred trading rules to be less complex, and a further modification that considered 'performance consistency' (PC). The details of the latter components were not clear in Becker and Seshadri's publications, but the methods we describe are able to achieve results consistent with theirs. We now deal with each aspect of the fitness function in turn.

The excess return is simply $E = r - r_{bh}$, where *r* is the return on an investment of \$1,000, and r_{bh} is the corresponding return that would have been achieved using a buy and hold strategy. To calculate *r* we use [7,19,21]:

$$r = \sum_{t=1}^{T} r_t I_b(t) + \sum_{t=1}^{T} r_f(t) I_s(t) + n \ln\left(\frac{1-c}{1+c}\right)$$

where: $r_t = \log P_t - \log P_{t-1}$ -- indicating the continuously compounded return, where P_t is the price at time *t*. Meanwhile, $I_b(t)$ indicates the buy signal, and is 1 if the rule indicates *buy* at time *t*, 0 otherwise. Similarly defined is the sell signal, $I_s(t)$. The first component of *r* therefore calculates the return on investment over the times when the investor is (as guided by the rule) in the market. In the second component, $r_f(t)$ indicates the risk-free return, which is taken for any particular day *t* from published US Treasury bill data (these data are available from http://research.stlouisfed.org/fred/data/irates/tb3ms). Hence, the second component represents time out of the market, in which it is assumed that the investor's funds are maintained in an account earning a standard risk-free interest. Finally, the third component is a correction for transaction costs. The cost of a single buy or sell transaction is assumed to be 0.25% (i.e. 0.0025) – e.g. \$2.50 for a transaction of volume \$1,000. The number of transactions actioned during the period by the rule is *n*. This component estimates the compounded loss from the expenditure on transactions.

The second main part of the fitness function, r_{bh}, is calculated as:

$$r_{bh} = \sum r_t + \ln\left(\frac{1-c}{1+c}\right)$$

where r_t is as indicated above. Hence it calculates the return over the period from risk-free investment in US Treasury bills, involving a single buy transaction.

The excess return *E*, calculated as described, was originally the objective function in [7], but improvements are recorded in [19,21] by adopting measures to counteract over-fitting. One of these is an adjustment to fitness according to the size of the tree. Given a fitness value *f*, the adjusted fitness becomes $5f/\max(5,depth)$, where *depth* is the depth of the tree being evaluated, and the constant 5 is a 'desired' depth. Clearly there are many ways to adjust fitness to take account for tree complexity, and we simply adopt the stated method, since this is what was used in [19,21]. The other aspect of the fitness function which led to more consistent results was as follows, which we call *Performance Consistency* (PC). The excess return *E* is calculated for each successive period of *K* months covering the entire test period. The value returned is simply the number of these periods for which *E* was greater than both the corresponding buy and hold return (from investing in the index over that period) and the risk-free return during that period. For example, for 12-month periods and a 5 year test period, there are 5 such successive periods – the fitness value returned is therefore simply an integer from 0 to 5.

Finally we can state the objective function f used in this work: the fitness of a GP tree was the PC-based fitness (i.e. a number from 0 to X, where there were X periods covering the test data), adjusted for tree complexity by $5f/\max(5,depth)$.

D. *Operators and Initialisation*

We used the four mutation operators described by Angeline [3], as follows:

- The grow mutation operator: randomly selects a leaf from the tree and replaces it with a randomly generated new subtree.
- The shrink mutation operator: randomly select an internal node from the tree and replace the subtree below it with a randomly generated terminal node.
- The switch mutation operator: randomly select an internal node from the tree and reorder its argument subtrees.
- The cycle mutation operator: select a random node and replace it with a new node of the same type. If a terminal node is selected, then it is replaced by a terminal node. If an internal node is selected, then it is replaced by a function that takes an equivalent number of arguments.

We used standard subtree-swap crossover [2]. Finally, we note that the population was initialized by growing trees to a maximum depth of 5, however no constraint was placed on tree size beyond the initial generation, other than the pressure towards compact trees offered by the objective function.

III. EXPERIMENTS

A. *GP Parameters*

In all the experiments we describe, the GP system was as described in the last section, and used a population size of 500. In each generation, the current best was copied into the next generation, and the remainder were the product of 499 offspring-production events in which each such event was: decide crossover (probability 0.7) or mutation. If crossover, select two parents by rank-based selection, perform crossover; if mutation, select one parent, and apply mutation (choosing uniformly from the available mutation operators). Each run continued for 50 generations.

