A Comparison of Adaptive and Static Agents in Equity Market Trading

Cyril Schoreels, Jonathan M. Garibaldi
School of Computer Science and IT
University of Nottingham
czs@cs.nott.ac.uk

Abstract

This paper aims to determine whether an adaptive agent population performs better than a static population. A static population is evolved on historical equity market data from the DAX-30, split into training and testing segments. An adaptive population is retrained continuously over the most recent available data that becomes available with each passing day. For comparison their performance over the out-of-sample test data is measured. Results obtained indicate a clear superiority of the adaptive over the static approach.

1. Introduction

With increasing processing power and ever more sophisticated prediction models, automated trading systems and strategies for use in financial markets are becoming increasingly popular. However, the ability of automated trading systems to outperform their human equivalent portfolio managers, the market itself or other financial alternatives, such as fixed interest bonds, is still debated. This paper focuses on the use of two general approaches to evolving successful systems, broadly classified into static and adaptive, to determine whether either presents significant advantages over the other.

Allen and Karjalainen [2] first developed a system that used a genetic algorithm to create composite trading rules on which to base trading decisions. They found that, including transaction costs, their rules did not earn excess returns over a buy-and-hold strategy. Following on from this work, Becker and Seshadri [3] used a genetic programming approach and demonstrated that it was possible to outperform buy-and-hold. Similarly, Potvin, Soriano and Vallee [17] argued that their genetic programming based approach offered an effective alternative to the buy-and-hold strategy. Many other papers have been published using various approaches ranging from neural networks to genetic programming and genetic algorithms, for example [6, 9, 10, 11, 14, 18]. However all featured a similar segmentation of their data that could be classified as training and testing sets. This approach is referred to in this paper as ‘static’ evolution.

However, there are multiple inherent problems with using such a static approach. For instance, Pictet et al described the risks of over fitting [16], while similarly, Cheng and Yeh argued that in continuously changing environments an evolved solution might quickly become obsolete during testing [5]. Hence, if the aim of an approach is to obtain the best possible performance, a clear need exists to allow agents to continue evolution indefinitely, as discussed by Leiviskä [12]. He argues that a system must not be frozen after a learning phase but must be capable of adaptation. He continues by discussing the means necessary to achieve this as well as outlining the types of adaptation that exist. However for the purpose of this investigation we will only draw on the argument that systems must not only continue to evolve or adapt, but in doing so will perform better than frozen or static equivalents. Pi and Rögnvaldsson [15] for example continuously retuned their neural network model, minimizing the initial training data period while obtaining a very large out-of sample test. Similarly, Lettau [13] looked at the concept of bounded rationality in agents using a genetic algorithm. Interestingly, though he used a continuous learning approach, he found that agents were still not able to “take rare events into account”.

Rather than mainly assess performance of the presented system compared to the market overall or other strategies such as buy-and-hold, this paper focuses on comparing the suitability of an adaptive approach compared to the more common static approach in the evolution of trading strategies. Genetic algorithms have been used in many applications, such as equity trading, just as the comparison of static and dynamic agent learning has been well studied. This work combines these two to investigate whether adaptive learning is also useful in genetic algorithm based technical trading. First, we discuss the general system setup used for experimentation, prior to describing the approach specific implementation of static and adaptive evolution in detail. This is then followed by a presentation of the results, its analysis and concluding comments.

2. System outline

The static system used in this investigation, on which the adaptive implementation is also based, has been previously described in [19]. However for completeness a detailed description is included in the following section.

2.1. Agent design

Each agent represents an individual trader with a personal portfolio and capital holdings. Using a set starting capital, at the end of every trading day it uses historical price data to make a decision for every security whether to buy, sell or hold. No limitations are set on the number of trades conducted. Parameters used in analysis and all decision processes are determined by the agent's genome.

Every agent uses a series of technical indicators that are used to generate trading signals on every security that is assessed. Depending on the signals returned, an overall decision of buy, sell or hold is made depending on its decision type. The amount of capital to invest at the end of a trading day is again determined by the genome, whereby the sum is equally allocated to each security flagged for acquisition. Securities that are sold are converted into capital at their current closing price.

