Adaptive Trading with Grammatical Evolution

Ian Dempsey, Michael O'Neill and Anthony Brabazon

Abstract— **This study reports on the performance of an on-line evolutionary automatic programming methodology for uncovering technical trading rules for the S&P 500 and Nikkei 225 indices. The system adopts a variable sized investment strategy based on the strength of the signals produced by the trading rules. Two approaches are explored, one using a single population of rules which is adapted over the lifetime of the data and another whereby a new population is created for each step across the time series. The results show profitable performance for the trading periods explored with clear advantages for an adaptive population of rules.**

I. Introduction

Following on from previous studies, which use Grammatical Evolution (GE) to evolve financial trading systems [1], [2], [3], [4], [5], [6], [7] the objective of this study was to perform an evaluation of adaptive evolution and trading using GE where the system is set "live" in a dynamic environment. Previous work in this area approached the problem by first evolving trading rules over a training set and then applying the best rules to the out-of- sample or 'test' data. While these studies demonstrated the utility of GE for financial prediction, their results deteriorated over time as the trading rules remained static over the out-of-sample period. In this study the evolved trading system is adaptive, in that it continues to retrain as new data becomes available using a variant on the *moving window* approach, while still maintaining a *memory* of past good rules. This permits the system to adapt to dynamic market conditions, while not completely losing the memory of good solutions that worked well in past market environments. It also has the benefit of of being able to take variable sized positions in the market depending on the strength of the signal produced by the technical rules. The population of trading rules is initially created using a set of training data in the usual manner, and after a certain number of generations the system goes 'live' and begins actively trading. The trading rules undergo more generations of evolution as new, additional, data becomes available during the out-of-sample period

This paper continues with a brief description of Grammatical evolution in Section II followed by an examination of the problem domain including the trading methodology used in Section III, and results are presented in Section IV. Finally the paper concludes in Section V.

II. Grammatical Evolution

Grammatical Evolution (GE) [12], [13], [11] is an automatic programming system, similar to Genetic Programming (GP) [10], in that it uses an evolutionary process to generate computer programs. Unlike GP, GE uses a population of linear binary strings, and a mapping process that transforms these strings into programs. This transformation is governed through the use of a BNF (Backus-Naur Form) grammar, which specifies the language of the produced solutions.

The use of a mapping process creates a distinction between the search space and the solution space; genotype strings are evolved with no knowledge (or respect) of their phenotypic equivalent, apart from a fitness measure. This has been shown to create a phenotypic "ripple effect" when standard genetic operators are applied to the genotype strings, but evidence suggests that this effect promotes a useful exchange of derivation subsequences during crossover events [15].

A. Mapping Process

An example of the mapping process employed by GE is shown in Figure 1. Starting with a binary string, an integer string is evolved, typically by using 8 bits per *codon*; these are then used to choose rules from a given BNF grammar, to generate a phenotypic program.

Starting with a given *start symbol*, a production associated with that symbol is chosen to replace it, by using the first codon available on the integer string. In the example, the codon 8 is mapped into the number of available productions associated with <E>, which are 6, so $8 \; mod \; 6 = 2$, and <E> is replaced with the sequence (-<E><E>).

The mapping process continues in this manner, always replacing the leftmost non-terminal symbol by a production associated with it on the grammar, which is chosen by a codon; the mapping terminates and is successful when a sequence of terminal symbols is generated.

III. The Trading Methodology

```
<expressions> ::= <expressions> <op> <exp>
                | <exp>
<exp> ::= MA( <numExp> ) | <numExp>
<op> ::= - | + | * | /
<numExp> ::= <int> <op> <int> | <int>
<int> ::= <int><digit> | <digit>
<digit> ::= 0|1|2|3|4|5|6|7|8|9
```

In the trading model developed, the rules evolved by GE using the above grammar [6], generate a signal that can result in one of three actions to be taken by the model: Buy, Sell or Do Nothing. The <expressions> non-terminal produces the signal through expressions using the MA terminal which is a moving average function.

Ian Dempsey, Michael O'Neill and Anthony Brabazon are with the Natural Computing Research & Applications Group, University College Dublin, Belfield, Dublin 4, Ireland (email: ian.dempsey@gmail.com, m.oneill@ucd.ie and anthony.brabazon@ucd.ie).

