Proceedings, Florida Artificial Intelligence Research Symposium, USA, 1999

Improving Technical Analysis Predictions: An Application of Genetic Programming

JIN LI AND EDWARD P.K. TSANG[*]

Department of Computer Science, University of Essex, Wivenhoe Park, Colchester, United Kingdom
{jli, edward}@essex.ac.uk
URL: http://cswww.essex.ac.uk/FAT

ABSTRACT

Recent studies in finance domain suggest that technical analysis may have merit to predictability of stock. Technical rules are widely used for market assessment and timing. For example, moving average rules are used to make "buy" or "sell" decisions at each day. In this paper, to explore the potential prediction power of technical analysis, we present a genetic programming based system FGP (Financial Genetic Programming), which specialises in taking some well known technical rules and adapting them to prediction problems. FGP uses the power of genetic programming to generate decision trees through efficient combination of technical rules with self-adjusted thresholds. The generated rules are more suitable for the prediction problem at hand. FGP was tested extensively on historical S&P 500 data through a specific prediction problem. Preliminary results show that it outperforms commonly used, non-adaptive, individual technical rules with respect to prediction accuracy and annualised rate of return over two different out-of-sample test periods (three and a half year in each period).

KEY WORDS: finance; prediction; genetic programming; investment, technical rules

Biography

Jin Li *holds Master degree in Computer Science (Hefei University of Technology, China). He is currently a PhD student in the Computer Science Department, University of Essex, UK. His research is in financial applications of artificial intelligence.*

Dr Edward Tsang *holds a Bachelor degree in Business Administration (Chinese University of Hong Kong) and a PhD in Computer Science (University of Essex, UK). He is currently a Reader in Computer Science, University of Essex. He has broad research interest in applied artificial intelligence.*

[*] Contact author: Dr Edward Tsang, Department of Computer Science, University of Essex, Colchester CO4 3SQ, UK; tel: +44 1206 872774; fax: +44 1206 872788

1. INTRODUCTION

As an approach to financial forecasting, technical analysis is based on the belief that historical price series, trading volume, and other market statistics exhibit regularities. There are two general approaches in technical analysis: one involves qualitative techniques and the other quantitative techniques. The qualitative techniques rely on the interpretation of the form of geometric patterns such as *double bottoms*, *head-and-shoulders,* and *support and resistance levels*; whilst the quantitative techniques try to create indicators such as *moving average* (MV*), relative strength indicators* (RSI), etc. Nonetheless, both techniques can be characterised by appropriate sequences of local minima and/or maxima (Neftci 1991).

According to the weak form of *efficient market hypothesis* (EMH) (Fama 1970; Malkiel 1992), since historical price information is already reflected in the present price, technical analysis is totally useless for predicting future price movements. In recent years, however, this hypothesis has been directly challenged by a fair amount of studies, which supply evidence of predictability of security return from historical price patterns. Evidence in support of technical analysis can be classified into two categories: (1) evidence on systematic dependencies in security return (e.g., autocorrelations of security returns, see Campbell *et al.* 1997; Jegadeesh 1990; Lo & MacKinlay 1990). Researchers have reported positive or negative series correlation in returns of individual stocks or indices on various time period bases (e.g. daily, weekly, several monthly or yearly); (2) evidence on the returns earned by technical rules, including momentum and contrarian strategies (e.g., Jegadeesh & Titman 1993; Lehmann 1990; Werner *et al.* 1987), moving average rules and trade range breakout rules (Lukac *et al.* 1988, Brock *et al.* 1992). It is not the purpose here to provide theoretical or empirical justification for the technical analysis. The aim of this study is to show how genetic programming (GP) (Koza 1992), a class of algorithms in evolutionary computation, can be employed to improve technical rules. We demonstrate our approach in a particular forecasting task based on the Dow Jones Industrial Average (DJIA).