In order to allow for differing approaches to selecting securities for purchase or sale, four decision types were implemented in the decision process. 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 8 possible signals 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 also stipulates that for a buy or sell action to occur, at least half of all signals must be in favor. 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 positive and sell signals as negative, as well as including a weighting process on each signal, increasing or decreasing its magnitude and hence 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.

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.

2.2. Genetic algorithm

For selection, elitism [8] is used, whereby a portion of the most successful agents carries forward unaltered every generation. Immigration [4] 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. The only 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. The elite population and killed population are each respectively the top and bottom 25% of the entire agent population.

A general representation of how the agent population is evolved is shown in Figure 1 below.

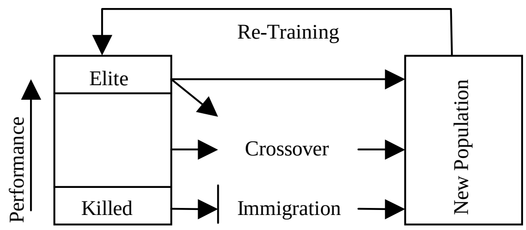

Figure 1. Evolution of agent populations

In wider literature, 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 accurate 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. Therefore for the purposes of this investigation, performance refers to the area under an agent's total asset graph for the trading period being assessed.

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 Stochastic Oscillator D variable and Bollinger Band deviation variable with a cardinality of 5. All genes are defined as in Table 1 below, with abbreviations explained in the following section.

Table 1. 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

2.3. Technical Indicators

Financial technical analysis is "the study of prices to make better investments" [1], where technical indicators represent the tools used for analysis. In other words, indicators are mathematical models used to indicate buy or sell conditions. They are commonly based on closing price or volume data, though in this system price information was used exclusively. They exploit the existence of trends in the market to determine conditions. Though markets are often argued to move randomly [7], others point out that irregularities do appear in markets due to various factors, which lead to observed phenomena such as seasonal cycle for example [20]. The following is a brief description of the indicators used, and how an agent's genome is used to personalize analysis and interpretation of the securities. The following descriptions are primarily based on a summary presented by Achelis [1]. They include calculations and interpretations used in the system itself, including the translation of a number of gene values. When an agent is initialized, 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 without translation, this is not possible for G_{13}. The translation is based on achieving an average value approximate to that suggested in literature (e.g. a common value for the RSI buy threshold is 20-30, therefore $5G_{15}$ was used).

A simple Moving Average (MA) shows the average value of a securities price over time. If the moving average over the short-term is larger than that 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 to generate a buy signal. The MA is calculated as:

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

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

The Relative Strength Index (RSI) is a price-following oscillator that compares the internal strength of a single security. Quite simply, if the calculated value lies above or below the sell or buy threshold respectively, the appropriate signal will be generated. For instance, if the calculated value is 80, a sell signal is generated. The threshold values were $5G_{15}$ for buy and $(4G_{16})+50$ for sell. The RSI 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 Price Rate-of-Change (ROC) indicator is based on the assumption of cyclical price movements, and considers the relative change of prices over time to indicate trends. The same principles apply for long- or short-term analysis, 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. 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 Stochastic Oscillator (SO) compares a security's price relative to its price range over a given time period. 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, where if K and/or D is smaller than the buy threshold a buy signal is generated, or vice versa. Threshold values are $3.5G_{22}$ for buy and $(4G_{23})+50$ for sell. D is a moving average of K over G_{21} days. K is calculated as:

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

where $N=1.5G_{20}$.

The Moving Average Convergence Divergence (MACD) is a "trend following momentum indicator that shows the relationship between two moving averages of prices" [1]. It 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. 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. The signal line is translated as $1.5G_{26}$.

Lastly, Bollinger Bands (BB) are generally used to provide a form of guideline, indicating possible trend reversals. In this case, if 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. 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 $N=3.5G_{27}$.

2.4. 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 range from 01.01.1990 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 2 using their Wertpapierkennummer (German security identification number) below.

Table 2. Securities used in system

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

A detailed breakdown of the data splits used for the static and adaptive approaches can be found in the following section.

3. Static and adaptive system specifications

To facilitate a fair comparison of performance, in addition to using the same data for out-of-sample testing, the identical system setup was used for both the static and adaptive implementation. However, a few modifications were required for each specifically, as outlined below.