B. *Data Period and Fitness Period Variation*

In a given experiment, a 31-year period (in common with [7] and [19] is used for training – this is 1960—1991. However, we explored two different regimes for choosing and evaluating a rule from the training run. In regime 1, the fittest rule found during training (as measured on the training date) was applied to test data in an immediately succeeding period of N years. In regime two, each rule found during training was validated against the ensuing N year period, and the rule that was best during this validation period was chosen, and tested over a period from year $31+N+1$ to $31+N+K$. These two regimes were each explored for 3 data period splits:

- Split1: 31 yrs training; N=12, K=5
- Split2: 31 yrs training, N=8, K=8
- Split3: 31 yrs training, N=9, K=9

In [7,19,21] the same training period is used, with testing only on the subsequent 12-year period. Although not explicit, we think that the rules were chosen for evaluation in [7,19,21] via what we call regime 1. In Fig. 2, we show the S&P Index during the years 1960—2008, indicating the different splits used in the experiments. Notice how the different splits explore salient challenges for the rule development process. In Splits 2 and 3, when evaluation regime 1 is used, the test period is one in which the market grew strongly, at a faster overall rate than during the training period. Outperforming B&H is always a stringent challenge, but moreso in such a period of growth. In split 1, the challenge is more varied, with the test period covering 9 years of growth followed by 3 of decline. Meanwhile, when regime 2 is used for Split 1, we expect validation over a varied period (split1 V) to help select a rule that performs well in a growth-only period. Again, this is a significant challenge to outperform buy and hold. With splits 2 and 3, regime 2 is challenged to produce a rule that performs well over a period that is roughly half decline and half growth, despite training and validation being done over periods largely in growth,

We experimented also with the evaluation periods within the Performance Consistency (PC) term of the fitness function. In Becker and Seshadri's work, the employment of the PC term clearly results in improved performance (this is also true in our replication; we omit the comparative results for reasons of space). However they only report on the use of 12-month periods. We experiment with four different lengths for the "PC period", namely 6, 12, 24 and 30 months. These are referred to later as PC6 PC12, PC224 and PC36.

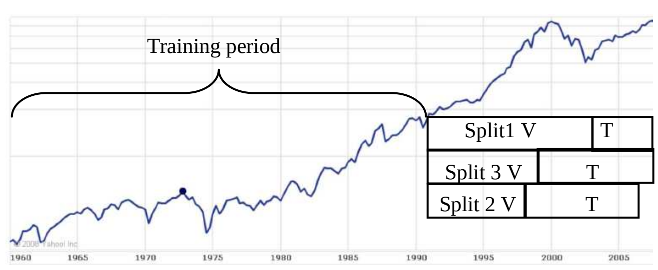

Figure 2. The S&P500 index over the periods examined. The training period is 1960—1991, and three data Splits are shown. "V" indicates either a test set used for evaluation regime 1, in which case the rule tested is the one that performed best over the training period. "T" indicates the test set for evaluation regime 2, in which case the rule tested is the one which, during training, performed best on the "V" period.

C. Results

Following many preliminary investigations, which converged on confirming that the parameters and strategies recorded in the previous section were consistent with the performance reported in [17, 21], we performed 10 runs each for each of the 24 scenarios (3 data Splits, 4 PC periods, 2 rule selection and evaluation regimes). All results are summarized in Table I. In the table, each row indicates the mean over 10 runs of two pairs of related figures. First we give the mean excess over Buy and Hold during the training period. That is, if investing $1,000 in a buy and hold strategy would yield $5,000 over the training period, and the selected rule yielded $20,000, this value would be $15,000. Next, we express the latter as a ratio – the return from the rule divided by the return from buy-and-hold. Note that these two figures are of course the same for regimes 1 and 2. The final two columns are more interesting, again expressing the excess and ratio in comparison to buy and hold, but this time on the appropriate test set.

TABLE I. SUMMARY OF RESULTS FOR EACH DATA SPLIT, PC PERIOD AND EVALUATION REGIME

Data split	PC Period	Eval.	Excess over BH (train)	Excess over BH (train /ratio)	Excess over BH (test)	Excess over BH (test/ rato)
1	6	1	$19,750	4.38	$1,919	1.73
		2			$142	1.09
2	6	1	$17,890	4.46	$175	1.05
		2			$611	1.53
3	6	1	$23,800	5.36	$588	1.13
		2			$1,081	2.78
1	12	1	$29,135	6.39	$2,283	1.87
		2			$107	1.06
2	12	1	$33,077	7.06	$5	1.002
		2			$691	1.61
3	12	1	$37,413	7.86	$358	1.08
		2			$1,037	2.70
1	18	1	$46,641	8.63	$1,706	1.65
		2			$103	1.06
2	18	1	$39,022	8.15	-$492	0.87
		2			$871	1.76
3	18	1	$41,161	8.54	-$437	0.90
		2			$1,299	3.14
1	24	1	$43,364	8.95	$1,788	1.67
		2			$98	1.06
2	24	1	$39,249	8.19	-$491	0.86
		2			$655	1.57
3	24	1	$44,211	8.92	-$736	0.83
		2			$913	2.49

To help digest the raw results in Table I, Table II provides summary notes, focusing on the test set results for each regime, as a function of the data splts.