Quantitative technical rules are often used to generate buy or sell signals based on each rule interpretation. One may want to use technical rules to answer questions such as "*is today is a good time to buy if I want to achieve a return of 4% or more within the next 63 trading days?*" and "*is today the right time to sell if I want to avoid a lost of 5% or more within the next 10 days?*". However, the way technical rules are commonly used may not be adequate to answer these questions. How to efficiently apply them and adapt them to these specific prediction problems is a non-trivial task. We propose a GP approach that is capable of combining individual technical rules and adapt the thresholds based on past data. Rules generated by our GP can achieve performances that cannot be achieved by those individual technical rules in their normal usage.

EDDIE (which stands for Evolutionary Dynamic Data Investment Evaluator) is a forecasting system to help investors to make the best use of the information available to them (Tsang *et al*. 1998). Such information may include technical rule indicators, individual company's performance indicators, expert predictions, etc. Our first implementation, EDDIE-1, was applied to the horse racing domain, whose similarity with financial forecasting has well been documented (Asch *et. al*. 1984; Hausch & Ziemba 1985; de la Maza 1989). The idea was then extended to financial forecasting. EDDIE allows a potential investor to make hypotheses about the factors that are relevant to a forecast. It then tests those hypotheses using historical data and evolves, by way of natural selection, decision trees which aim to provide a good return on investment (ROI). Preliminary but promising results were presented in (Butler 1997; Tsang *et. al.* 1998). FGP (Financial Genetic Programming) is a descendent of EDDIE. In this paper, we will exam how FGP can be applied to a specific prediction problem: ≥4% return within 63 trading days in the DJIA index.

2. BACKGROUND

Evolutionary computation is a standard term that encompasses a class of search, adaptation, and optimisation techniques based on the principles of natural selection and evolution. These techniques include *genetic algorithms* (GAs) (Holland 1975; Goldberg 1989), *evolution strategies* (ESs) (Rechenberg 1973; Schwefel 1981) and *evolutionary programming* (EP) (Fogel *et. al.* 1966). Bäck (1996) provides an excellent review of all these three paradigms. *Genetic programming* (GP) is a promising variant of genetic algorithms that uses tree representations instead of strings. In evolutionary computation, a population (set) of candidate solutions is maintained. For example, a candidate solution could be a decision tree for forecasting. A fitness function is needed to evaluate the quality of each candidate solution with regard to the task to be performed (e.g. how good is a rule for forecasting in our application?). Candidate solutions are selected randomly, biased by their fitness, for involvement in generating members of the next generation. General mechanisms (referred to as *genetic operators*, e.g. reproduction, crossover, mutation) are used to combine or change the selected candidate solutions to generate offspring, which will form the population in the next generation. A simple common evolutionary algorithm (EA) is summarised by Figure 1.

```
1. Create randomly the initial population $P(0)$;  set $i = 0$
2. Repeat
    (a) Evaluate the fitness of each individual in $P(i)$ using fitness function
    (b) Select parents from $P(i)$ using selection strategies
    (c) Generate new generation $P(i +1)$ using genetic operators
3. Until $i < N$ or the time is up; where $N$  is maximum generation set by users
```

Figure 1. A simple evolutionary algorithm

Evolutionary computation has been applied to a broad range of problems with some success from traditional optimisation in engineering and operational research (Bäck 1997) to non-traditional areas such as data mining, composition of music and financial prediction (Kinnear 1994; Angeline & Kinnear 1996;   Koza 1996).

In FGP, a candidate solution is represented by a *genetic decision tree* (GDT). The basic elements of GDTs are *rules* and *forecast values*, which correspond to the *functions* and *terminals* in GP. Figure 2 shows an example of a simple GDT. A useful GDT in the real world is almost certainly a lot more sophisticated than this. In GP terms, the questions in the example GDT are *functions*, and the proposed actions are *terminals*, which may also be forecast values. In this example, the GDT is binary; in general, this need not be the case.

A GDT can be seen as a set of rules. For example, one of the rules expressed in the GDT in figure 2 is:

> IF X's price-earning ratio is 10% or more below the average in DJIA
>
> AND X's price has risen by 5% or more than the minimum price of last 63 days,
>
> THEN Buy X.