It is also important to note that both approaches require 100 days of historical data prior to the first trading day to perform technical analysis. This is due to some indicators using large amounts of historical data in their calculations. For this reason a gap year was inserted between the static's training data and the test data, to allow performance monitored trading to commence on the 1st of the year. 1826 trading days worth of data was used representing the time period 01.01.1997 to 31.12.2002. This time period was chosen as the Index gained only 0.14%, however it also included the boom and crash periods around the turn of the millennium.

3.1. Static evolution

In this approach six years worth of data prior to the testing period plus the gap year was used to train a population of 100 agents. 1465 trading days worth of data was used representing the time period from 01.01.1990 to 29.12.1995. As shown previously [19], this time period was assumed to be long enough to evolve a competitive population. This population was trained using the GA outlined in section 2.2 after every run to produce a final solution. The fitness function used the total area under their total asset graph over the entire training duration.

3.2. Adaptive evolution

The adaptive approach is defined by its continuous retraining of the agent population. Consequently, the data is not divided into training and testing data but represents one continuous set. Two initialization procedures were used to also investigate their impact on results. For the first one, 163 days of data from the gap year were reserved for the initial training phase, run over 1000 generations of agents. In the other, the trained-up population from the static system of the corresponding test run was used for initialization. Thereafter both followed the same procedure, where the agent population was retrained after every trading day for 50 generations using the most recent 50 trading days closing price information. The first set is referred to as i, while the latter as ii.

To achieve this, in the first initialization process a trading population (TP) is first established using randomly generated genomes, while for the second initialization process the elite of the evolved agents from the static approach were used to initialize the TP. In this experiment, the TP size was equivalent to that of the elite for the static approach. 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, as outlined in section 2.2 again 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. 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 Figure 2.

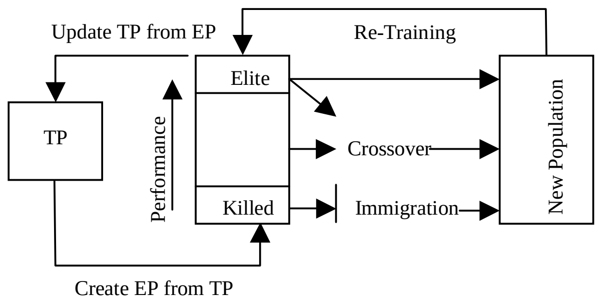

Figure 2. Adaptive evolution

Fitness was determined by comparing the total area under their total asset graph over the retraining duration.

4. Results

For comparison, performance of both approaches over the test data was used, with performance of the buy-and-hold strategy shown to provide some contextual reference on how the market did overall. Each experiment was repeated five times with the average and standard deviation shown. Furthermore, results are shown as the total profit obtained at the end of the period as well as its fitness. Fitness being defined as previously, the above or below average performance of a group over the period being considered, as determined by the area under their total asset graph over time (AUC).

Experiments are denoted as “S-“ followed by a number if from the static system, or “A-“ if from the adaptive system. Furthermore, A-1 to A-5 were initialized randomly and are referred to as i, while A-6 to A-10 were initialized each using the evolved populations from S-1 to S-5 respectively and are referred to collectively as ii.

Table 3. Static system results

	Profit	AUC
Buy-and-Hold	02.34%	270×10^6
S-1	23.74%	251×10^6
S-2	26.19%	247×10^6
S-3	11.42%	238×10^6
S-4	33.65%	262×10^6
S-5	07.90%	247×10^6
Average	20.58%	249×10^6
Std. Dev.	0.10691	9×10^6

Table 4. Adaptive (i) system results

	Profit	AUC
Buy-and-Hold	02.34%	270×10^6
A-1	120.68%	279×10^6
A-2	51.26%	278×10^6
A-3	87.29%	286×10^6
A-4	69.12%	264×10^6
A-5	147.13%	312×10^6
Average	95.10%	284×10^6
Std. Dev.	0.38791	18×10^6

Table 5. Adaptive (ii) system results

	Profit	AUC
Buy-and-Hold	02.34%	270×10^6
A-6	54.51%	282×10^6
A-7	43.51%	266×10^6
A-8	141.92%	383×10^6
A-9	90.59%	359×10^6
A-10	31.35%	294×10^6
Average	72.37%	317×10^6
Std. Dev.	0.44732	51×10^6

As is evident from the tables above, the adaptive approach clearly outperforms its static equivalent in both profit at the end of the test period and more importantly, overall performance in terms of AUC. As might have been expected, as the time period covered included the extreme market boom and crash around the millennium, generally classified as an anomalous period in wider literature, the adaptive approach fared better than its static equivalent.

