Proceedings, Congress on Evolutionary Computation, Washington DC, USA, 6-9 July 1999

# Investment Decision Making Using FGP: A Case Study

**Jin Li and Edward P. K. Tsang**
Department of Computer Science
University of Essex
Wivenhoe Park, Colchester
United Kingdom
{jli, edward}@essex.ac.uk

**Abstract- Financial investment decision making is extremely difficult due to the complexity of the domain. Many factors could influence the change of share prices. FGP (Financial Genetic Programming) is a genetic programming based forecasting system, which is designed to help users evaluate impact of factors and explore their interactions in relation to future prices. Users channel into FGP factors that they believe are relevant to the prediction. Examples of such factors may include fundamental factors such as "price-earning ratio", "inflation rate" or/and technical factors such as "5-days moving average", "63-days trading range breakout", etc. FGP uses the power of genetic programming to generate decision trees through combination of technical rules with self-adjusted thresholds. In earlier papers, we have reported how FGP used well-known technical analysis rules to make investment decisions. This paper tests the versatility of FGP by testing it on shorter-term investment decisions. To evaluate FGP more thoroughly, we also compare it with C4.5, a well-known machine learning classifier system. We used six and a half years' daily closing price of the Dow Jones Industrial Average (DJIA) index for training and over three and half years' data for testing and obtained favourable results for FGP.**

## 1 Introduction

Financial investment decision making in the stock market is extremely difficult due to the inherent complexity of the domain. Many factors could affect the future prices. For example, the future price of a share may be influenced by fundamental factors, such as "price-earning ratio", "inflation rate", as well as technical factors such as "n-days moving average", "n-days trading range breakout", etc. Other influential factors may include market indices, who said what in public, etc. Prediction is made more difficult by the fact that various factors often interact with each other. To help users evaluate the impact of different factors and explore their interactions in relation to market movement, we have developed a genetic programming based system FGP (Financial Genetic Programming).

Genetic Programming (GP) (Koza 1992, Angeline & Kinnear 1996) is a promising variant of genetic algorithms (GAs) that uses tree instead of string representations. Genetic algorithms have been studied in financial markets for quite some time. Bauer (1994) reported his GAs intelligent systems that aimed at finding tactic market timing strategies. Allen & Karjalainen (1995) used genetic programming to identify profitable trading rules in the stock market. By conducting experiments using genetic programming, Chen & Yeh (1996) attempted to formalise the notion of unpredictability (as in the efficient market hypothesis) in terms of search intensity and the chance of success. 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. In our earlier work (Butler 1997; Tsang et al.1998; Li & Tsang 1999), we reported a genetic programming based system for predicting whether a return of r or more could be achievable within the next n trading days in a stock market. Specifically, we tested n = 63 (which is equivalent of 3 months) and r = 4% over ten years' data on the S&P 500 and Dow Jones Industrial Average (DJIA) indices. In both indices, FGP was able to generate rules that outperformed random runs and commonly used technical rules, in terms of prediction accuracy (to be elaborated below) and average annualised rate of return. The work in this paper is to extend our earlier work in following aspects:

1) To test the versatility of FGP by asking it to make investment decisions over a relatively shorter period of time. Specifically, we ask FGP to recognize investment opportunities where a return of 2.2% or more can be achieved within 21 trading days (i.e. one month). We would like to test the effectiveness of FGP when r and n take on different values.
2) To compare FGP with other machine learning system. Specifically, we compare FGP to C4.5 (Quinlan 1993), a well-known machine learning classifier system. Such comparison is highly relevant because like C4.5, FGP generates decision trees.

Procedure **GA**

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

**Figure 1** ***A simple genetic algorithm***

In next section we shall explain how FGP works. Section 3 describes the experimental data and performance criteria that we adopt. Section 4 reports comparison results among random runs, FGP and C4.5. Finally, we draw conclusion.

## 2 Background OF FGP

FGP is based on Genetic programming (GP), which belongs to the paradigm of genetic algorithms (GAs) (Holland 1975 and Goldberg 1989). In GA, a candidate solution is represented by a string. In GP, a candidate solution is represented by a tree. 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 prediction 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, such as reproduction, crossover, mutation) are used to combine or change the selected candidate solutions in order to generate offspring, which will form the population in the next generation. The basic operations of GAs are summarised in Figure 1.