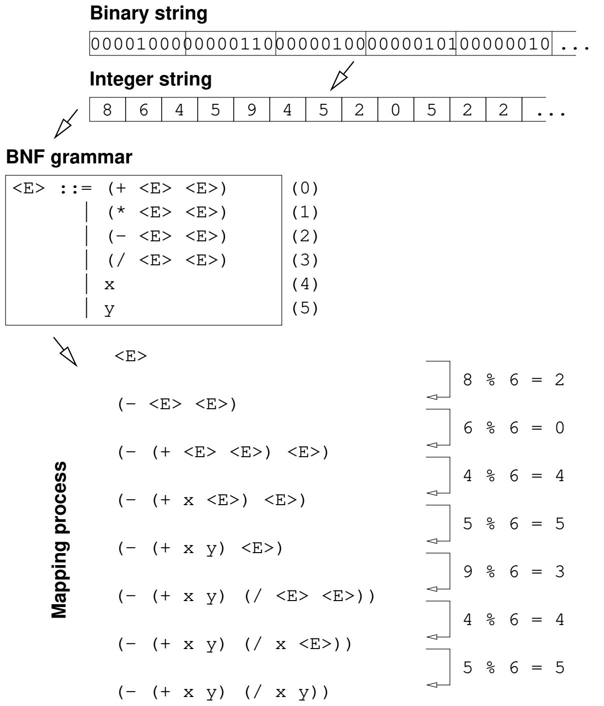

Fig. 1. GE mapping process

A. Variable Position Trading

The *live GE* methodology adopted firstly behaves as in earlier studies. An initial training period is set aside on which the population of proto-trading rules is trained, with the aim that a competent population is evolved after a certain number of generations, G. The system then goes 'live', and a number of actions are taken by the system. The trading system takes the best performing rule from the initial training period, and uses this rule to trade for each of the following x days. After x days have elapsed, the *training window* moves forward in the time series by x days, and the population is retrained over the new data window for a number of generations g, where $g < G$. This embeds both a *memory* and an *adaptive potential* in the trading system, as knowledge of good past trading rules is not completely lost, rather it serves as a starting point for their subsequent adaptation.

A small value of g means that memory is emphasised over adaptation, as the new data has relatively less chance to influence the trading rules. This could be considered a tuning parameter that could be used to alter the adaptive characteristics of the system, and could itself be open to adaptation. For example, in periods of rapid market change a trading system with a 'long memory' could be disadvantageous, whereas in stable periods, a longer memory could well be advantageous. Similar comments can be made for the parameter x. If its value is large, the trading rules are altered less frequently, but each adaptive 'step' will tend to be larger.

In prior studies which applied GE to evolve financial trading systems, the *entry strategy* for each trade was to invest a constant dollar amount on the production of a Buy or Sell signal. The relative strength of the buy or sell signal was not considered. In contrast in this study, the trading system adopts a more complex entry strategy, and a *variable size* investment is made, depending on the strength of the trading signal. The stronger the signal the greater the amount invested, subject to a maximum investment amount of \$1,000 (arbitrary). Therefore the amount invested for each signal is:

$$Amount\ invested\ = \frac{Size\ of\ trading\ signal}{Maximum\ trading\ signal} \quad (1)$$

Signals received from individuals will oscillate around a pivot point of zero. Signals greater than zero constitute a buy signal. To allow the system to decide how much to invest on a given trade, the maximum size of a trading signal must be determined. Initially we set the size of the maximum signal as being the size of the first buy signal generated by the system. If a signal is subsequently generated that is stronger than this, the maximum trading signal is reset to the new amount, weaker signals invest a smaller amount in the same ratio as their signal to the maximum. If the sum to be invested is greater than the cash available, the model will invest the cash available less the costs of the transaction. Upon receipt of a sell signal (one less than zero) all positions are closed.

B. Return Calculation

The total return in this model is a combination of its generated return from market activity and the interest gained from the risk free cash position. In this case the interest rates used were the actual US Deposit Interest rates for the same period as the S&P 500 data set, affording an accurate reflection in the cash position. Transaction costs in this model are based on the cost structure used by online trading houses, where flat fees are incurred for the opening and closing of positions, \$10 fees were charged upon entry and exit. When the transaction costs represent a certain percentage of the sum to be invested, the investment becomes unfeasible. Therefore the model is made aware of the transaction costs, when the entry and exit costs arise to 20% of the sum to be invested the model will not take the position, effectively ruling out very small investments arising from weak trading signals. This represents the Do Nothing stance, the model will hold all current positions. The average monthly return over the training period was chosen as the fitness measure.