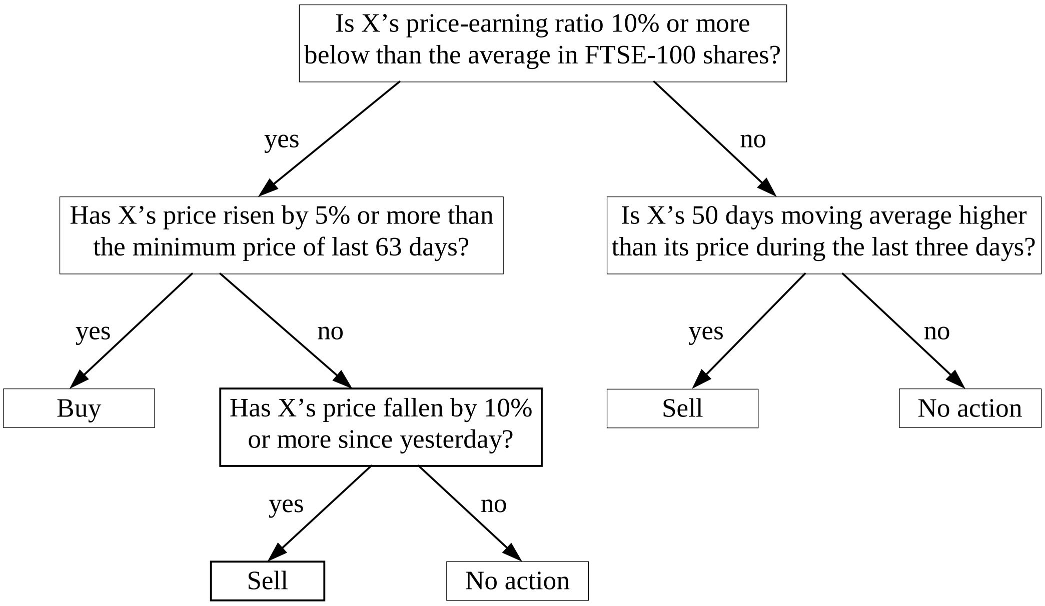
Is X's price-earning ratio 10% or more below than the average in FTSE-100 shares?
yes
no
Has X's price risen by 5% or more than the minimum price of last 63 days?
Is X's 50 days moving average higher than its price during the last three days?
yes
no
yes
no
Buy
Has X's price fallen by 10% or more since yesterday?
Sell
No action
yes
no
Sell
No action

Figure 2. A (simplistic) GDT concerning the actions to take with Share X

GP is attractive for financial applications because it manipulates GDTs (as opposed to strings in GAs). This allows one to handle rule sets of variable size (typical GAs use representations with fixed length). Besides, rules are easy to understand and evaluate by human users, which makes them more attractive than neural networks, most of which are black boxes (Goonatilake & Treleaven 1995).

For a GP to work, one must be able to evaluate each GDT, and assign to it a *fitness* value, which reflects the quality of the GDT to solve the problem at hand. Our GP maintains a set of GDTs called a *population* and works in iterations. In each iteration, FGP creates a new generation of population using standard genetic crossover, mutation and reproduction operators. Given two GDTs (called parents), the crossover operator in FGP works as follows:

a) randomly select a node within each parent GDTs as a crossover point,

b) exchange the subtrees rooted at the selected nodes to generate two children.

Mutation is employed to keep a population with sufficient diversification. It works as follows:

a) randomly select a node within parent tree as the mutation point

b) generate a new tree with a limited depth

c) replace the subtree rooted at the selected node with the generated tree.

In general, crossover is employed with high probability (e.g. 0.9), whereas mutation has a low probability (e.g. 0.01) or less depending on problems. Besides, the reproduction is also used in the process to evolve new generation as asexual operator (e.g. probability 0.1) in order to increase the number of occurrences of individual GDTs with higher fitness. FGP provides the users with two selection strategies, namely *roulette wheel* and *tournament*. We used the latter in our tests. (The advantage of tournament selection is justified in (Miller & Goldberg 1995).) Its algorithm, which is simple to implement, is described in the following pseudocode:

```
Tournament_Selection()
Begin
    Randomly select $N$ individuals from population;        /* N is tournament size */
    Choose the one individual with the highest fitness value among  the $N$ individuals
End
```

There are many variations in the way the initial population is generated, the way that the population is updated, the way that crossover and mutation is done, etc. (e.g. see Angeline & Kinnear 1996; Koza *et. al*. 1992, 1996; Mitchell, 1996). These will not elaborated here.

