Financial Data Mining Using Genetic Algorithms Technique: Application to KOSPI 200

Kyung-shik Shin*, Kyoung-jae Kim* and Ingoo Han*

Abstract

This study intends to mine reasonable trading rules using genetic algorithms for Korea Stock Price Index 200 (KOSPI 200) futures. We have found trading rule which would have yielded the highest return over a certain time period using historical data. Simulated results of buying and selling of trading rules were outstanding. These preliminary results suggest that genetic algorithms are promising methods for extracting profitable trading rules.

Key words: data mining, stock market prediction, trading rule extraction, genetic algorithms

1. Introduction

Over the last decades, there has been much research interests directed at understanding and predicting future. Among them, to forecast price movements in stock markets is a major challenge confronting investors, speculator and businesses. In their quest to forecast the markets, they assume that future occurrences are based at least in part on present and past events and data. However, financial time series are among the 'noisiest' and most difficult signals to forecast.

While rule-based technologies improved dramatically, many of stock market applications were less than successful. For this reason, the trend toward automatic learning systems is particularly evident in the financial services sector. Advances in chaos theory provide the theoretical justification for constructing nonlinear models, which is typically the goal in machine learning. Previous studies on this issue suggest that artificial intelligence techniques such as artificial neural networks (ANN) have more frequent chances to detect nonlinear patterns in stock market (Ahmadi, 1990; Kamijo & Tanikawa, 1990; Kimoto *et al.*, 1990; Yoon & Swales, 1991). ANN, however, has a drawback that the users of the model can not readily comprehend the final rules.

In this paper, we propose data mining approach using genetic algorithms (GAs) to solve the knowledge acquisition problems that are inherent in constructing and maintaining rule-based applications for stock market. Although there are an infinite number of possible rules by which we could trade, but only a few of them would have made us a profit if we had been following them. This study intends to find good sets of rules which would have made the most money over a certain historical period.

This paper is organized as follows. The following section provides a brief description of prior research on

* Graduate School of Management
Korea Advanced Institute of Science and Technology

stock market applications using artificial intelligence techniques. Section 3 describes the characteristics of GAs. Section 4 explains the rule extraction methods using GAs. Section 5 and 6 report the experiments and empirical results of Korean stock market application. The final section discusses the conclusions and future research issues.

2. Stock Market Applications Using Artificial Intelligence Techniques

Kimoto *et al.* (1990) used several learning algorithm and prediction method for the Tokyo stock exchange prices index (TOPIX) prediction system. This system used modular neural network that learned the relationships between various factors. The output of this system was the best timing for when to buy and sell stocks. They executed simulation of buy and sell stocks to evaluate the effect of system. In this study, vector curve, turn-over ratio, foreign exchange rate and interest rate were used as input variables. Trading profit using this system revealed more than that of "Buy and hold strategy".

Kamijo and Tanikawa (1990) classified the changing pattern of TOPIX to triangle pattern by use of candlestick chart. They have learned these patterns using the recurrent neural network. The test set of triangle was accurately classified in 15 out of 16 experiments.

Ahmadi (1990) tried to test the 'Arbitrage pricing theory (APT)' by ANN. This study used backpropagation neural network with generalized delta rule to learn relationship between the return of individual stocks and market factors.

Yoon and Swales (1991) performed prediction using mixed qualitative and quantitative data. Qualitative data in this research was information about confidence, economic factors, new products and expected loss, etc. from the Fortune 500 and Business Week's "Top 1000". Quantitative data was obtained from the firm's annual report to the stock holders. The architecture of neural network model was a four-layered network. After the experiment, they compared the results of artificial neural network and multiple discriminant analysis (MDA) and found that the neural network model outperformed MDA approach in prediction of the stock price.

Lee *et al.* (1989) developed the intelligent stock portfolio management system (ISPMS). They attempted to build expert systems aided by optimization techniques. There were two independent external relational databases. One provided the current information about individual stocks and the others provided the selected historical instance of investment. In this study, part of knowledge was generated by machine learning and the other was derived from the experts' opinion. Individual investors displayed personal preference via the preference revelation system. To associate the qualitative factors in the knowledge and preference base (KPB) with quadratic programming (QP) model, the factors in the KPB should be interpreted as decision variables or additional constraints in the QP model. This study demonstrated the capacity of the system to correctly predict whether the price will be up, down or sustained to between 68% and 82%.