IV. Results

The system was first applied to the S&P 500 for the period January 1st 1991 to December 1st 1997 for two experimental setups, the better of which was then applied to the Nikkei 225 for the period December 10th 1992 to December 3rd 1997, with the first 440 days of each index comprising of the initial training period including a 75 day buffer at the beginning to allow for technical indicator calculations, and 364 days being the size of the training window for subsequent increments through the time series. A population size 500 individuals was used with 100 generations of training for the initial period and 10 generations for each increment in the training window. A generational rank replacement strategy was used with 25% of the weakest performing members of the population being replaced with newly generated individuals on each generation with 30 runs conducted for each setting with a crossover rate of 0.9 and a mutation rate of 0.1 as in [6].

This section is broken up into two parts. The first, IV-A, reports the returns of each setup over its trading range for the final generation of each trading window increment versus the returns made by the index over the same period in training. The second section IV-B reports the returns made during live trading versus the indices. Figures 2 and 3 display the index time series for the S&P 500 and the Nikkei 225 under investigation.

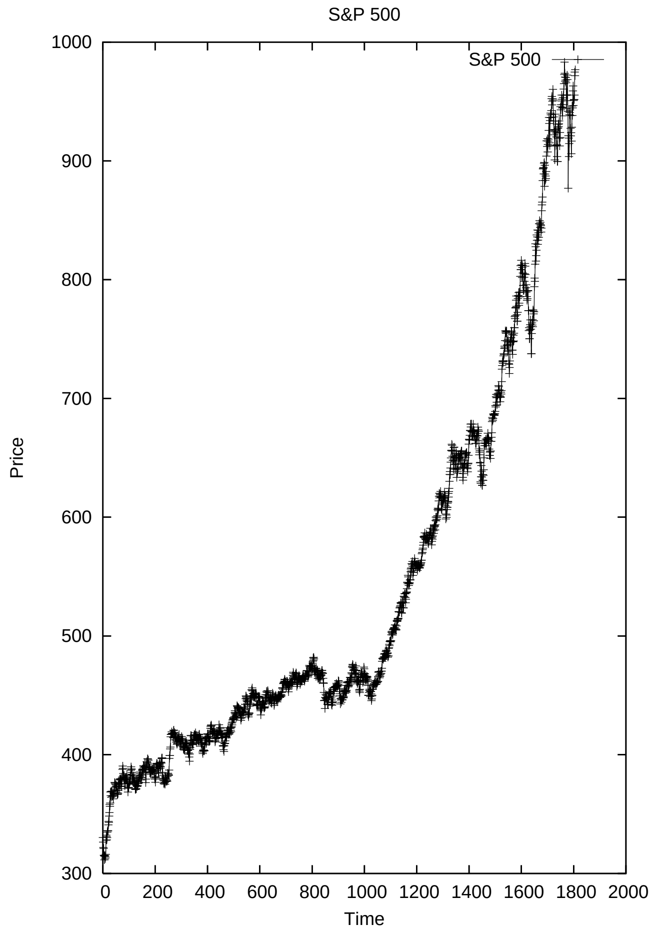

Fig. 2. The S&P 500 Index 1/1/1991 to 3/12/1997.

A. Live Training Returns

For the adaptive setup upon each increment in the training window, the system's population is retrained to incorporate the new data for 10 generations. On the final generation of each live training period the average monthly return of the best performer, i.e. its fitness over the training window, was taken for each increment in the window, this led to the sampling of 273 points for the S&P 500 and 171 for the Nikkei 225. These points represent the average return of the best rule on the final generation for each window increment. The result is a pair of graphs, figures 4 and 5, reporting the avg best discovered monthly return, which can be compared against their respective indices.

