The Effect of Varying Parameters on Performance for Adaptive Agents in Technical Equity Market Trading

Cyril Schoreels
ASAP Research Group, School of Computer Science and IT
University of Nottingham
Nottingham, NG8 1BB
United Kingdom
czs@cs.nott.ac.uk

Jonathan M. Garibaldi
ASAP Research Group, School of Computer Science and IT
University of Nottingham
Nottingham, NG8 1BB
United Kingdom
jmg@cs.nott.ac.uk

Abstract – **This paper investigates the impact of varying the quantity of data and the number of generations run for every re-training step in an adaptive trading system. Using historical equity data, populations of agents are continuously retrained and assessed based on their performance across an out-of-sample data set. Comparison was performed using three test sets, for which each had one variable altered for every run. Results showed significant differences in performance when varying the number of trading days, while no difference was found when varying generations.**

I. INTRODUCTION

There exist a variety of arguments in favour of adopting an adaptive approach to evolving trading strategies. Leiviskä for instance argued that any system must be capable of indefinitely continuing its training to enable adaptation, and must not be frozen after an initial training period [1]. To accomplish this, he argued that a system must be able to detect changes in the system, to which it can then respond by changing itself and therefore adapting to the change. Previous to this Trojanowski and Michalewicz also suggested that change could be initiated through deduction, where a change in the environment is noted by a fall in performance [2]. The obvious advantage offered by such a system would be its ability to continue operations autonomously responding whenever a change in the environment is noted, without intervention being required by a human user. As a static system lacks the ability to continuously re-optimise its own parameters, sub-optimal performance will form an inherent disadvantage [3].

Angelov et al for example discussed an adaptive system in the context of fuzzy systems and neural networks [4]. In their discussion they argued that adaptive systems ought to possess the properties of evolving an adaptation mechanism, accumulating experience, autonomy, intelligence and the ability to deal with unexpected input in addition to being able to evolve their own structure. The use of adaptive systems in other research areas has shown positive results [5, 6, 7].

Pi and Rögnvaldsson provide a good example where an adaptive approach was found to be beneficial [8]. In their paper a futures trading system using a neural network was presented, where their model was retrained with every additional piece of data that became available over time. They found this helped them overcome the problem of continuously changing conditions present in financial data series as well as made more efficient use of the available historical data.

The work presented here employs a genetic algorithm to evolve the trading characteristics of agents in an equity market. Similar work was first published by Allen and Karjalainen [9], where they also used a genetic algorithm to evolve a set of trading strategies. When transaction costs were included however, their system did not earn excess returns compared to a buy-and-hold strategy. Other authors have since argued that despite including transaction costs, similar performance to the buy-and-hold strategy can be produced [10, 11, 12].

Following such positive results in automated trading and evolution of strategies, some research has placed greater emphasis on comparison of approaches and strategies. The Penn-Lehman Automated Trading project for instance offers a platform for different strategies to compete on equal terms and allows a ranking based on performance among different strategies entered [13]. Initial work has shown promising results with a majority of clients showing positive earnings.

This paper will start by introducing the technical aspects of the implementation used, followed by a description of the adaptive system employed and variables altered for comparison. It concludes with a statistical analysis of the results and a brief discussion of their implication in this and future research.

II. AGENT DESIGN

The adaptive system used in this investigation is based on continuous retraining of an agent population, with each agent being representative of a real-life trader in an equity market, as discussed in depth below.

A. Definition of a Trader

Every agent represents an individual trader with a personal portfolio and capital holdings, collectively referred to as their total assets or worth. With a fixed starting capital, at the end of every trading day each trader uses historical price data to make a decision on every security whether to buy, sell or hold. No limitations, apart from capital constraints, are set on the number of trades conducted.

Every agent uses technical indicators to generate trading signals for every security that is assessed. Indicators used are the Simple Moving Average (MA), Relative Strength Index (RSI), Price-Rate-of-Change (ROC), Stochastic Oscillator (SO), Moving Average Convergence Divergence (MACD) and Bollinger Bands (BB). Depending on the signals returned, an overall decision of buy, sell or hold is made for a security depending on the agent's decision type. The amount of capital invested is again determined by the genome, with it equally allocated between each security flagged for acquisition. Securities that are sold are converted into capital at their current closing price.