Trippi and DeSieno (1992) executed daily prediction of up and down direction of S&P 500 Index Futures using ANN. Generating a composite recommendation for the current day's position. Input variables in this study were technical variables for the two-week period to the trading day, open, high, low, close price, open price and the price fifteen minutes after the market opening of the current trading day. The output variable was long or short recommendation. They performed composite rule generation procedure to generate rules for combining outputs of networks. They reported prediction accuracy was 45.3% - 52.8%.

Duke and Long (1993) also executed daily prediction of German Government Bond Futures using feed-forward back-propagation neural network. In this study, they used opening range (obtained from the highest and lowest bids at the open), highest, lowest price, closing price, volume of traders, open Interest, industrial production, consumer prices, current account balance, unemployment rates, short and long term interest rates, wholesale price index, M3 combined supply and benchmark bond yield as input variables. The following day's closing price was output. A result of the network's predictions as compared to the actual movement was 53.94%.

Choi *et al.* (1995) performed daily prediction of up/down direction of S&P 500 Index Futures. They used open, high, low, close price, moving average, technical indicators like ROC (rate of change), RSI (relative strength index), Market Breakdown as input variables. Their prediction accuracy was 62.5% in test set and 63.8% in whole data.

3. Genetic Algorithms (GAs)

GAs are search algorithm based on the mechanics of natural selection and genetics and they combine survival of the fittest among string structures to form a search algorithm (Davis,1991; Goldberg, 1989; Holland, 1975). GAs have been demonstrated to be effective and robust in searching very large spaces in a wide range of applications (Colin, 1994; Davis, 1991, Fogel, 1993; Goldberg, 1989; Han *et al*, 1997; Klimasauskas, 1992; Koza, 1993). GAs are particularly suitable for multi-parameter optimization problems with an objective function subject to numerous hard and soft constraints.

The financial application of GAs is growing with successful applications in trading system (Colin, 1994; Deboeck, 1994), stock selection (Mahfoud and Mani, 1995), portfolio selection (Rutan, 1993), bankruptcy prediction (Kingdom and Feldman, 1995), credit evaluation (Walker *et al*, 1995) and budget allocation (Packard, 1990).

The main idea of GAs is to start with a population of solutions to a problem, and attempt to produce new generations of solutions which are better than the previous ones. GAs operate through a simple cycle consisting of the following four stages: initialization, selection, crossover, and mutation (Davis, 1991; Wong and Tan, 1994). Figure 1 shows the basic steps of genetic algorithms.

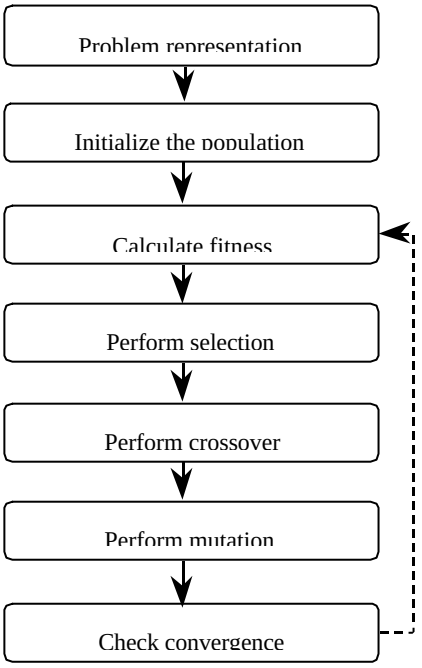

Figure 1. Basic steps of genetic algorithms

In the initialization stage, a population of genetic structures (called chromosomes) that are randomly distributed in the solution space, is selected as the starting point of the search. These chromosomes can be encoded using a variety of schemes including binary strings, real numbers or rules. After the initialization stage, each chromosome is evaluated using a user-defined fitness function. The goal of the fitness function is to numerically encode the performance of the chromosome. For real-

world applications of optimization methods such as GAs, the choice of the fitness function is the most critical step.

The mating convention for reproduction is such that only the high scoring members will preserve and propagate their worthy characteristics from generations to generation and thereby help in continuing the search for an optimal solution. The chromosomes with high performance may be chosen for replication several times whereas poor-performing structures may not be chosen at all. Such a selective process causes the best-performing chromosomes in the population to occupy an increasingly larger proportion of the population over time.