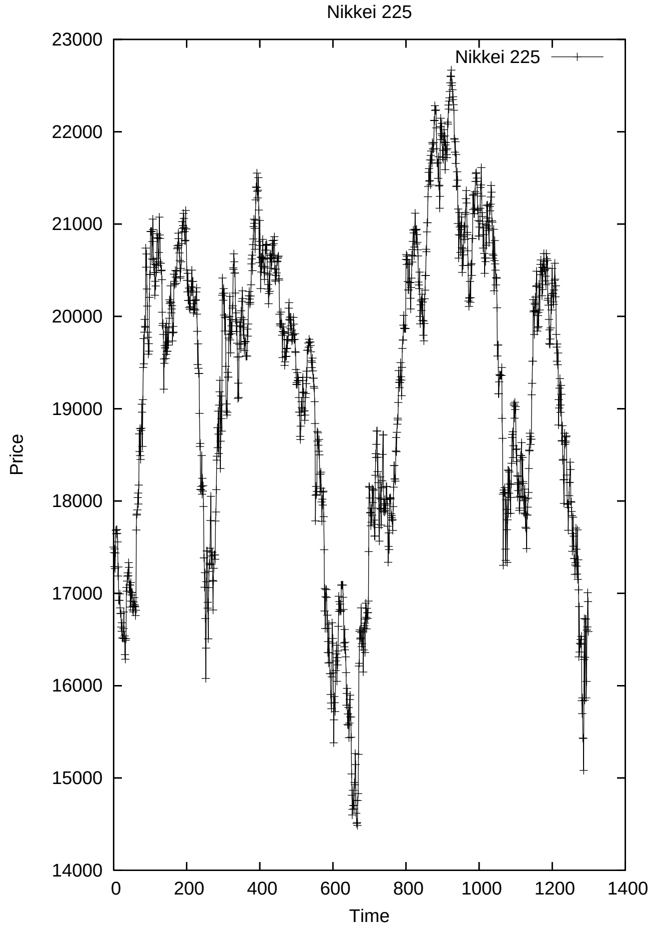

Fig. 3. The Nikkei 225 Index 10/12/1992 to 3/12/1997.

B. Live Trading Returns

Table I displays the results and a break down of the trading statistics for the trading conducted on the S&P 500 for the period 8/9/1992 to 3/12/1997. The adaptive populations are seen to perform well against the index with five out of the 30 runs outperforming the benchmark buy and hold strategy and an average return of 11% less than the index, a remarkable statistic given aggressive fee schedule and strongly up trending price index (fig. 2). In examining the restart population, no run outperformed the index with an average best return of 97% and its best performing run not equalling the average return of the adaptive system. The restart setup is also seen to execute a significantly larger number of trades over the course of a run on average, leading to the much higher cost of trading of \$1580 versus \$502 for the adaptive approach.

The Sharpe ratio [20] is a commonly used ratio for evaluating the risk adjusted returns of an investment where the excess returns of the investment are divided by the volatility

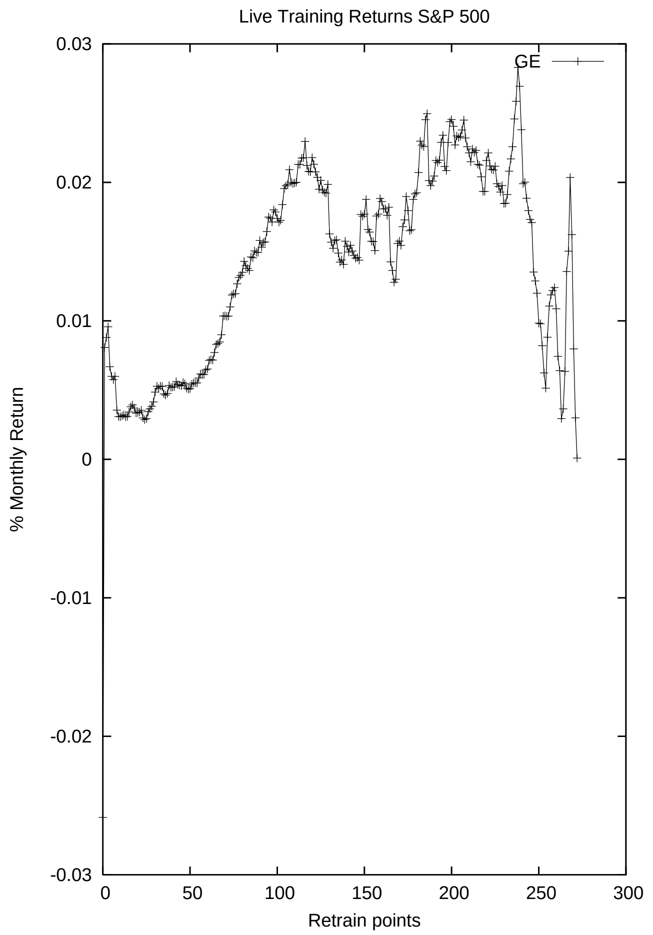

Fig. 4. Monthly return of the system at the end each retraining period for the S&P 500.