Parameters used in the agent's analysis and decision process are determined by the agent's genome, effectively determining its trading behaviour. The genome for every agent consists of a string of integers taking values 1-10, of length 28 with cardinality 10. The exceptions to this were genes 1, 2, 21 and 28, defining decision types with a cardinality of 4, the risk averseness factor with a cardinality of 2 as well as the SO D variable and BB deviation variable with a cardinality of 5. All genes are defined as in Table I.

TABLE I

GENE DESCRIPTIONS

Gene	Range	Description/Function
G_1	1-4	Decision type
G_2	1-2	Risk averseness factor
G_3	1-10	Capital investment proportion
G_4	1-10	Moving Average weight
G_5	1-10	RSI weight
G_6	1-10	Short-term ROC weight
G_7	1-10	Long-term Price ROC weight
G_8	1-10	SO interpretation 1 weight
G_9	1-10	SO interpretation 2 weight
G_{10}	1-10	MACD weight
G_{11}	1-10	BB weight
G_{12}	1-10	MA short-term value
G_{13}	1-10	MA long-term value
G_{14}	1-10	RSI time period
G_{15}	1-10	RSI buy threshold
G_{16}	1-10	RSI sell threshold
G_{17}	1-10	ROC level
G_{18}	1-10	ROC short-term value
G_{19}	1-10	ROC long-term value
G_{20}	1-10	SO K variable value
G_{21}	1-5	SO D variable value
G_{22}	1-10	SO buy threshold
G_{23}	1-10	SO sell threshold
G_{24}	1-10	MACD short-term value
G_{25}	1-10	MACD long-term value
G_{26}	1-10	MACD signal line
G_{27}	1-10	BB time period value
G_{28}	1-5	BB deviations number

Neither transaction costs nor interest on held capital were included, and the environment is assumed discrete and deterministic in a liquid market, meaning that an agent's actions cannot affect prices.

B. Technical Indicators

Technical indicators are tools used in the technical analysis of financial markets, exploiting the existence of trends to determine potential buy, sell or hold conditions. Indicators are mathematical formulae, commonly based on closing price or volume data, with price information being used exclusively in this system. Though markets are often argued to move randomly [14], regularities do appear and lead to observed phenomena such as seasonal cycles for example [15], which are exploited by the indicators.

The following is a brief description of the indicators employed, and how an agent's genome is used to individualize the calculations. The following descriptions are primarily based on a summary presented by Achelis [16], and also include the translation of a number of gene values. When an agent is initialised, some of its gene values would not be appropriate for direct use in the system and need to be modified. For instance, though G_{17} can be used directly in the technical indicator without translation, this is not possible for G_{13}. As an informal rule, the translation was based on achieving a representative average value approximate to the range used in wider literature, rounded to the next natural number.

A MA shows the average value of a securities price over time. The short-term and long-term moving average values are calculated as:

$$MA(t,N) = \frac{\sum_{i=t-N}^{t} price_i}{N}, \tag{1}$$

where $N=4G_{12}$ for the short-term and $N=(5G_{13})+50$ for the long term and t is the current trading day.

The RSI is a price-following oscillator that compares the internal strength of a single security to determine the current trend. It is calculated as:

$$RSI(t,N) = \frac{100}{1+\frac{avgDowns}{avgUps}}, \tag{2}$$

where $N=2.5G_{14}$ and avgUps/avgDowns is the average of the increase/decrease in price noted for every day it closed higher/lower over the last N days from day t.

The ROC indicator is based on the assumption of cyclical price movements, and considers the relative change of prices over time to indicate trends. It is calculated by:

$$ROC(t,N) = price_t - price_{t-N}, \tag{3}$$

where $N=2G_{18}$ for the short-term and $N=4G_{19}$ in the long-term.

The SO compares a security's price relative to its price range over a given time period, using two parameters commonly defined as K and D. K is calculated in (4) over $1.5G_{20}$ days:

$$K(t,N) = 100\left(\frac{price_t - lowestClose}{highestClose - lowestClose}\right), \tag{4}$$

and where D(t,N) is a moving average of K over G_{21} days.

The MACD "is a trend following momentum indicator that shows the relationship between two moving averages of prices" [16]. The MACD is calculated:

$$exMA(t,N) = \frac{2\,price_t + prevMA(N-1)}{N+1}, \quad (5)$$

$$MACD(t,N,O) = exMA(t,N) - exMA(t,O), \quad (6)$$

where N=$2G_{24}$, O=$4.5G_{25}$ and prevMA is the previous exponential moving average apart from the first instance, where a simple moving average is used.