In FGP, a candidate solution is represented by a genetic decision tree (GDT). The basic elements of GDTs are conditions and recommendations. A single condition comprises one financial indicator, such as 50-days moving average, one relational operator such as "greater than", or "less than", etc, and a threshold (a number). Conditions are combined in GDTs through logic operators such as "Or", "And", "Not", and "If-Then-Else". In this paper, we ask FGP to recognize investment opportunities where a return of 2.2% or more can be achieved within the next 21 trading days. Recommendation in this application is either positive (which suggests that a return of 2.2% or more can be achieved within the next 21 trading days) or negative (otherwise). Indicators in rules should be relevant to decision problem at hand. We limit ourselves to technical indicators that are derived from finance literature (see, e.g., Alexander1964, Sweeney 1988, Brock et al. 1992, and Fama & Blume 1966). They comprise the following three types of technical indicators (derived from three trading rules, namely moving average rules, filter rules, and trade range break rules in the literature).

(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

```
S ::= <PatternTree>;
<PatternTree> ::= "If-then-else" <Condition> <ThenBranch> <ElseBranch>;
<Condition> ::= <Condition> "And" <Condition> | <Condition> "Or" <Condition> |
                "Not" <Condition> | <Indicator> <RelationOperation> <Threshold> ;
<ThenBranch> ::= <PatternTree> | <Recommendation> ;
<ElseBranch> ::= <PatternTree> | <Recommendation> ;
<RelationOperation> ::= ">" | "<" | "=" ;
<Indicator> ::= "MV_12" | "MV_50 " | "Filter_5" | "Filter_63"| "TRB_5 " | "TRB_50";
<Threshold> ::= Real Number;
<Recommendation> ::= "Positive" | "Negative";
```

**Figure 2** ***The BNF grammar that FGP uses for constructing GDTs***

(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

The syntax used in FGP to build GDTs can be precisely described by using the backus normal form (BNF) grammar (Backus, 1959), as shown in Figure 2.

Figure 3 shows an example of a simple GDT built by using the above grammar. A useful GDT in the real world may be a lot more sophisticated than this.

```
(IF (PMV_50 < -18.45) THEN Positive
ELSE (IF ((TRB_5 > -19.48) AND (Filter_63 < 36.24))
   THEN Negative
   ELSE Positive))
```

**Figure 3** *A (simplistic) GDT for decision making*

This rule suggests that if today's price is 18.45 or more below the average price of the last 50 days, then the goal can be achieved if we invest today (we call today a positive position). Otherwise decision depends on the values of TRB_5 and Filter_63. If today's price is no more than 19.48 above the maximum price of the previous 5 trading days or today's price is more than 36.24 above the minimum price in the last 63 days, then it is also an alternative good opportunity to make a buy decision. The search space for GDTs is enormous. One has to search in the space of indicators, operators as well as thresholds (such as 36.24). The hope is that GP will explore this search space efficiently.

FGP maintains a population (a set) of GDTs and works in iterations. Initially, each GDT is generated randomly. In each iteration, FGP creates a new generation of population using standard genetic operators, including crossover, mutation and reproduction. Given two GDTs (called parents), the crossover operator in FGP works as follows:

a) FGP randomly selects a node within each parent GDTs as a crossover point;
b) To generate two children, FGP exchanges the subtrees rooted at the selected nodes.

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

a) FGP randomly selects a node x within a GDT as the mutation point; x can be an internal node as well as a terminal node;
b) FGP replaces the subtree below x with a new tree with a limited depth.

In general, we employ a high crossover rate (e.g. 0.9) and a low mutation rate (e.g. 0.01). Besides, reproduction is used in the process (e.g. with probability 0.1) to increase the number of occurrences of individual GDTs with higher fitness. FGP provides the users with the choice of two selection strategies, namely roulette wheel and tournament. We used the latter in our tests reported in this paper.