Crossover causes to form a new offspring between two randomly selected 'good parents'. Crossover operates by swapping corresponding segments of a string representation of the parents and extends the search for new solution in far-reaching direction. The crossover occurs only with some probability (the crossover rate). There are many different types of crossover that can be performed: the one-point, the two-point, and the uniform type (Syswerda,1989).

Mutation is a GA mechanism where we randomly choose a member of the population and change one randomly chosen bit in its bit string representation. Although the reproduction and crossover produce many new strings, they do not introduce any new information into the population at the bit level. If the mutant member is feasible, it replaces the member which was mutated in the population. The presence of mutation ensures that the probability of reaching any point in the search space is never zero.

4. Trading Rule Extraction

Although there are an infinite number of possible rules by which we could trade, it seems that only a few of them would have made a profit. To find the rule that would have yielded the most profit had it been used to trade stocks on a given set of historical data, firstly, we develop trading rules of this general form:

IF the indicator 1 is GREATER THAN OR EQUAL TO (LESS THAN) X1,
AND the indicator 2 is GREATER THAN OR EQUAL TO (LESS THAN) X2,
AND the indicator 3 is GREATER THAN OR EQUAL TO (LESS THAN) X3,
AND the indicator 4 is GREATER THAN OR EQUAL TO (LESS THAN) X4,
AND the indicator 5 is GREATER THAN OR EQUAL TO (LESS THAN) X5,
THEN buy, ELSE sell

There are five conditions that are evaluated for each trading day. If the all of five conditions are satisfied, then the model will produce ‘buy’ signal on that day, otherwise it will suggest ‘sell.’ X1 to X5 denotes the cutoff values. The cutoff values range from 0 to 1, and represent the percentage of the data source's range. For example, if RSI (relative strength index) ranges from 0 to 100, then a cutoff value of 0.0 would match a RSI of 0, a cutoff value of 1.0 would match a RSI of 100, and a cutoff value of 0.5 would match a RSI of 50. This allows the rules to refer to any data source, regardless of the values it takes on.

We consider additional flexibility regarding the indicator component of the rule structure such as ‘today’s value,’ ‘last day’s value,’ and ‘change since the last day’s value.’ Translating this in its full form, for example, would yield the following statement:

IF TODAY’ S VALUE of ROC is GREATER THAN OR EQAUAL TO 30.0,
AND CHANGE SINCE THE LAST DAY’ S VALUE of RSI is LESS THAN 60.0,

AND LAST DAY' S VALUE of Stochastic %D is LESS THAN 51.0,

AND LAST DAY' S VALUE of A/D Oscillator is LESS THAN 12.5,

AND TODAY' S VALUE of Stochastic %K is LESS THAN 75.9,

THEN buy ELSE sell

Above rule structure is summarized in Table 1. In Table 1, 'which data' means data source the rule refers to, and 'modifier' means a modifier value that determines if the value itself should be examined, or if the last day's value or the change since the last day should be examined.

There has been much debate regarding the development of trading system using historical data. We agree that the future is never exactly like the past, however, a common investment approach is to employ systems that would probably have worked well in the past and that seem to have a reasonable chance of doing well in the future. So, we define a goal of the system as finding a rule which would have yielded the highest return over a certain time period.

In setting up the genetic optimization problem, we need the parameters that have to be coded for the problem and an objective or fitness function to evaluate the performance of each string. The parameters that are coded are the cell values of Table 1. The varying parameters generate a number of combinations of our general rules.

The task of defining a fitness function is always application specific. In this case, the objective of the system is to find a trading rule which would have yielded the highest return over a certain time period. We apply the trading profit to the fitness function for this study.

5. Data and Variables

The research data used in this study is KOSPI 200 (Korea stock price index 200) from May 1996 through October 1998. KOSPI 200 is the underlying index of KOSPI 200 future which is the first derivative instrument in Korean stock market. Futures are the standard forms that decide the quantity and price in the certified market (trading place) at certain future point of time (delivery date). General functions of futures market are supplying information about future price of commodities, function of speculation and hedging (Kolb & Hamada, 1988). Being different from the spot market, futures market does not have continuity of price data. That is because futures market has price data by contract. So, in futures market analysis, nearest contract data method is mainly used and incorporated in this research. We collected a sample of 660 trading days.