Lastly, BBs are generally used to provide a form of guideline, indicating possible trend reversals. The upper and lower bands are calculated as:

$$BBstdDev(t,N) = G_{28}\sum_{i=t-N}^{t}\left(price_i - MA(i,N)\right)^2, \quad (7)$$

$$upperBand(t,N) = MA(t,N) + BBstdDev(t,N), \quad (8)$$

$$lowerBand(t,N) = MA(t,N) - BBstdDev(t,N), \quad (9)$$

where N=$3.5G_{27}$.

C. The Decision Process and Decision Types

There are essentially three steps that every agent follows to determine whether or not to add a particular security to its acquisition or sale list. To analyse a security it first performs the calculations described above, each using historical closing price information. Depending on the indicator, one or more values are then returned which then need to be interpreted for the calculated values to gain meaning. Based on this interpretation, every agent then compiles a list of buy and sell signals for every security as the second step in this process. The final step is generating an overall buy, sell or hold decision for each security, based on the agent's decision type. The following descriptions provide the decision process and interpretation for each indicator. For notational convenience, the parameter t was dropped in the code below.

If the MA over the short-term is larger than over the long-term, it indicates an upward trend and a buy signal would be generated or vice versa. Additionally, the short-term moving average can be compared to the current price of the security, which if greater, would indicate a downward trend and hence a sell signal should be generated. In this implementation, if the agent is risk averse, as determined by G_2, it bases its interpretation on a logical AND between those two interpretations and is therefore more reluctant to generate a buy signal. On the other hand, if the agent is risk taking, a logical OR is used and either interpretation suggesting a buy would suffice for the agent to consider this a buy signal. In pseudo-code:

```
BOOLEAN A = MA(4G12) < current price
BOOLEAN B = MA((5G13)+50) < MA(4G12)
BOOLEAN C = (G2 == 1)
IF ( A AND B AND C ) OR (NOT C AND ( A OR B )
    Action: buy
ELSE
    Action: sell
ENDIF
```

For the RSI, if the calculated value lies above or below the sell or buy threshold respectively, the appropriate signal will be generated.

```
IF RSI(2.5G14) >= (4G16)+50
    Action: sell
ELSEIF RSI(2.5G14) <= 5G15
    Action: buy
ELSE
    Action: hold
ENDIF
```

The ROC is repeated for both long- and short-term analysis as, where if the calculated value lies below the negative threshold value it indicates a buy, while a calculated value above the positive threshold value indicates a sell in both instances. The following example is for the short-term, with ROC($2G_{18}$) being replaced by ROC($4G_{19}$) for the long-term.

```
IF ROC(2G18) < -G17
    Action: buy
ELSEIF ROC(2G18) > G17
    Action: sell
ENDIF
```

For the SO multiple interpretations are possible, though the following two are used in this instance. First, it can be considered a buy signal if the K value is larger than the D value or vice versa. Second, threshold values can be used for both K and D. In that case, if K and/or D is smaller than the buy threshold a buy signal is generated, or equally, if K and/or D is larger than the sell threshold a sell signal is generated.

```
IF K(1.5G20) > D(G21)
    Action: buy
ELSE
    Action: sell
ENDIF

BOOLEAN A = K(1.5G20) < 3.5G22
BOOLEAN B = D(G21) < 3.5G22
BOOLEAN C = K(1.5G20) > (4G23)+50
BOOLEAN D = D(G21) > (4G23)+50
IF  ( G2 & A & B ) | ( !G2 & (A | B) )
    Action: buy
ELSEIF ( G2 & C & D ) | ( !G2 & (C | D) )
    Action: sell
ENDIF
```

The MACD compares its calculated value to a moving average of itself over a time period, whereby a buy signal is generated if the moving average is smaller, and a sell signal if the moving average is larger.

```
IF MACD(2G24,4.5G25) >
        MA(1.5G26,MACD(2G24, 4.5G25))
    Action: buy
ELSE
    Action: sell
ENDIF
```

BBs indicate that when the current price breaks through the lower Bollinger Band it is considered a buy signal, while if it breaks through the upper band it is considered a sell signal.