3. FGP FOR PREDICATION IN DJIA INDEX

We took the Dow Jones Industrial Average (DJIA) index data from 7 April 1969 to 11 October 1976 (1,900 trading days) as training data (or in-sample data) to generate GDTs, and tested them on the data (or out-of-sample data) from 12 October 1976 to 5 May 1980 (900 trading days), which we shall refer to as "test data I". We used a population size of 1,800, crossover rate of 90%, reproduction rate of 10% and a mutation rate of 1%. The termination condition was 2 hours on a Pentium II (200 MHz) or 30 generations, whichever reached first. Each run generates one GDT; 20 runs were completed in our experiments. The generated rules were used to predict whether the following goal is achievable at any given day:

Goal G: the index will rise by 4 % or more within the next 63 trading days (3 months).

Accordingly, each trading day is classified into "buy" category if G holds or "not-buy" category if G does not hold. The whole training and test period contained roughly 50% of "buy" categories. We used {If-then-else, And, Or, Not, <, >} as functions. Terminals were conclusions, numbers or indicators. Conclusions could be either *Positive* (meaning that G is predicted to be achievable) or *Negative*. Six technical rule indicators were derived from rules in the finance literature, such as (Alexander 1964; Brock *et. al*. 1992; Fama & Blume 1966; Sweeney 1988). They are listed as follows:

(1) MV_12 = Today's price – the average price of the previous 12 trading days

(2) MV_50 = Today's price – the average price of the previous 50 trading days

(3) Filter_5 = Today's price – the minimum price of the previous 5 trading days

(4) Filter_63 = Today's price – the minimum price of the previous 63 trading days

(5) TRB_5 = Today's price – the maximum price of the previous 5 trading days (based on the Trading Range Breakout rule [Brock et. al. 1992]).

(6) TRB_50 = Today's price – the maximum price of the previous 50 trading days

Each of the above six indicators is related to some technical analysis rules in the literature. We compared the corresponding six individual technical rules with the GDTs generated by FGP in terms of two criteria: *prediction accuracy* (the percentage of correct predictions) and *annualised rate of return* (ARR). Six individual technical rules using the above six indicators generate "buy" or "not-buy" signals in the following ways. The *moving average rules* (1) and (2) generate "buy" signals if today's price is greater than the average price of the preceding *n* days (*n* = 12 and 50 respectively). The *filter rules* (3) and (4) generate "buy" signals if today's price has risen 1% than minimum of previous *n* days (*n* = 5 and 63 respectively). Here 1% is a threshold which an investor has to choose. The *trading range breakout rules* (5) and (6) generate "buy" signals if today's price is greater than the maximum price of previous *n* days (*n* = 5 and 50 respectively). ARR was calculated based on the following trading behaviour:

> ***Hypothetical trading behaviour****: we assume that whenever a buy signal is indicated by a rule, one unit of money was invested in a portfolio reflecting the DJIA index. If the DJIA index does rise by 4% or more at day t within the next 63 days, then we sell the portfolio at the index price of day t. If not, we sell the portfolio on the 63rd day, regardless of the price.*

For simplicity, we ignored transaction costs and any difference between buying and selling prices. Rules generated by FGP were tested against the above six individual technical rules in the test data. Results are shown in the column of "On test data I" in table 1. Among the six technical rules, Filter_5 performed best in this set of data. It achieved an accuracy of 52.67% and ARR of 23.03%. The 20 FGP achieved an accuracy of 57.97% in average and an average ARR of 27.79%, which is better than the Filter_50 rule. In fact, even the poorest GDT achieved an accuracy of 53.00 (rule 18) and ARR of

23.57% (rule 2), which are still better than the Filter_50 rule. Our results show conclusively that FGP is capable of generating good rules based on the same indicators used by the technical rules.