The evolutionary process in FGP is driven by a fitness function. It evaluates each GDT, and assigns to it a fitness value, which reflects the quality of the GDT to solve the problem at hand. Based on whether the given goal can be achieved, any given day is classified into either a positive position (goal is achievable) or a negative position (goal cannot be achieved). In this paper, we use the Rate of Correctness (RC), which we refer to as prediction accuracy, as the fitness function.

$$\text{RC} = \frac{P + N}{T}$$

where

P = The number of positive positions predicted correctly (i.e. the number of times that FGP recommends "positive" on a positive position);
N = The number of correct negative positions predicted;
T = The total number of predictions made.

## 3 Experimental Index Data and Performance Criteria

Available to us is the closing prices of the Dow Jones Industrial Average (DJIA) Index from 7 April 1969 to 5 May 1980, which includes 2,800 data cases. We took the index data from 7 April 1969 to 11 October 1976 (1,900 cases) as training data, and took the index data from 12 October 1976 to 5 May 1980 (900 data cases) as test data. The whole data series can be visualised in Figure 4. The whole training data and test data contain roughly 50% of positive positions. Our prediction problem is essentially a two-classification problem, to which the classifier system C4.5 is applicable.

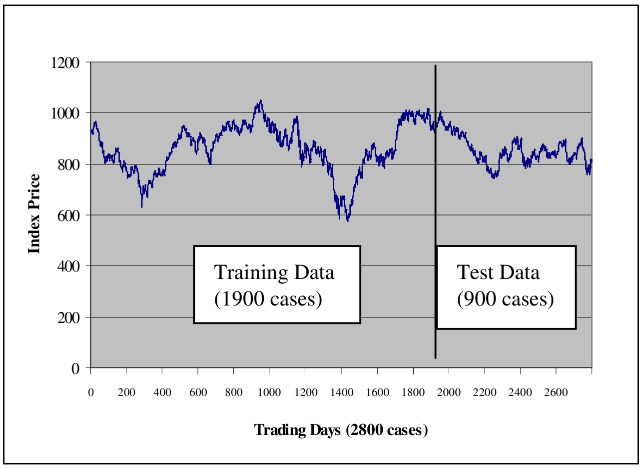


**Figure 4** *Training and test DJIA index series*

| | |
|---|---|
| Objective | To find FGP rules with high accuracy |
| Input terminals (six technical indicators and real values) | $I_{MV\_12, t}$, $I_{MV\_50, t}$ , $I_{TRB\_5, t}$ , $I_{TRB\_50, t}$ , $I_{Filter\_5, t}$ , $I_{Filter\_63, t}$ and Real values as thresholds. |
| Prediction terminals | {0, 1}, with 1 representing "Positive"; 0 representing "Negative" |
| Non-terminals | If-then-else , And, Or, Not, , > ,<., = |
| Data | Total data cases: 2800 (07/04/1969 to 05/05/1980)<br>The training data cases: 1900 (07/04/1969 to 11/10/1976)<br>The test data cases : 900 (12/10/1976 to 05/05/1980) |
| Fitness function | RC = no. of correct decisions over total no. of decisions made |
| Crossover rate | 0.9 |
| Mutation rate | 0.01 |
| Parameters | Population size = 1,200; maximum no. of generations = 30 |
| Termination criterion | The maximum number of generations has been run or FGP programme has run for more than 2 hours. |
| Selection strategy | Tournament selection, Size = 4 |
| Max depth of individual program | 17 |
| Max depth of initial individual program | 4 |
| Run times (hours) | 1-2 |
| Hardware and operating system | Pentium PC 200MHz running Windows 95 with 64M RAM |
| Software | Borland C++ (version 4.5) |

**Table 1** ***Tableau for FGP experiments on DJIA data***

RC was used as the fitness function in FGP, hence it should be used to evaluate how well FGP does its job. For reference, as well as for practical reasons, we would also like to have some way of measuring the return should the GDTs generated be used for investment decisions. The point is, even if the predictions are accurate, the actual return depends on how we use the predictions. This depends on our trading behaviour. For simplicity, we use the following hypothetical trading behaviour:

*we assume that when a positive position is predicted, one unit of money was invested in a portfolio reflecting the DJIA index. If the DJIA index does rise by 2.2% or more at day t within the next 21 trading days, then we sell the investment at the index price of day t. If not, we sell the investment on the 21st day, regardless of the price.*[1]

For simplicity, we ignore transaction cost. We define two investment performance criteria, namely the average annualised rate of return (AARR) and rate of positive return (RPR) for evaluating the performance of the GDTs based on the above trading behaviour. An annualised rate of return (ARR) is defined as follows:

$$\mathrm{ARR}i = \frac{253}{t} * \frac{P_t - P_0}{P_0}$$

where $P_0$ is the buy price, $P_t$ is the sell price. $t$ is the number of days that the investment is held, 253 is total number of trading days in one calendar year.

Therefore, for one GDT which generates $N_+$ number of predictive positive positions, its average ARR is:

$$\mathrm{AARR} = \frac{1}{N_+} \sum_{i=1}^{N_+} ARR_i .$$

Even if the goal cannot be achieved, we would like to know how often a GDT recommends investments that cause the investor to lose money. For this purpose, we define RPR, the proportion of positive returns to the total number of positive recommendations:

$$\mathrm{RPR} = \frac{1}{N_+} \sum_{i=1}^{N_+} \mathrm{I}_i \quad \text{where } \mathrm{I}_i = \begin{cases} 1 & \text{if } ARRi > 0 \\ 0 & \text{otherwise} \end{cases}$$

It is worth re-iterating that RC should be the main criterion for evaluating the performance of both FGP and C4.5 because it is what FGP and C4.5 was asked to maximize. AARR and RPR should only be used for reference.

1 A better approach would be to run a version of FGP to predict when it is a good time to buy, and another version which recommends when it is a good time to sell..

| Numbers | FGP GDT Results | | | Random Run Results | | | C4.5 Ruleset Results | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | RC | AARR | RPR | RC | AARR | RPR | -c | RC | AARR | RPR |
| 1 | 55.44% | 45.01% | 61.77% | 50.44% | 35.48% | 56.52% | 1 | 49.44% | 37.36% | 55.83% |
| 2 | 54.67% | 43.56% | 62.61% | 48.22% | 35.75% | 58.30% | 5 | 52.11% | 40.22% | 57.14% |
| 3 | 55.67% | 47.75% | 66.01% | 50.00% | 35.51% | 56.58% | 10 | 52.89% | 40.93% | 58.81% |
| 4 | 54.44% | 45.13% | 61.07% | 50.89% | 40.51% | 57.83% | 25 | 54.89% | 47.46% | 60.78% |
| 5 | 54.44% | 43.29% | 60.22% | 46.44% | 33.36% | 54.16% | 50 | 54.67% | 44.22% | 59.92% |
| 6 | 54.00% | 47.40% | 62.50% | 53.67% | 39.68% | 59.57% | 75 | 54.67% | 44.92% | 60.17% |
| 7 | 55.33% | 47.67% | 62.06% | 49.78% | 37.94% | 57.25% | 100 | 55.11% | 46.08% | 62.05% |
| 8 | 54.78% | 47.35% | 62.24% | 46.67% | 32.54% | 55.98% | | | | |
| 9 | 55.00% | 46.53% | 61.00% | 48.67% | 35.24% | 59.06% | | | | |
| 10 | 54.00% | 46.99% | 61.79% | 50.56% | 42.35% | 57.47% | | | | |
| Highest | 55.67% | 47.75% | 66.01% | 53.67% | 42.35% | 59.57% | | 55.11% | 47.46% | 62.05% |
| Lowest | 54.00% | 43.29% | 61.00% | 46.44% | 32.54% | 54.16% | | 49.44% | 37.36% | 55.83% |
| Mean | **54.78%** | **46.07%** | **62.13%** | **49.53%** | **36.84%** | **57.27%** | | **53.40%** | **43.03%** | **59.24%** |
| STD | 0.58% | 1.70% | 1.6% | 2.14% | 3.18% | 1.6% | | 2.08% | 3.61% | 2.15% |