IF lowerBand(3.5G_{27}) >= current price
  Action: buy
ELSEIF upperBand(3.5G_{27}) <= current price
  Action: sell
ENDIF

In order to allow for different approaches that exist among real traders to selecting securities for purchase or sale, four agent decision types were implemented. Decision type 1 performs a simple comparison between the number of buy and sell signals, taking the appropriate action if one is greater than the other for any particular security. For instance, out of the 8 possible signals used here for a security, if 3 are buy and 2 are sell, an agent of this type would want to purchase this security. Decision type 2 follows the same principle as decision type 1. However it also stipulates that for a buy or sell action to occur, at least half of all signals must be in favour. In this instance for example, out of 8 possible signals, if only 3 are buy signals even though no sell signals exist, no action will be taken as it failed to reach the minimum buy threshold. Decision type 3 sums the indicators by taking buy signals as +1 and sell signals as -1, as well as including a weighting process on each signal (G_4 to G_{11}), increasing or decreasing its impact on the final sum. Therefore a positive sum would translate into an overall buy signal, while a negative sum into an overall sell signal. Decision type 4 follows the same principle as decision type 3, except for adjusting the final sum to create a threshold value which it needs to exceed prior to resulting in an overall buy or sell decision. For example, though decision type 3 would generate a buy signal for a value of +5, the threshold for decision type 4 is set at +/-10 and would therefore result in a neutral hold signal.

III. ADAPTIVE SYSTEM SPECIFICATION

To assess the impact of altering the number of generations and number of trading days used during each retraining step, the following system set-up was used.

A. Historical Market Data

The system uses historical financial data taken from the DAX-30, which is an index listing the top 30 capital weighted companies registered in the German market, with various weighting factors applied to each listed company to determine their impact on the Index. Data used in the system covers the time from 01.01.1996 to 31.12.2002, however it does not include all securities from the DAX for its entire span. Due to changes in the constituents of the Index, daily closing price information was only available for 20 securities over the desired time span. Those included are listed in Table II using their Wertpapierkennummer (German stock market security identification number) below.

Furthermore, it is also important to note that 100 days of historical data prior to the first trading day are required to perform technical analysis. This is due to some indicators using large amounts of historical data in their calculations. For this reason a year of data preceding the test data set was kept for training, to allow performance monitored trading to commence on the 1st of the year. Test data therefore covered 1826 trading days representing the time period 01.01.1997 to 31.12.2002. This period was chosen as it exhibited significant movement, including the boom and crash periods around the millennium, but overall only resulted in the Index gaining 0.14%. It therefore considers a good range of trading environments within the trading period while overall essentially representing little change, allowing for effective comparison.

TABLE II

SECURITIES USED IN SYSTEM

840400	648300	519000	717200	515100
760080	823212	803200	723610	575200
802200	703712	761440	766400	695200
593700	843002	750000	543900	514000

B. Genetic Algorithm

A general representation of how the agent population is evolved is shown in Fig. 1 below. The elite population and killed population are each respectively the top and bottom 25% of the entire agent population.

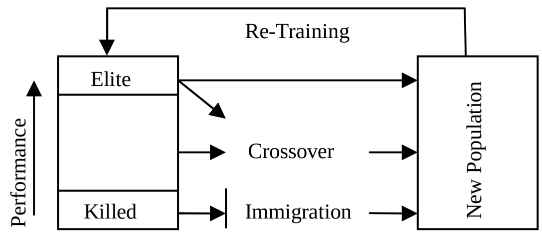

Fig. 1. Evolution of agent populations

For selection, elitism [17] is used, whereby a portion of the most successful agents carries forward unaltered every generation. Immigration [18] was also employed, where a portion of the worst performers are killed off and replaced by a new randomly generated group of agents immigrating into the system, constantly introducing new genetic material. This facilitates greater coverage of the search space, while also avoiding premature convergence and non-exclusion of other possible solutions not present in the original base population's gene pool. Each agent in the mediocre population, those not killed or part of the elite, randomly selects another agent from the mediocre and elite population and uses two-point crossover to create an offspring. This offspring then replaces the parent from the mediocre population. Two randomly selected agents of the same decision type can mate, with a random part of the first agent's genome being replaced by the equivalent section from the second agents genome, forming a new genome combination. One restriction in this process is that there exists a 25% chance of an agent mating with an agent of a different decision type, as for instance weighting genes will not have been relevant to types 1 or 2 previously.

In other research, performance tends to generally be measured as an agent's capital and value of all holdings at the end of the trading period. However, as this biases results based on the cut-off date used, a more overall and therefore more representative picture of performance throughout the entire testing period can be obtained if the area under an agent's total asset graph is considered as its fitness. For this investigation, we therefore propose that performance refers to the area under an agent's total asset graph for the trading period being assessed.