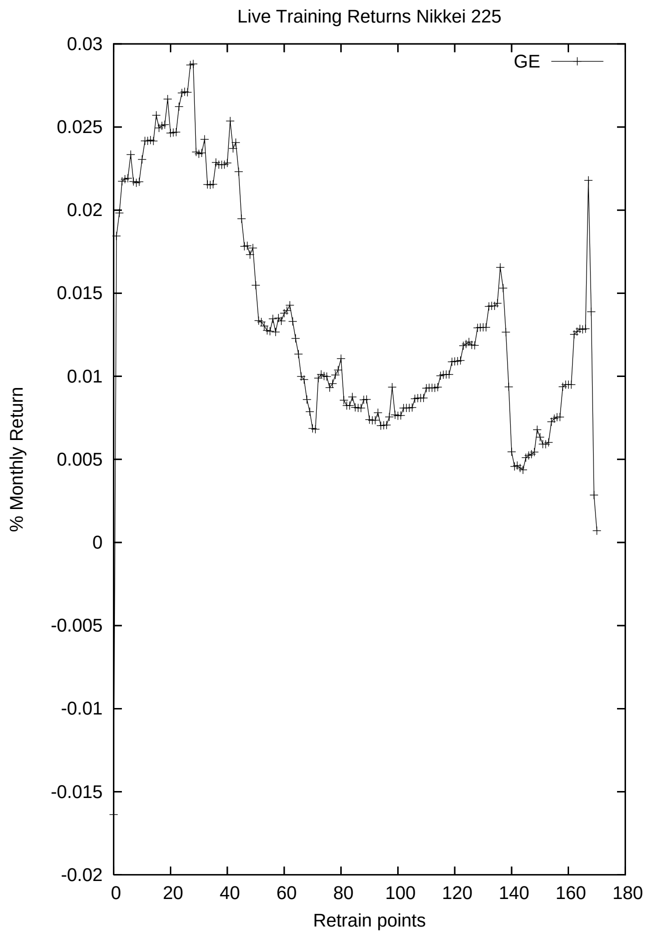

Fig. 5. Monthly return of the system at the end each retraining period for the Nikkei 225.

of the returns. Here this was calculated as the monthly rate of return less interest divided by the standard deviation of the returns. In this statistic we see that the adaptive setup again out performs the restart performance.

In the final two statistics the restart setup is seen to spend more time in the market with 1104 days in the market versus 1007 for the adaptive system. Given the strong upward trend of the S&P 500 over this period this should give it an advantage however when we combine this with the fact that it conducted over 300% more trades it can be concluded that the rules generated by the restart system are of a much more sensitive nature and may suffer from over fitting of the training data.

This point is empasised in the figures 6 and 7 which display the average cumulative fees paid by the adaptive and restart setups along with the average monthly return respectively. In fig. 6 over the first twenty four months the restart setup is seen to conduct significantly more trades, this period also coincides with the least definitively trending area of the SP 500 as seen in fig. 2 for the period 440 to 900 on the graph. For the same period in the monthly returns in fig. 7 the restart system is seen to actually make a loss in its trading activity while the adaptive approach maintains a positive to even

TABLE I

TRADING STATISTICS FOR EACH SETUP ON THE S&P 500.

	Adapt	**Restart**
Index(%)	133	133
Best(%)	138	121
Avg. Best%	122	97
Avg. No. Trades	46	144
Avg. Profitable Trades(%)	76	71
Avg. Losing Trades(%)	24	29
Avg. Fees($)	502	1580
Avg. Sharpe Ratio	22.56	17.32
Avg. No Days In	1007	1104
Avg. No Days Out	359	262

return. After point 1000 in fig. 2 the SP 500 is begins a steady strong upward trend for a long period of the data set. This correlates to point 27 in figures 6 and 7 where both methods monthly returns begin to reflect the same increases, and the fees, for a while, remain static as both systems maintain long positions. However when changes in the SP 500 trend occur towards the end restart system conducts new trading activity, driving its average fees higher over the adaptive system which continues to maintain its long positions thus flatlining its average fees.