**Table 2** ***Performance of random decisions, FGP and C4.5 on DJIA index, 12/10/1976 to 5/5/1980 (900 data cases)***

## 4 Experimental Results

In our experiments, we ran FGP 10 times. The termination condition was set to 2 hours on a Pentium PC (200 MHz) or 30 generation, whichever reached first. Main parameters of the experiments are displayed in Table 1. For each run, a GDT generated, based on the training data, was applied to the test data. The results of applying the 10 GDTs on the test data were recorded in Table 2. To assess the quality of FGP results, we compare its results with those of random decisions and results generated by C4.5, both of which are also reported in Table 2.

According to the efficient market hypothesis (EMH) (Malkiel 1992), stock prices follow a random walk behaviour and therefore no trading rules could out-perform random decisions. Our empirical results do demonstrate that there is some predictability in the DJIA index based on historical data alone. For GDTs, the mean RC, AARR, and RPR are 54.78%, 46.07% and 62.13% respectively. These are much higher than the mean RC (49.53%), AARR (36.84%) and RPR (57.27%) achieved by the 10 random decisions. In fact, even the poorest results of 10 GDTs (54.00%, 43.29%, and 61.00% for RC, AARR, and RPR respectively) is better than the best results of the 10 random runs under each of three criteria used here (53.67%, 42.35%, and 59.57% respectively). Results here are consistent with our results achieved in the past (Tsang et al.1998; Li & Tsang 1999), which shows that FGP is capable of out-performing random decisions under any one of above three criteria.

C4.5 is one of the most commonly used decision tree learning classifier systems, which was developed by Quinlan (1986, 1993). Both FGP and C4.5 take the same type of input (training examples) and generate decision trees, which C4.5 converts to rulesets that is more easily understood by people. We fed C4.5 with the six technical rule indicators that we used for FGP. We ran C4.5 system on the same training data and applied the rulesets that it generated to the test data. Following is an example of a single rule generated by C4.5:

If (PMV_50 > -33.075) And (PMV_50 <= -28.0292) And
(TRB_50 <= -69.15 And (Filter_5 > -0.26)
Then Positive Position

A parameter that significantly affects the performance of the rulesets generated by C4.5 is the "certainty factor" (run with -c CF), which ranges from 0 to 100.[2] The certainty factor is used to controls pruning, details of which will not be elaborated here. The value -c 25 represents default pruning in C4.5. Small values usually lead to small rulesets, whereas large values imply less pruning and therefore large rulesets. Showed in Table 2 under the row of "C4.5 Ruleset Results" are -c options with seven different CF values and their performances of the corresponding rulesets generated. Mean results for RC, AARR, and RPR are 53.40%,

[2] Other parameters can be used to run C4.5. These could potentially influence the performances of rule-sets generated. These parameters are "confidence" (-F) and "redundancy" (-r). According to our experience, using confidence values other than the default, such as 20, 10, 5, 1, and 0.1, made no difference to C4.5's performance on our data set. Quinlan (1993, p88) pointed out that the setting of the redundancy parameter is only beneficiary if the user knows the data well and can estimate appropriate values. We have no knowledge to guide us on setting the redundancy value. However, to allow a fair comparison, we experimented with redundancy values of 2.0, 3.0 and 4.0 in addition to the default value (1.0). We observed no results better than those reported in Table 2.

| Groups | FGP Vs C4.5 | | | C4.5 Vs Random Runs | | |
|---|---|---|---|---|---|---|
| Criteria | For RC | For AARR | For RPR | For RC | For AARR | For RPR |
| *t* values | 2.0133 | 2.3406 | 3.2201 | 3.7003 | 3.7379 | 2.1883 |
| *p* values | 0.0312 | 0.0167 | 0.0029 | 0.0011 | 0.0009 | 0.0224 |
| *df* (degrees of freedom) | 15 | 15 | 15 | 15 | 15 | 15 |

**Table 3** ***t-statistics for comparing mean performances of two groups (FGP versus C4.5 and C4.5 versus Random Runs)***

43.03%, and 59.24%, each of which is lower than the corresponding mean result of GDTs, but higher than the mean result of random runs respectively.