Table 1. The general structure of trading rule

Rule number	1	2	3	4	5	Description
Which data	IND_{i1}	IND_{i2}	IND_{i3}	IND_{i4}	IND_{i5}	IND_{Ii} (i=1, …,n, j = cond. number)
Less than / greater than or equal to	1 or 2	1 or 2	1 or 2	1 or 2	1 or 2	1= Less than / 2= greater than or equal to
Cutoff value	X_1	X_2	X_3	X_4	X_5	Cutoff X_j (j = cond. number)
Modifier	1,2 or 3	1,2 or 3	1,2 or 3	1,2 or 3	1,2 or 3	1= today's value, 2= last day's, 3= change since the last day

Table 2 Technical indicators (adapted from Achelis, 1995 and Kolb & Hamada, 1988)

Name	Description	Formulas
Stochastic %K	The Stochastic Oscillator compares where a security's price closed relative to its price range over a given time period.	$\frac{C_t - L_n}{H_n - L_n} \times 100$
Stochastic %D	The Stochastic Oscillator compares where a security's price closed relative to its price range over a given time period.	$\frac{\sum_{i=0}^{n-1} \%K_{t-i}}{n}$
Momentum	The Momentum indicator measures the amount that a security's price has changed over a given time span.	$C_t - C_{t-4}$
ROC	The Price Rate-of-Change (ROC) indicator displays the difference between the current price and the price x periods ago.	$\frac{C_t}{C_{t-n}} \times 100$
A/D Oscillator	The A/D Oscillator measures the accumulation and distribution of market power.	$\frac{H_t - C_{t-1}}{H_t - L_t}$
Disparity (5 days)	The Disparity means the distance of current spot price and moving average.	$\frac{C_t}{MA_n} \times 100$
CCI	The Commodity Channel Index (CCI) measures the variation of a security's price from its statistical mean.	$\frac{(M_t - SM_t)}{(0.015 \times D_t)}$
OSCP	The Price Oscillator shows the difference between two moving averages of a security's price.	$\frac{MA_5 - MA_{10}}{MA_5} \times 100$
RSI	The RSI is price following oscillator that ranges from 0 to 100.	$100 - \frac{100}{1 + \frac{\sum_{i=0}^{n-1} Up_t / n}{\sum_{i=0}^{n-1} Dw_t / n}}$

Note) C: Closing price, L: Low price, H: High price, Volume: Trading volumes

MA: Moving average of price,

$$M_t : \frac{(H_t + L_t + C_t)}{3}$$

$$SM_t : \frac{\sum_{i=1}^{n} M_{t-i+1}}{n}$$

$$D_t : \frac{\sum_{i=1}^{n} |M_{t-i+1} - SM_t|}{n}$$

Many previous stock market analyses have used technical or fundamental indicator. In general, fundamental indicators are mostly used for long-term trend analysis while technical indicators are used for short-term pattern analysis. In this research, we use the technical indicators as input variables. We choose 9 technical indicators to narrow the set of variables. The total available indicators used to search trading rules are technical indicators such as Stochastic %K, Stochastic %D, Momentum, ROC (rate of change), A/D Oscillator

(accumulation / distribution oscillator), Disparity 5 days, CCI (commodity channel index), OSCP (price oscillator) and RSI (relative strength index). The description and formulas of technical indicators are presented in Table 2.

6. Experimental Result

To find the profitable trading rules, we apply GAs model proposed in the previous section. We use 500 chromosomes in the population for this study. The crossover and mutation rates are changed to prevent the output from falling into the local optima. The crossover rate ranges 0.5 - 0.7 and the mutation rate ranges 0.06 - 0.12 for this experiment. These processes are done by the genetic algorithms software package *Evolver* 4.0, called from an Excel macro.

We extract three trading rules by genetic search process. The derived rules are alternatively good trading rules although there is minor difference in simulated performance. Each of rules consists of five conditions referring input factors. The descriptions of rules derived are illustrated in Table 3.

Trading profit earned from simulation results are summarized in Table 4 and Figure 2. While the underlying index decreased more than 58% during the training period, we could find rules that would have yielded the high level of profit had it been used to trade stocks on a given set of historical data. The rules derived by learning training data using GAs are applied validation (holdout) samples to verify the effectiveness of the proposed approach. While the underlying index decreased about 19% during the validation period (January 1998 – October 1998), the trading strategies followed by the derived rules earn 13% to 26% of trading profit during the period. These preliminary results demonstrate that GAs are promising methods for extracting profitable trading rules. Their success is due to their ability to learn nonlinear relationships among the input variables.