Considering the strengths of the adaptive setup the system

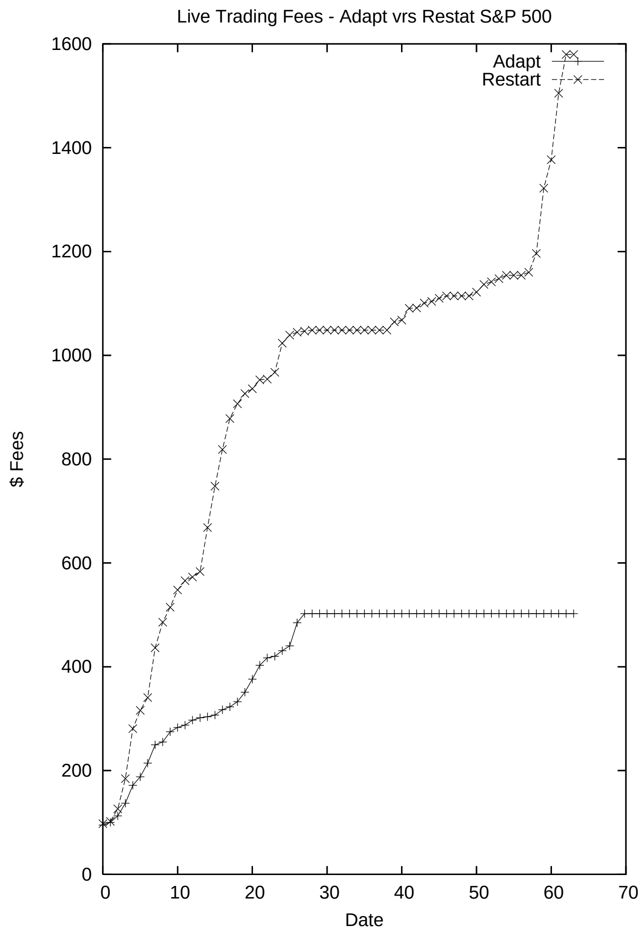

Fig. 6. The average cummulative fees paid.

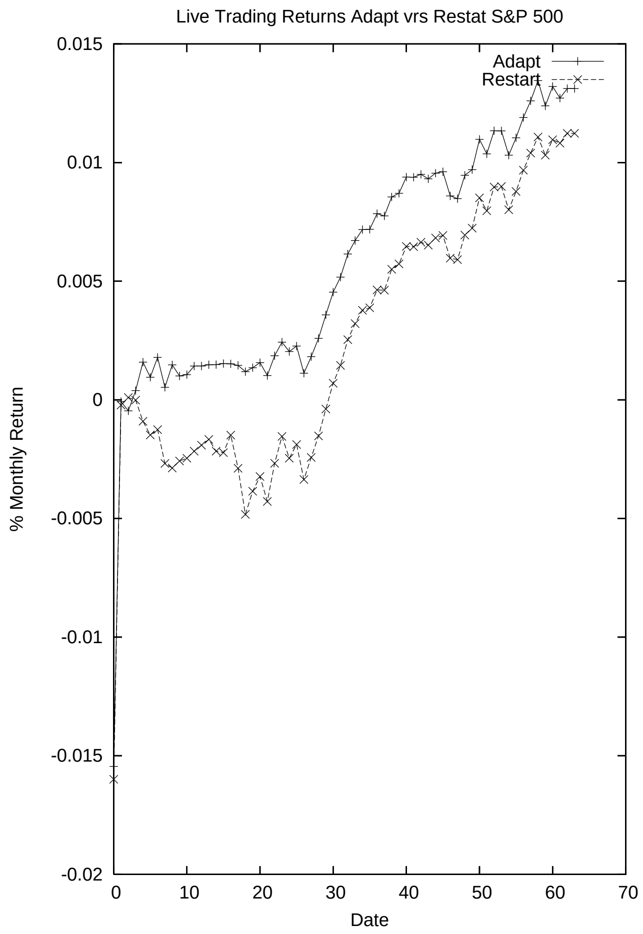

Fig. 7. The average monthly return.

was applied against the Nikkei 225 for the period 18/8/1994 to 3/12/1997 which exhibits a vastly different behaviour in comparison to the SP 500 as seen in figures 2 and 3. Table II displays the live trading statistics.