TABLE II. SUMMARY NOTES COMPARING DIFFERNET SETUPS

Experiment Setup	Results – evaluation regime 1	Results – evaluation regime 2
Split 1: 12 years V, 5 years T (see fig. 2)	Always outperformed B&H, for all PC periods, by around 60—80%	Consistent but modest excess over B&H, around 5—10%,
Split 2: 8 years V, 8 years T (see fig. 2)	Modest excess over B&H for PC6 and PC12, but loses to B&H for longer PC periods.	Around 50-80% improvement over B&H, consistently better, in all PC periods.
Split 3: 9 years V, 9 years T (see Fig. 2)	Modest improvemt over B&H for PC6 and PC12, but loses to B&H in longer PC periods	Consistently 250—300% improvement over B&H, for all PC periods.

As we can see from the results in Table I, and their summary in Table II, the use of evaluation regime 2, in which the rule we select for trading is one that has been cross-validated on an intervening period, provides more reliable and consistent results. This in itself is not surprising, but it suggests that this additional protection against over-fitting (over and above the measures used in the design of the fitness function) is worth using. This approach always yielded rules that outperformed B&H and risk-free investment, whether the market was consistently rising during the test period (Split 1) or mixed (Splits 2 and 3), although the excess over B&H for the consistently rising market was modest, but was far from trival.

When Split 1 was used (corresponding to the training and test set used in [7,19,21], we find that regime 1 (as also used in [7,19,21] always outperforms B&H, but this is not so for the other data splits. Regime 1 simply means using all available data fully for the training process, with the intention of putting the resulting chosen rule into operation at the first opportunity after training – in practice, our data split 1, regime 1, evaluates how well this strategy would work if it were currently the end of 1991, and we generated a strategy (training on 1960—1991 data) to use from 1992 onwards. It is quite attractive to use all available data in this way, and this is clearly what was done in [7,19,21]; however, our experiments suggest that the positive performance in these works was probably lucky, since this strategy is still likely to overfit, and is sensitive to the data split. Earmarking the most recent period to use as a validation set (regime 2) is clearly more successful.

IV. CONCLUDING DISCUSSION

The discovery of technical trading rules by Genetic Programming is an interesting and active thread of research, in which a string of research articles have been published in the past decade, exhibiting varied levels of success. The chief difficulty is that, although successful rules can be found, these are often not competitive with the much simpler strategy of exploiting the opportunity for risk-free investments, or (in upwardly moving markets) using a buy-and-hold strategy for the test period. Some work has shown success in that regard, however [19,21], and we have replicated and explored that work further in this paper. In particular, we have shown that the basic setup used in [19,21] is sensitive to the data periods involved, and it is clearly better to use a validation set to choose the rule. And, concerning period length in the *Performance Consistency* aspect of the fitness function, shorter periods seem more robust – in our experiments the only examples of underperformance compared with buy-and-hold occurred with longer periods of 24 months and 18 months (both with the simple regime 1 evaluation setup).

So, with modification to the approach used for choosing the rule to evaluate by using a validation set, our experiments find that the Becker & Seshadri variations on the original AK approach can provide robust out-performance over buy and hold. Such out-performance is revealed in three different data-split scenarios, one involving an upward market during the text period, and the other two being more volatile. Interested researchers can pick up our source code at http://www.macs.hw.ac.uk/~dwcorne/gptrcode.

Although the work reported here has some clear limitations – e.g. we intend to explore a much wider range of data splits, and gain an understanding of how performance varies given the nature of the market movements in the training, validation and test periods – this has served as a basic replication, validation and extension of the approach, and confirms this line of work as promising for future exploration. Additional directions include the use of multi-objective formalizations, to provide a more principled way to handle both the performance consistency and complexity aspects of the fitness function, and also we would like to do a comparison with gene expression programming, which is now being explored in financial applications [1]. Finally, we note that one of the major differences between AK's original work and Becker & Seshadri's improvement was the move from daily to monthly trading. It is not clear whether or not the current approach will be successful (in terms of outperforming buy-and-hold) in the context of daily, or even weekly, trading. The contributing factors in this include the general volatility in the market, and how this varies over different timescales, as well as the relatively different challenges for optimization that emerge from larger datasets. These are all issues we are exploring in ongoing work.

References

[1] Brabazon, A. & O'Neill, M. (2005). Biologically Inspired Algorithms for Financial Modelling (Natural Computing Series), New York: Springer.