In terms of agent performance, both static and adaptive outperformed the buy-and-hold strategy in profit, with the adaptive in particular successfully outperforming it with respect to the AUC as well. However the absence of transaction costs must be highlighted in this respect.

Average performance for each trading day from each of the five runs for the three groups relative to the Index is illustrated in Figure 3 below.

5. Analysis and discussion

In all cases the adaptive approach outperformed the static approach. As is clearly evident, whether initialized with the same set of knowledge as the static approach or using a different starting set of agents, the static approach failed to attain similar positive results. Overall, this observation is in-line with expectations, as the market represents a continuously changing complex environment where underlying rules, though generally valid, may from time to time need to be adjusted and different strategies will need to be employed. As the adaptive system provides this ability, it can exploit the emergence of successful short-term trading strategies over the general long-term strategies the static system is dependent on.

Analyzing the difference between adaptive and static approaches, trading activity can also be considered and observations are shown in Table 6 below.

Table 6. Trading activity

	Avg. Buys	Avg. Sells	Correlation (Std. Dev.)
Static	3.88	1.49	0.963 (0.016)
Adaptive i	1.60	0.81	0.588 (0.180)
Adaptive ii	1.54	0.79	0.718 (0.126)

Table 6 illustrates the average number of daily buy and sell actions taken by each agent population from each approach, as well as the average correlation between their performance and that of buy-and-hold. Between the static and adaptive results, a noticeable difference exists in the number of trades conducted for both purchases and sales, indicating a more active approach taken by each group. Both adaptive approaches also have noticeably different correlation value compared to the static approach, as further highlighted by Figure 3, reflecting the reduced trend-following trading behavior exhibited.

Lastly, a Mann-Whitney U Test was used to compare the static results to each adaptive result set, and the adaptive sets to each other. Additionally, trading activity was compared between each group. Results are presented in Table 7 below.

Table 7. Mann-Whitney U Test Results

Test	P-value
Static – Adaptive (i)	0.00794
Static – Adaptive (ii)	0.00794
Adaptive (i) – Adaptive (ii)	0.30952
Static Buy – Adaptive (i) Buy	0.00008
Static Buy – Adaptive (ii) Buy	0.00008