Table 1. Performance comparisons between 20 FGP-generated GDTs and six technical rules on test data I (12/10/1976-05/05/1980-900 trading days) and on test data II (10/04/1981-29/10/1984-900 trading days) of the Dow Jones Industrial Average (DJIA) Index

20	**On test data I**		**On test data II**	
	Accuracy	**ARR**	**Accuracy**	**ARR**
Rule 1	60.22%	27.56%	53.89%	55.25%
Rule 2	53.67%	23.57%	53.11%	54.74%
Rule 3	62.00%	31.71%	57.89%	57.80%
Rule 4	58.00%	36.55%	52.33%	61.81%
Rule 5	60.22%	28.23%	63.33%	58.53%
Rule 6	55.11%	29.76%	55.00%	67.00%
Rule 7	61.33%	30.52%	58.00%	56.23%
Rule 8	57.89%	27.16%	54.00%	54.54%
Rule 9	60.67%	28.75%	61.56%	57.69%
Rule 10	55.78%	26.34%	59.11%	59.83%
Rule 11	62.44%	25.93%	55.22%	57.37%
Rule 12	56.78%	25.88%	56.44%	53.19%
Rule 13	56.11%	26.85%	56.44%	59.91%
Rule 14	60.56%	29.66%	54.89%	53.12%
Rule 15	54.78%	25.43%	55.11%	58.05%
Rule 16	56.00%	25.82%	58.11%	59.10%
Rule 17	60.56%	29.18%	58.00%	58.82%
Rule 18	53.00%	23.82%	58.67%	56.61%
Rule 19	60.67%	28.80%	62.89%	58.56%
Rule 20	53.67%	24.18%	57.22%	56.38%
Highest	62.44%	31.71%	63.33%	67.00%
Lowest	53.00%	23.57%	52.33%	53.12%
Mean	**57.97%**	**27.79%**	**57.06%**	**57.73%**
Standard Deviation	3.07%	3.06%	3.06%	3.15%
6 Technical Rules	(the best result in each column is highlighted)			
PMV_12	51.44%	20.68%	44.89%	36.66%
PMV_50	42.56%	16.94%	41.89%	46.85%
TRB_5	49.44%	18.18%	47.44%	55.33%
TRB_50	47.44%	-5.34%	48.67%	**67.00%**
Filter_5	**52.67%**	**23.03%**	**49.44%**	54.53%
Filter_63	50.56%	22.77%	48.89%	48.76%

Following is the simplest GDT produced by FGP: (from GDT 14 shown in Table 1)

```
(IF  (PMV_50  <  -28.45) THEN  Positive
ELSE (IF ((TRB_5 > -29.38) AND (Filter_63 < 66.24))
THEN Negative
ELSE Positive))
```

This rule suggests that if today's price is more than 28.45 below the average price of last 50 days, then it is a good time to buy (i.e. the rule predicts that one could achieve a return of 4% or more within the next 3 months); otherwise whether one should buy or not depends on the values of TRB_5 and Filter_63. If today's price is no more than 29.38 above the maximum price of the previous 5 trading days or today's price is more than 66.24 above the minimum price in the last 63 days, then it is another good chance to buy. The logical structure and thresholds involved in the above rule may not be found manually by investors. We argue that the success of FGP in this set of test data was due to effective combination of individual technical indicators and exploration of thresholds in these rules.

To test the robustness of the 20 GDTs across different time periods, we applied them to a more recent period. We tested them on the DJIA index from 10 April 1981 to 29 October 1984 (900 trading days), which we shall refer to as "test data II". The test results are illustrated in the column of "On test data II" in table 1. The GDTs achieved an average accuracy of 57.06%, which out-performs all the six technical rules. As in test data set I, even the poorest GDT performed better than all the technical rules on prediction accuracy. The GDTs achieved an average ARR of 57.73%, which is also better than the ARRs produced by the technical rules except the TRB_50 rule.[1] Test results on data set II further demonstrates the qualify of the GDTs generated by FGP.