C. System Description

To start the system with a trained population, 163 days of data preceding the out-of-sample test data set were used as an initial training phase, run over 1000 generations of agents. Thereafter the agent population was retrained after every trading day for Y generations using the most recent X trading day's closing price information. Y and X representing the parameters that are altered in every experiment, as shown in Table III.

A trading population (TP) is first established using randomly generated genomes. The size of the TP was equivalent to that of the elite in retraining. For the initial training phase and for every retraining phase during the experiment, a second population is created referred to as the evolution population (EP). The EP is based on the TP, with a replica of the TP making up 25% of the EP, a further 25% are new randomly generated agents and the remaining 50% are offspring from the TP using two-point crossover. The EP then evolves using the GA described above and, once finished, updates the TP and is then discarded. It is important to emphasize however that the TP does not necessarily change from trading day to trading day, but less successful agents in the TP have their genome replaced by the genome of more successful agents that have emerged from the EP. This way holdings and capital remain continuous in the TP. This could be loosely based on an analogy where a group of traders who manage their own individual portfolio in a company, visit a school for a week, then later return to their job with a possibly new perspective on how to trade but with the same holdings and capital position. In other words, the phenotype of an agent might be changed during the update process. Once the TP is updated, the EP is removed and only recreated at the end of the next trading day to restart the whole process. A representation of this process is shown in Fig. 2.

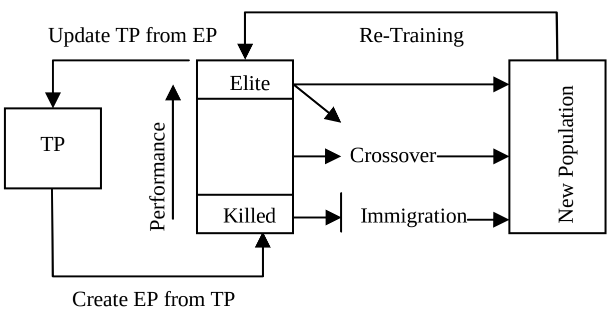

Fig. 2. Adaptive evolution

Fitness being defined as the total area under their total asset graph over the retraining duration.

IV. RESULTS AND ANALYSIS

Each experiment was repeated five times, with the average and standard deviation of those results shown in Table III. For convenience, results from experiment A were repeated in the table. The large standard deviation in Experiment B was due to one result set having returned a profit of 377.07% and an AUC of 519x10^{6}. When removed and averaged across only 4 results, profit would have been 44.57% with a standard deviation of 27.69% and AUC would have been 245x10^{6} with a standard deviation of 28.0x10^{6} for experiment B, far more inline with the other results observed.

TABLE III

EXPERIMENT'S PARAMETERS AND RESULTS

	Y	X	Profit (%) Mean (S.D.)	AUC (x10^{6}) Mean (S.D.)
A	50	50	95.10 (38.79)	284 (17.7)
B	10	250	111.07 (150.62)	300 (125.1)
C	250	10	48.10 (9.36)	261 (8.2)
A	50	50	95.10 (38.79)	284 (17.7)
D	50	25	12.25 (27.87)	230 (19.1)
E	50	100	21.74 (5.11)	263 (25.1)
A	50	50	95.10 (38.79)	284 (17.7)
F	25	50	62.15 (36.08)	277 (36.0)
G	100	50	46.57 (36.60)	264 (24.2)

Comparison between experiments A-F-G aimed to determine whether changing the number of generations used in retraining affected performance with a constant number of trading days. Comparison between A-D-E assessed whether changing the number of trading days used in retraining affected performance with a constant number of generations used. Experiments A-B-C maintained a constant number of days across the training phase (e.g. 50 trading days over 50 generations = 2500 days or 250 trading days over 10 generations = 2500 days), but varied in trading days and generations run. If no difference could be observed, this would indicate that the change in either variable was counterbalanced by the change in the other. Kruskal-Wallis tests were done on each of those three groups using their AUC for comparison. Results are shown in Table IV.

TABLE IV

KRUSKAL-WALLIS TEST RESULTS

Comparison	H-value	P-value
A-B-C	4.38	>0.102
A-D-E	8.06	0.009
A-F-G	1.82	>0.102

At a significance value of 0.01, it is evident that there exists no statistical difference between A-B-C, implying that the deciding factor in retraining is the total exposure to data, as determined by retraining generations * retraining trading days, rather than either parameter independently. Similarly, for A-F-G it seems that the system adapts after only few generations of retraining, as there exists no significant difference between the experiments either. A-D-E however showed a significant difference, implying that a change in trading days without compensation through additional generations run will affect performance.