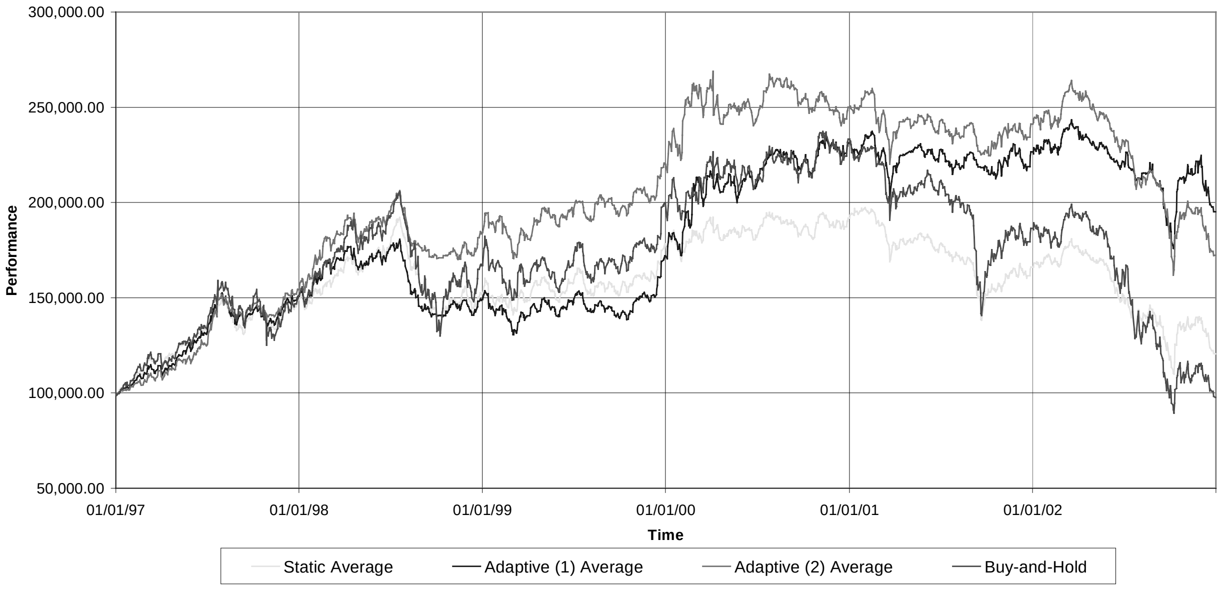

Figure 3. Averaged performance graphs

Adaptive (i) Buy – Adaptive (ii) Buy	0.39710
Static Sell – Adaptive (i) Sell	0.00007
Static Sell – Adaptive (ii) Sell	0.00008
Adaptive (i) Sell – Adaptive (ii) Sell	0.10995

At a significance value of 0.01 there exists a significant difference between the results obtained from the static and adaptive approach, supporting the above conclusions. Furthermore, the remaining results indicate that a significant difference exists between static and adaptive trading behavior, with no significant difference being found between the two adaptive groups.

6. Conclusions

Based on these results, it appears that an adaptive approach offers clear advantages over its static alternative for optimizing trading strategies or populations using technical analysis. Though both implementations used here could be optimized further, by varying re-training periods or population sizes for instance, neither was really given this advantage. Furthermore, previous research [19] seemed to indicate a negligible impact of varying those factors in the static system. Overall therefore it seems advisable to use an adaptive approach in future research attempting to optimize trading strategies using technical trading rules.

7. References

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

[2] F. Allen and R. Karjalainen, “Using genetic algorithms to find technical trading rules”, *J. Financial Economics*, 1999, Vol. 51, pp. 245-271.

[3] 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.

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

[5] 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.

[6] C.L. Dunis and J. Jalilov, “Neural Network Regression and Alternative Forecasting Techniques for Predicting Financial Variables”, *Neural Network World*, 2002, Vol. 12, pp. 113-139.

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

[8] D. Goldberg, *Genetics Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, USA, 1989.

[9] S. Gutjahr, M. Riedmiller and J. Klingemann, "Daily Prediction of the Foreign Exchange Rate Between the US Dollar and the German Mark Using Neural Networks", *Proc. PACES/ SPICIS*, Singapore, 1997.

[10] C. Haefke and C. Helmenstein, "Index Forecasting and Model Selection", *Int'l J. Intelligent Systems in Accounting, Finance & Management*, 2002, Vol. 11, pp. 119-135.

[11] K. Kohara, T. Ishikawa, Y. Fukuhara and Y. Nakamura, “Stock Price Prediction Using Prior Knowledge and Neural Networks”, *Intelligent Systems in Accounting, Finance and Management*, 1997, Vol. 6, pp. 11-22.

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

[13] M. Lettau, "Explaining the facts with adaptive agents: The case of mutual fund flows", *J. Economic Dynamics and Control*, 1997, Vol. 21, pp. 1117-1147.

[14] C. Neely, P. Weller and R. Dittmar, *Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach*, Working Paper 96-006C, Federal Reserve Bank of St. Louis, 1997.

[15] 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*, Singapore, 1996, pp. 17- 25.

[16] O.V. Pictet, M.M. Dacorogna, B. Chopard, M. Oudsaidene, R. Schirru, and M. Tomassini, “Using genetic algorithm for robust optimization in financial applications”, *Neural Network World*, 1995, Vol. 5, pp. 573- 587.

[17] 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.

[18] S. Schulenburg and P. Ross, “An Adaptive Agent Based Economic Model”, *Learning Classifier Systems: From Foundations to Applications*, 2000, Vol. 1813, pp. 265-284.

[19] C. Schoreels and J.M. Garibaldi, "A preliminary investigation into multi-agent trading simulations using a genetic algorithm", *Proc. UK Computational Intelligence Workshop*, Loughborough, UK, 2004, pp. 62-69.

[20] 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.