To determine whether result differences are statistically significant, the statistical one-tailed unpaired t-test can be applied on the null hypothesis that mean performances of two groups are not statistically different under each of the three criteria. Showed in Table 3 are test results for both comparisons of FGP versus C4.5 and C4.5 versus random runs. p values for comparing FGP and C4.5 sample groups are 0.0312, 0.0167 and 0.0029 for RC, AARR, and RPR respectively. Notably, all three p values are less than 0.05, which means results of FGP are at least better than those of C4.5 in terms of any one of three criteria at the conventional statistical significant level (p=0.05). Meanwhile, p values for comparing C4.5 and random runs demonstrate that C4.5 outperforms random runs with more than 95% statistical confidence level in terms of any three criteria. It is encouraging that both FGP GDTs and C4.5 Rulesets seem to grasp plausible hidden patterns in financial data as to achieve better performances that cannot explained by random decisions. More important is that our FGP outperforms C4.5 statistically significantly in this case. Poor performances of C4.5 may be contributed to its overfitting problem. On training data, the results of rulesets are much higher than results of GDTs (both results are not showed in this paper). This means rulesets are too overfitting on training data to be as good as GDTs on test data.

## 5 Conclusion and Further Work

In this paper, FGP takes some well-known technical analysis rules as input indicators to predict whether one can achieve 2.2% or more within 21 trading days. Using technical analysis rules for financial prediction is supported by a fairly quantity of finance studies (e.g. Lukac et al. 1988, Neftci 1991, Brock et al. 1992, Campbell 1997). Consistent with the findings in our earlier work, results in this paper demonstrate that even over a shorter period (21 trading days versus 63 trading days in our earlier paper), FGP still reliably generated accurate GDTs. Though the number of runs for FGP is small in this paper, results are consistent and statistically significant that FGP outperforms C4.5 in terms of any criteria including one mainly concerned prediction performance of RC and two reference investment performances of AARR and RPR.

So far, our consistent and encouraging results have demonstrated that FGP is a promising genetic programming based system that is worth further investigation and development. We are exploring the possibility of incorporating constraint satisfaction techniques (Tsang 1993, Lau & Tsang 1997) into FGP to improve its capability as a forecasting tool. More specifically, we are now advancing the fitness function in FGP to generate specific rules which are able to meet different demands for investors with different investment attitudes.

## Acknowledgments

Jin Li is supported by an Overseas Research Students Award (ORS) and a University of Essex Studentship. The DJIA index data was generously provided by Professor Blake LeBaron in University of Wisconsin - Madison. Thanks to Colin Ho and Tung-Leng Lau for valuable discussions, especially their advice on C4.5. The anonymous referees provided valuable comments.

## Bibliography

Alexander, S.S., (1964), “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, 338-372.

Allen, F. & Karjalainen, R., (1995), “Using genetic algorithms to find technical trading rules,” Working paper at Rodney L. White Centre for Financial Research.

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

Backus, J.W., (1959), “The syntax and semantics of the proposed international algebraic language of Zurich”, ACM-GAMM conference, ICIP, Paris, June.

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

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

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

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

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

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

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

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

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

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

Li, J. & Tsang, E.P.K, (1999), "Improving technical analysis predictions: an application of genetic programming," in *proceedings of 12th Annual Florida Artificial Intelligence International Research Conference* (FLAIRS-99), Orlando, Florida, USA, May 3-5, to appear.

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

Mahfoud, S. & Mani, G., (1996), "Financial forecasting using genetic algorithms," *Journal of Applied Artificial Intelligence*, Vol.10, Num 6, 543-565.

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

Neely, C., Weller, P. & Ditmar, R., (1997), "Is technical analysis in the foreign exchange market profitable? a genetic programming approach," *Journal of Financial and Quantitative Analysis*, 32, 405-26.

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

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

Quinlan, J.R., (1986), "Induction of decision tree," *Machine Learning,* 1, 81-106.

Quinlan, J.R., (1993), *C4.5: Programs for machine learning,* Morgan Kaufmann, San Mateo, CA.

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

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

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