Additional advantage of this approach is that the model generates comprehensible rules while ANN has a drawback that the users can not readily comprehend the final rules. These rules can serve as approximate explanations of how the various technical indicators related to future stock market returns.

However, trading rules generated by GAs produce predictions only when the rules are fired. Each of the rules extracted produces predictions less than 20% of time. We may think that GAs do not produce predictions when the market is nearly equally likely to move in either direction.

Table 3. Trading rules generated

Rule number	Description of rules
Rule 1	If yesterday's AD OSC is greater than or equal to 0.329, AND change in ROC is less than 0.667, AND RSI is less than 0.642, AND yesterday's %D is less than 0.733, AND Disparity5 is less than 0.755, THEN buy the stock, Else sell or hold.
Rule 2	If yesterday's AD OSC is greater than or equal to 0.343, AND change in ROC is less than 0.641, AND RSI is less than 0.604, AND yesterday's %D is less than 0.695 AND Disparity5 is less than 0.778, THEN buy the stock, Else sell or hold.
Rule 3	If yesterday's AD OSC is greater than or equal to 0.343, AND change in ROC is less than 0.641, AND RSI is less than 0.618, AND yesterday's %D is less than 0.695, AND Disparity5 is less than 0.778, THEN buy the stock, Else sell or hold.

Table 4. The profit of buying and selling simulations

Trading Strategy	Accumulated amount (assume initial investment of 10,000 won)	
	Training period (May 1996 – November 1997)	Validation period (January 1998 – October, 1998)
Following Buy & Hold	4,167 won (-58.3%)	8,105 won (-19.0%)
Following the rule set 1	14,938 won (49.4%)	11,314 won (13.1%)
Following the rule set 2	13,237 won (32.4%)	12,641 won (26.4%)
Following the rule set 3	13,252 won (32.5%)	12,641 won (26.4%)

* US$1 = 1,400 won (approximate)

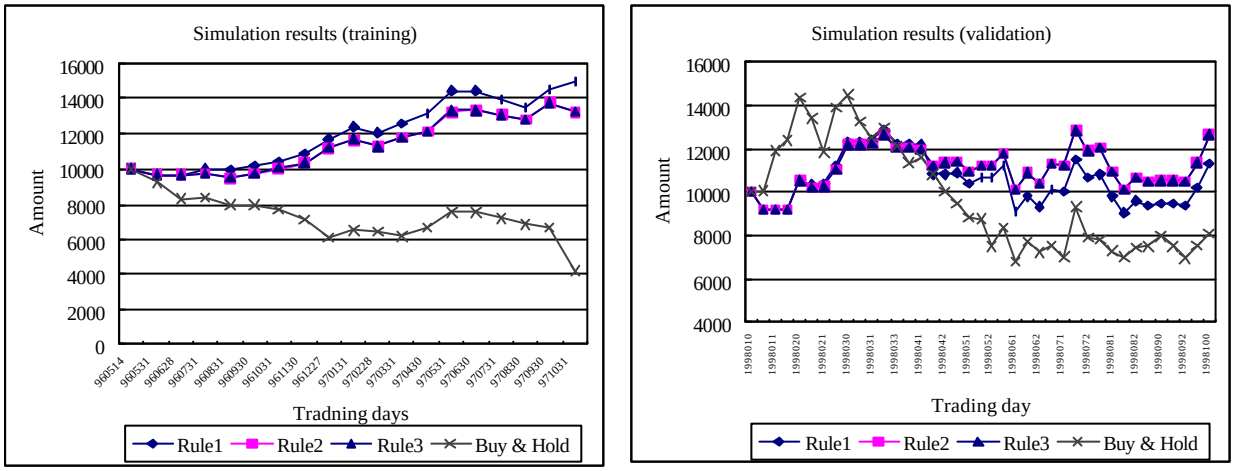

Figure 2. Performance of simulations

7. Concluding Remarks

This study intends to mine reasonable trading rules using GAs for Korea Stock Price Index 200 (KOSPI 200) futures. We have found three alternative good rules which would have yielded the high return over a certain time period. Simulated results of buying and selling of trading rules were outstanding. These preliminary results suggest that GAs are promising methods for extracting profitable trading rules.

However, the future is never exactly like the past. Although the trading systems that have worked well in the past seem to have a reasonable chance of doing well in the future, we need a more extensive validation process. Since the entire history of Korean future market is less than 2 and half years, we have difficulty in doing out-of-sample validations in this study. We are working towards verifying and enhancing trading rules using current data in Korean future market.