TABLE II
TRADING STATISTICS FOR THE ADAPTIVE APPROACH ON THE NIKKEI 225.

	Adapt
Index(%)	-19
Best(%)	65
Avg. Best%	55
Avg. No. Trades	75
Avg. Profitable Trades(%)	80
Avg. Losing Trades(%)	20
Avg. Fees($)	815
Avg. Sharpe Ratio	12.83
Avg. No Days In	246
Avg. No Days Out	613

Key to these statistics is the system's performance relative to the benchmark buy and hold, where the index over the same period closes down 19% the system on average yields a return of 55%, indicating that it took advantage of the index's wide volatility. Evidence of this is also seen in the number of trades executed at 75 suggesting the presence of more buy and sell opportunities when compared to the S&P 500. Fig. 8 displays the monthly percentage return over the trading period.

The Sharpe ratio in this case is again strong at 12.83 yet is significantly lower than the ratio attained in the S&P 500 which is a function of the risk involved in the Nikkei despite the system outperforming the index on average.

In this case the number of days spent in the market is only 246 virus the 613 days spent in a cash position. This again highlights how the system took advantage of the index's volatility where it on average made a 55% return on the brief up trend cycles over the trading period.

V. CONCLUSIONS & FUTURE WORK

This study has examined the performance of an evolutionary approach to uncovering technical trading rules. Two methods were explored; one using a single adaptive population across the time series and the other which simply generated a new population of rules for each window increment. The population restart method was clearly seen to suffer from the generation of overly sensitive rules and potential overfitting to the training set.

The key feature of this study is that it demonstrated the ability of Grammatical Evolution to uncover trading rules

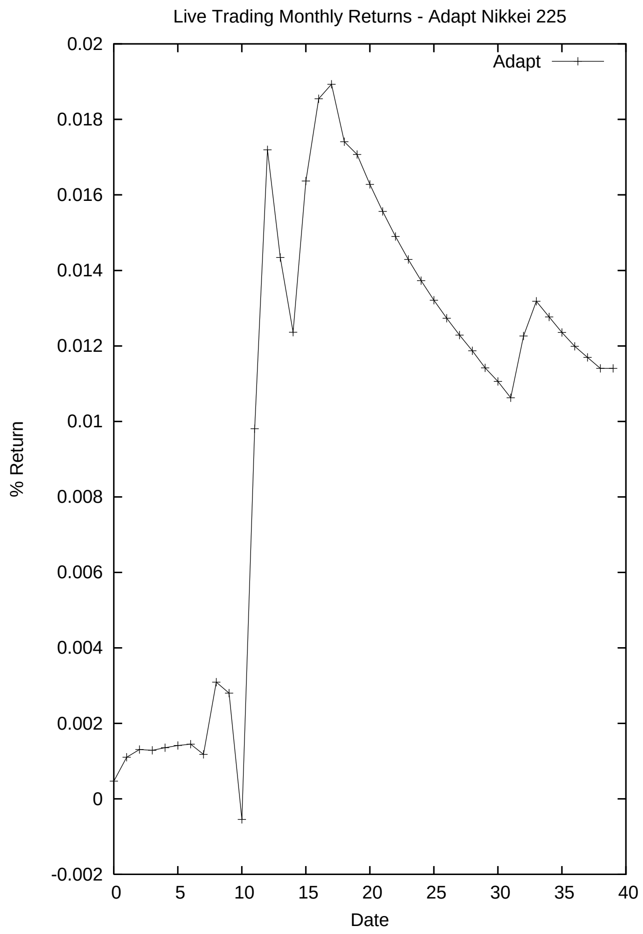

Fig. 8. The average monthly return.

which outperformed their benchmark indices. For the S&P 500 very little opportunity existed to take advantage of downturns in order to sell and reinvest at a lower price yet 5 runs produced returns superior to the benchmark and on average the system produced a comparable return of 122% compared to 133% return produced by the index. Where the Nikkei 225 was concerned the system was able to take full advantage of swings in the market generating returns on average 74% better than the index.