Two issues are worth pointing out. First, although the number of runs is relatively small, the results are significant because the amount of data tested is large and the results are consistent. It is encouraging to see that our GDTs achieve nearly the same mean of accuracy (57.97%, 57.06%) with almost the same value of standard deviation (3.07%, 3.06%) in the two test periods of different characteristics (test period II was more bullish than test period I). Second, it should be pointed out that our calculation of ARR assumes that funds are always available whenever a positive position is predicted, and such funds

[1] Note that the TRB_50 rule is not particularly reliable. It achieved the lowest ARR in test data I (−5.34%) but the highest ARR in test data II (67.00%). The erratic performance of the TRB_50 rule is partly due to the fact that it generates very few buy signals.

have no cost when idle. Exactly how one can make use of the predictions by the GDTs is a non-trivial issue.

4. RELATED WORK

Though still at its infancy, work on application of evolutionary computation in finance have been published. Bauer (1994) reported his GAs intelligent systems which aim at finding tactic market timing strategies; Chen & Yeh (1996) attempted to formalize the notion of unpredictability in the efficient market hypothesis in terms of search intensity and chance of success in the search conducted by genetic programming; Mahfoud & Mani (1996) presented a new genetic-algorithm-based system and applied it to the task of predicting the future performances of individual stocks; Neely *et al.* (1997) and Oussaidene *et al.* (1997) applied genetic programming to foreign exchange forecasting and reported some success. This work followed our earlier work in EDDIE (Butler 1997; Tsang et. al. 1998).

5. CONCLUSION AND FURTHER WORK

While technical analysis is widely used as an investment approach among practitioners, it is rarely accepted by academics. It is not our role to defend technical analysis here, although our results show that there is some predictability in the DJIA index based on historical data alone. Our main objective in this paper is to illustrate that FGP, a genetic programming based system, can improve technical rules by taking indicators used in them as input and generate decision trees. For the specific task of predicting whether one can "achieve 4% return with 63 trading days in the DJIA index", FGP reliably generated accurate GDTs. This involves combining indicators in individual technical rules and finding thresholds in different parts of the decision trees by the way of evolutionary back testing using historical data. Our experiment results show that the generated rules can achieve better performances which cannot be obtained by individual rules.

The application presented here is not complete since important issues such as transaction costs, risk of investment, and capital adequacy were totally ignored. In the future work, we are going to consider these factors. We also intend to bring in constraint satisfaction techniques, which have been demonstrated to be useful in genetic algorithms (Lau & Tsang 1997; Tsang 1993).

Acknowledgement

This work is partly supported by the Research Promotion Fund, 1997(DDP540), University of Essex. Jin Li is supported by Overseas Research Students Awards(ORS) and University of Essex Studentship. The DJIA index data was generously provided by Professor Blake LeBaron in University of Wisconsin - Madison.

REFERNECES

Alexander, S.S., *Price movement in speculative markets: Trend or random walks, No. 2,* in Cootner, P. (ed.), the random character of stock market prices, MIT Press, Cambridge, MA, 1964, 338-372.

Angeline, P. & Kinnear, K.E., (ed.), *Advances in genetic programming II*, MIT Press. 1996.

Bäck, T., *Evolutionary algorithms in theory and practice*, New York: Oxford University Press, 1996.

Bäck, T., (ed.), *Proceedings of the seventh international conference on genetic algorithms*, San Francisco, California: Morgan Kaufmann Publishers, Inc., 1997.

Bauer, R. J. Jr. , *Genetic Algorithms and Investment Strategies.* New York, John Wiley & Sons, Inc., 1994.

Brock, W., Lakonishok, J. & LeBaron, B., Simple technical trading rules and the stochastic properties of stock returns, *Journal of Finance*, 47, (1992), 1731-1764.

Butler, J.M., Eddie beats the market, data mining and decision support through genetic programming, *Developments*, Reuters Limited, (1997), Vol1.

Campbell, J.Y., Lo, A.W. & MacKinlay, A.C., *The econometrics of financial markets*, Princeton, N.J.: Princeton University Press, 1997.

Chen, S-H. & Yeh, C-H., Toward a computable approach to the efficient market hypothesis: An application of genetic programming, *Journal of Economic Dynamics and Control*, 21, (1996), 1043-1063.

Fama, E.F. & Blume, M.E., filter rules and stock-market trading, *Journal of Business* 39(1), (1966), 226-241.