Reference

Achelis, S. B., *Technical Analysis from A to Z*, Probus Publishing, 1995.

Ahmadi, H., "Testability of the arbitrage pricing theory by neural networks," *Proceedings of the IEEE International Conference on Neural Networks*, 1990, pp. 1385-1393.

Choi, J. H., Lee M. K. and Rhee M. W., "Trading S&P 500 stock index futures using a neural network," *Proceedings of the Third Annual International Conference on Artificial Intelligence Applications on Wall Street*, 1995, pp. 63-72.

Colin, A. M.,"Genetic algorithms for financial modeling," In Deboeck, G.J. (Eds.), *Trading On The Edge*, New York: John Wiley, 1994, pp.148-173.

Davis, L., *Handbook of genetic algorithms*, Van Nostrand Reinhold, NY, 1991.

Deboeck, G. J., "Using GAs to optimize a trading system," In Deboeck, G.J (Eds.), *Trading On The Edge.* New York: John Wiley, 1994, pp.174-188.

Duke, L.S. and Long, J., "Neural network futures trading - A feasibility study," *Adaptive Intelligent Systems*, Elsevier Science Publishers, 1993.

Fogel, D. B., "Applying evolutionary programming to selected traveling salesman problems," *Cybernetics and Systems*, Vol.24, No.1, 1993.

Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, MA : Addison-Wesley, 1989.

Han, I., Jo, H. and Shin, K. S., "The hybrid systems for credit rating," *Journal of the Korean Operations Research and Management Science Society*, 22(3), pp.163-173, 1997.

Holland, J. H., *Adaptation in natural and artificial systems*, Ann Arbor: The University of Michigan Press, 1975.

Kamijo, K. and Tanigawa, T., "Stock price pattern recognition: A recurrent neural network approach'" *Proceedings of the IEEE International Joint Conference on Neural Networks*, 1990, pp. 1215-1221.

Kimoto, T., Asakawa, K., Yoda, M. and Takeoka, M., "Stock market prediction system with modular neural networks," *Proceedings of the IEEE International Joint Conference on Neural Networks*, 1990, pp. 11-16.

Kingdom, J. and Feldman, K., *Genetic algorithms for bankruptcy prediction*, Search Space Research Report, No.01-95, Search Space Ltd. London, 1995.

Klimasauskas, C. C., "Hybrid neuro-genetic approach to trading algorithms," *Advanced Technology for Developers*, Vol.1, No.7, 1992.

Kolb, R. W. and Hamada, R. S., *Understanding Futures Markets*, Scott, Foreman and Company, 1988.

Koza, J., *Genetic programming*, Cambridge: The MIT Press, 1993.

Lee, J. K., Kim, H. S. and Chu S. C., "Intelligent stock portfolio management system," *Expert Systems*, Vol. 6, No. 2, 1989, pp. 74-87.

Mahfoud, S. and Mani, G., "Genetic algorithms for predicting individual stock performance," *Proceedings of the 3rd International Conference on Artificial Intelligence Applications on Wall Street*, pp.174-181, 1995.

Packard, N., "A genetic learning algorithm for the analysis of complex data," *Complex Systems*, 4, 1990, pp.543-572.

Rutan, E., "Experiments with optimal stock screens," *Proceedings of the 3rd International Conference on Artificial Intelligence Applications on Wall Street*, 1993, pp.269-273.

Syswerda, G., "Uniform crossover in genetic algorithms," In Schaffer, J.D. (Eds.), *Proc. 3d Int.Conf. Genetic Algorithms.* San Maeto, CA: Morgan Kaufmann, 1989.

Trippi, R. R. and DeSieno D., "Trading equity index futures with a neural network," *The Journal of Portfolio Management*, 1992.

Walker, R., Haasdijk, E. and Gerrets, M., "Credit evaluation using a genetic algorithm," In Coonatilake, S. and Treleaven, P. (Eds.), *Intelligent Systems for Finance and Business*, Chichester, John Wiley, 1995, pp.39-59.

Wong, F. and Tan, C., "Hybrid neural, genetic and fuzzy systems," In Deboeck, G.J. (Eds.), *Trading On The Edge*. New York: John Wiley, 1994, pp.245-247.

Yoon, Y. and Swales G., "Predicting stock price performance: A neural network approach," *Proceedings of the IEEE 24th Annual Conference on Systems Sciences*, 1991, pp. 156-162.