[2] Koza, J. R. (1992). Genetic Programming: On the Programming of Computers by means of Natural Selection, Cambridge, MA: MIT Press.

[3] Angeline, P. J. (1996). Genetic Programming's Continued Evolution. *Advances in Genetic Programming*, Vol. 2, editor, P. J. Angeline and K. Kinnear, Cambridge, MA: MIT Press, pp. 89-110.

[4] Banzhaf, W., Nordin, P., Keller, R. E. & Francone, F. D. (1998). *Genetic Programming - An Introduction*: On the Automatic Evolution of Computer Programs and Its Applications, San Francisco: Morgan Kaufmann

[5] Chen, S. H. & Yeh, C. H. (1996). Toward a Computable Approach to the Efficient Market Hypothesis: An Application of Genetic Programming, *J. of Economic Dynamics & Control*, 21: 1043-1063.

[6] Fyfe, C., Marney, J. P. & Tarbert, H. (1999). Technical Trading versus Market Efficiency: A Genetic Programming Approach, *Applied Financial Economics*, 9: 183-191.

[7] Allen, F. & Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules, *Journal of Financial Economics*, 51:245-271.

[8] Marney, J. P., Fyfe, C., Tarbert, H. & Miller, D. (2001). Risk Adjusted Returns to Technical Trading Rules: A Genetic Programming Approach, *Computing in Economics and Finance*, Soc. for Computational Economics, Yale University, USA, June 2001.

[9] Chen, S. H. (2002). Genetic Algorithms and Genetic Programming in Computational Finance, Boston, MA: Kluwer.

[10] Cheng, S. L. & Khai, Y. L. (2002). GP-Based Optimisation of Technical Trading Indicators and Profitability in FX Market, *Proceeding of the 9th International Conference on Neural Information Processing (ICONIP' 02)*, Vol. 3, pp. 1159-1163.

[11] Farnsworth, G. V., Kelly, J. A., Othling, A. S. & Pryor, R. J. (2004). Successful Technical Trading Agents Using Genetic Programming, SANDIA Report SAND2004-4774, Sandia National Laboratories.

[12] Potvin, J. Y., Soriano, P. & Vallée, M. (2004) Generating Trading Rules on the Stock Markets with Genetic Programming, *Computers and Operations Research*, Vol. 31, Issue 7 (June 2004): 1033 - 1047

[13] Pring, M. J. (1980). *Technical Analysis Explained*, New York: McGraw-Hill.

[14] Ruggiero, M. A. (1997). *Cybernetic Trading Strategies*, NY: Wiley.

[15] Murphy, J. J. (1999). *Technical Analysis of the Financial Markets*, New York: New York Institute of Finance.

[16] Lo, A. W., Mamaysky, H. & Wang, J. (2000). Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation, *The Journal of Finance*, 55:1705-1770, 2000.

[17] Marney, J. P., Miller, D., Fyfe, C. & Tarbert, H. (2000). Technical Analysis versus Market Efficiency: A Genetic Programming Approach, *Computing in Economics and Finance*, Society for Computational Economics, Barcelona, Spain, July 2000 (paper #169).

[18] Neely, C. (2001). Risk-adjusted, ex ante, optimal technical trading rules in equity markets, Working Papers 99-015D, Revised August 2001, Federal Reserve Bank of St. Louis.

[19] Becker, L.A. & Seshadri, M. (2003). Comprehensibility and Overfitting Avoidance in Genetic Programming for Technical Trading Rules, Worcester Polytechnic Institute, Computer Science Technical Report WPI-CS-TR-03-09.

[20] Becker, L.A. & Seshadri, M. (2003). Cooperative Coevolution of Technical Trading Rules, Worcester Polytechnic Institute, Computer Science Technical Report WPI-CS-TR-03-15.

[21] Becker, L.A. & Seshadri, M. (2003). GP-evolved technical trading rules can outperform buy and hold, In *Proc. 6th Int'l Conf. on Computational Intelligence and Natural Computing*, North Carolina USA, September 26-30 2003.

[22] Farnsworth, G. V., Kelly, J. A., Othling, A. S. & Pryor, R. J. (2004). Successful Technical Trading Agents Using Genetic Programming, SANDIA Report SAND2004-4774, Sandia National Laboratories.

[23] Sharpe, W. F. (1966). "Mutual Fund Performance". *Journal of Business* **39** (S1): 119–138. doi:10.1086/294846.

[24] Marney, J. P., Tarbert, H. & Fyfe, C. (2005). Risk Adjusted Returns from Technical Trading: A Genetic Programming Approach, *Applied Financial Economics*, 15: 1073-1077.