Fama, E.F., Efficient capital markets: A review of theory and empirical work, *Journal of Finance* 23, (1970), 383-417.

Fogel, L.J., Owens, A. & Walsh, M., *Artificial Intelligence Through Simulated Evolution*. New York: John Wiley & Sons, 1966.

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

Goonatilake, S. & Treleaven, P. (ed.), *Intelligent systems for finance and business*, Wiley, New York, 1995.

Holland, J.H., *Adaptation in natural and artificial system*, University of Michigan Press, 1975.

Jegadeesh, N, Evidence of predictable behaviour of security returns, *Journal of Finance* 45, No 3, (1990), 881-898.

Jegadeesh, N. & Titman, S., Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency, *Journal of Finance*, 48, No.1, (1993), 65-91.

Kinnear, K.E. (ed.), *Advances in genetic programming*, MIT Press, 1994.

Koza, J.R., *Genetic Programming: on the programming of computers by means of natural selection*. MIT Press, 1992.

Koza, J., Goldberg, D., Fogel, D. & Riolo, R. (ed.), *Procedings of the First Annual Confrence on Genetic programming*, MIT Press, 1996.

Lau, T.L. & Tsang, E.P.K., Solving the processor configuration problem with a mutation-based genetic algorithm, *International Journal on Artificial Intelligence Tools* (IJAIT), World Scientific, Vol.6, No.4, (1997), 567-585.

Lehmann, B.N., Fad, martingales, and market efficiency, *Quarterly Journal of Economics*, 105, (1990), 1-28.

Lo, A.W. & MacKinlay, A.C., When are contrarian profits due to stock market overreaction? *Review of Financial Studies* 3, (1990), 175-206.

Lukac, L.P., Brorsen, B.W. & Irwin, S.H., A test of futures market disequilibrium using twelve different technical trading systems, *Applied Economics*, 20, (1988), 623-639.

Mahfoud, S. & Mani, G., Financial Forecasting Using Genetic Algorithms, *Journal of Applied Artificial Intelligence* Vol.10, Num 6, (1996), 543-565.

Malkiel, B., Efficient market Hypothesis, in Newman,P., Milgate, M. and Eatwell, J.(eds.), *New Palgrave Dictionary of Money and Finance*, Macmillan, London, (1992), pp739.

Miller, B.L. & Goldberg, D.E., Genetic Algorithms, tournament selection, and the effects of Noise. IlliGAL Report No. 95006. 1995.

Mitchell, M., *An introduction to genetic algorithms*, MIT Press, 1996.

Neely, C., Weller, P. & Ditmar, R., Is technical analysis in the foreign exchange market profitable? a genetic programming approach, in Dunis, C. & Rustem, B.(ed.), *Proceedings, Forecasting Financial Markets: Advances for Exchange Rates, Interest Rates and Asset Management*, London. 1997.

Neftci, S.N., Naïve trading rules in financial markets and Wiener-Kolmogorov prediction theory: A study of 'technical analysis' , *Journal of Business*, 64, (1991), 549-571.

Oussaidene, M., Chopard, B., Pictet, O. & Tomassini, M., Practical aspects and experiences - Parallel genetic programming and its application to trading model induction, *Journal of Parallel Computing* Vol. 23, No. 8, (1997), 1183-1198.

Rechenberg, I., *Evolutionsstratrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Stuttgart: Frommann-Holzboog, 1973.

Schwefel, H.-P., *Numerical optimisation of computer models*, Wiley, Chichester, 1981.

Sweeney, R. J., Some new filter rule test: Methods and results, *Journal of Financial and Quantitative Analysis*, 23, (1988), 285-300.

Tsang, E.P.K., Li, J. & Butler, J.M., EDDIE beats the bookies, *International Journal of Software, Practice & Experience*, Wiley, Vol.28 (10), 1033-1043, August 1998.

Tsang, E.P.K. *Foundations of constraint satisfaction*, Academic Press, London, 1993.

Werner, F.M., Bondt, D. & Thaler, R., Further Evidence on Investor Overreaction and Stock Market Seasonality, *Journal of Finance*, 42, no. 3 , (July 1987), 557-581.