Based on these results, it appears essential to optimise the number of trading days in re-training for an adaptive system or risk sub-optimal performance. This could for instance be achieved or regulated by another evolutionary mechanism in the system. On the other hand compared to A-B-C, it is unexpected that when maintaining a constant number of generations, trading days should be shown to have such an effect on performance in A-D-E.

Volatility, measuring the average daily relative change in value, was also investigated, but it was found that there was no noticeable difference.

V. CONCLUSIONS

Results from this set of experiments indicate that varying retraining generations or maintaining an equal total exposure to data has no statistically significant effect on performance. A statistically significant difference was found however when varying the number of retraining trading days. Therefore it is important that the appropriate number of trading days is chosen or optimised, as this has the greatest effect on performance and could severely affect results obtained.

VI. REFERENCES

[1] K. Leiviskä, “EUNITE Roadmap (ver.1.1)”, *European Network on Intelligent Technologies for Smart Adaptive Systems*, 2002.

[2] K. Trojanowski and Z. Michalewicz, "Evolutionary Algorithms for Non-Stationary Environments", *Proc. Intelligent Information Systems VIII*, Poland, 1999.

[3] S.H. Cheng and C.H. Yeh, "Evolving traders and the business school with genetic programming: A new architecture of the agent-based artificial stock market", *J. Economic Dynamics and Control*, 2001, Vol. 25, Iss. 3-4, pp. 363-393.

[4] P.P. Angelov, R.A. Buswell, J.A. Wright and D.L. Loveday, "Evolving Rules-Based Control", *European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adapative Systems*, Aachen, Mainz, 2001, pp. 36-41.

[5] M. Ayob and G. Kendall, “An Investigation of An Adaptive Scheduling Approach for Multi Head Placement Machines”, *Proc. 1st Multidisciplinary Int’l Conf. on Scheduling: Theory and Applications*, Nottingham, UK, 2003, pp. 363-380.

[6] P. Cowling, G. Kendall, L. Han, "An Adaptive Length Chromosome Hyperheuristic Genetic Algorithm for a Trainer Scheduling Problem", *Proc. 4th Asia-Pacific Conf. Simulated Evolution And Learning*, Singapore, 2002, pp. 267-271.

[7] G. Kendall and M. Willdig, "An Investigation of an Adaptive Poker player", *Proc. 14th Australian J.Conf. Artificial Intelligence*, Adelaide, Australia, 2001, pp. 189-200.

[8] H. Pi and T.S. Rögnvaldsson, "A neural network approach to futures trading", *Proc. 3rd Int'l Conf. on Neural Networks in the Capital Markets, Progress in Neural Processing*, Singapore, 1996, pp. 17- 25.

[9] F. Allen and R. Karjalainen, "Using genetic algorithms to find technical trading rules", *J. Financial Economics*, 1999, Vol. 51, No. 2, pp. 245-271.

[10] L.A. Becker and M. Seshadri, "GP-evolved Technical Trading Rules Can Outperform Buy and Hold", *Proc. 6th Int'l Conf. on Computational Intelligence and Natural Computing*, 2003, pp. 26-30.

[11] J-Y. Potvin, P. Soriano and M. Valée, "Generating trading rules on the stock markets with genetic programming", *Computers & Operations Research*, 2004, Vol. 31, pp. 1033-1047.

[12] G. Kendall and Y. Su, "Co-evolution of Successful Trading Strategies in A Simulated Stock Market", *Proc. 2003 Int'l Conf. on Machine Learning and Applications*, Los Angeles, USA, 2003, pp 200-206.

[13] M. Kearns and L. Ortiz, "The Penn-Lehman Automated Trading Project", *IEEE Intelligent Systems*, Nov/Dec 2003.

[14] E.F. Fama, "Efficient capital markets: A review of theory and empirical work", *J. Finance*, 1970, Vol. 25, pp. 383-417.

[15] N.S. Thomaidis and G.D. Dounias, "Behavioural Finance: A New Challenge to Computational Intelligence", *Proc. European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems*, Aachen, Germany, 2004, pp. 79-92.

[16] S.B. Achelis, *Technical Analysis from A to Z*, McGraw-Hill Trade, 2000.

[17] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[18] J. Branke, "Evolutionary Approaches to Dynamic Optimization Problems - A Survey", *GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems*, 1999, 134-137.