Future work in this area should focus on the relationship between the memory and performance of an adaptive population versus the number of generations evolved at each increment and the size of each increment along with different sizes of trading window with care taken to avoid data snooping.

REFERENCES

[1] O'Neill, M., Brabazon, A., Ryan, C., and Collins, J. (2001a). Developing a market timing system using grammatical evolution. In Spector, L., Goodman, E. D., Wu, A., Langdon, W. B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M. H., and Burke, E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 1375–1381, San Francisco, California, USA. Morgan Kaufmann.

[2] O'Neill, M., Brabazon, A., Ryan, C., and Collins, J. J. (2001b). Evolving market index trading rules using grammatical evolution. In Boers, E. J. W., Cagnoni, S., Gottlieb, J., Hart, E., Lanzi, P. L., Raidl, G. R., Smith, R. E., and Tijink, H., editors, *Applications of Evolutionary Computing*, volume 2037 of *LNCS*, pages 343–352, Lake Como, Italy. Springer-Verlag.

[3] O'Neill, M., Brabazon, A., and Ryan, C. (2002). Forecasting market indices using evolutionary automatic programming: A case study. In Chen, S.-H., editor, *Genetic Algorithms and Genetic Programming in Economics and Finance*. Kluwer Academic Publishers.

[4] Brabazon, A. and O'Neill, M. (2002). Trading foreign exchange markets using evolutionary automatic programming. In Barry, A. M., editor, *GECCO 2002: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference*, pages 133–136, New York. AAAI.

[5] Brabazon, A. and O'Neill, M. (2004). Evolving Technical Trading Rules for Spot Foreign-Exchange Markets Using Grammatical Evolution. *Computational Management Science*. Springer, 2004.

[6] Dempsey, I., O'Neill, M. and Brabazon, T. (2002). Investigations into Market Index Trading Models Using Evolutionary Automatic Programming, In *LNAI 2464, Proceedings of the 13th Irish Conference in Artificial Intelligence and Cognitive Science*, pp. 165-170, edited by M. O'Neill, R. Sutcliffe, C. Ryan, M. Eaton and N. Griffith, Berlin: Springer-Verlag.

[7] Dempsey, I., O'Neill, M. and Brabazon, T., Live Trading with Grammatical Evolution. In Grammatical Evolution Workshop GEWS at the Genetic and Evolutionary Computation Conference, GECCO 04.

[8] Imamura, K., Heckendorn, R.B., Soule, T., Foster, J.A. (2002). N-Version Genetic Programming via Fault Masking. In *LNCS 2278, Proceedings of EuroGP 2002*, Kinsale, Ireland, pp. 172-181. Springer 2002.

[9] Burke, E., Gustafson, S., Kendall, G. A Survey and Analysis of Diversity Measures in Genetic Programming. In *Proceedings of GECCO 2002, the Genetic and Evolutionary Computation Conference*, New York, USA, pp. 716-723, Morgan Kaufmann.

[10] Koza, J.R. (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press.

[11] O'Neill, M., Ryan, C. (2003). *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*. Kluwer Academic Publishers.

[12] O'Neill, M. (2001). *Automatic Programming in an Arbitrary Language: Evolving Programs in Grammatical Evolution*. PhD thesis, University of Limerick, 2001.

[13] O'Neill, M., Ryan, C. (2001) Grammatical Evolution, *IEEE Trans. Evolutionary Computation*, 5(4):349-358, 2001.

[14] 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*, 83-95, Springer-Verlag.

[15] O'Neill, M., Ryan, C., Keijzer M., Cattolico M. (2003). Crossover in Grammatical Evolution. *Genetic Programming and Evolvable Machines*, Vol. 4 No. 1. Kluwer Academic Publishers, 2003.

[16] Koza, J.R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press.

[17] 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*. Morgan Kaufmann.

[18] Koza, J.R., Andre, D., Bennett III, F.H., Keane, M. (1999). *Genetic Programming 3: Darwinian Invention and Problem Solving*. Morgan Kaufmann.

[19] Koza, J.R., Keane, M., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G. (2003). *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers.

[20] Sharpe, W. F., *The Sharpe Ratio* The Journal of Portfolio Management, Fall 1994. Institutional Investor Inc.