

Taiwan Stock Forecasting with the Genetic Programming*

Siao-Ming Jhou, Chang-Biau Yang[†] and Hung-Hsin Chen

Department of Computer Science and Engineering

National Sun Yat-sen University

Kaohsiung 80424, Taiwan

Abstract—In this paper, we propose a model for generating profitable trading strategies for Taiwan stock market. Our model applies the genetic programming (GP) to obtain profitable and stable trading strategies in the training period, and then the strategies are applied to trade the stock in the testing period. The variables for GP include 6 basic information and 25 technical indicators. We perform five experiments on Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) from 2000/9/14 to 2010/5/21. In these experiments, we find that the trading strategies generated by GP with two arithmetic trees have more stable returns. In addition, if we obtain the trading strategies in three historical periods which are the most similar to the current training period, we are able to earn higher return in the testing period. In each experiment, 24 cases are considered. The testing period is rolling updated with the sliding window scheme. The best cumulative return 166.57% occurs when 545-day training period pairs with 365-day testing period, which is much higher than the buy-and-hold strategy.

Keywords—stock; Taiwan Stock Exchange Capitalization Weighted Stock Index; genetic programming; annualized return; feature set.

I. INTRODUCTION

Predicting the fluctuation of the stock market is a very popular research. Many investors are interested in investment, but forecasting the movement of the stock market index is very difficult, because the stock market is usually affected by many external factors, such as the interaction of international financial markets, the political factors, the human operations, etc. Generally, investors do not have sufficient knowledge and information about investment so that they cannot gain return easily.

To get the patterns of price fluctuation hidden in the stock market, some studies [1], [9], [10] investigated historical data to find a precise solution of the stock market volatility by using the methods of artificial intelligence and statistics. Furthermore, some studies tried to generate trading strategies for deciding the timing of buying and selling. The methods for forecasting the stock market fluctuation include *support vector regression* (SVR) [10], *support vector machine* (SVM) [5], [6], [9], [14], *genetic algorithm* (GA) [1], [3], [4], [5], [6], [7], [11], [16], [17], *genetic programming* (GP) [13], [19], *artificial neural network* (NN) [7], [10], [11], [12], [16], [18]. The goal of these studies is to mine the pattern of the price fluctuation and to improve the accuracy of predicting the trend of the stock market. Among these studies, three benchmarks

are usually used. The first is to predict whether the trend of the stock market is raising or falling. [5], [6], [9], [14], [16]. The second benchmark is the difference of predicted indices and the real ones. The most commonly used measurements are *root mean squared error* (RMSE) and *mean absolute percentage error* (MAPE) [2], [7], [10], [12], [18]. The third benchmark is to apply the machine learning techniques to mine the trading strategies from the historical data, and the return got by applying these strategies to trade one stock is usually compared with the *buy-and-hold* strategy [1], [3], [4], [13].

In this paper, our goal is to earn stable return in Taiwan stock market. The trading strategy is applied to Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) in the testing period. The total trading period is from 2000/9/14 to 2010/5/21, approximately 10 years in total. The best cumulative return 166.57% occurs when 545-day training period pairs with 365-day testing period, which is much higher than the return of the buy-and-hold strategy 1.19%.

The rest of this paper is organized as follows. We will introduce the genetic programming (GP) in Section II. In Section III, we will present our method that the trading strategies are generated by GP. In Section IV, we will show our experimental results. Finally, in Section V, we will give the conclusion and some future works.

II. PRELIMINARIES

In this section, we will give an introduction of the genetic programming, which is the background knowledge of this paper. The *genetic programming* (GP), which is extended from the *genetic algorithm* (GA), was proposed by Koza at 1992 [8]. The great advantage of GP is that it can be applied in varieties of problems with some constraints. The solutions of these problems generated by GP are represented as formulas. The representation of the chromosome of GP is more flexible than GA. The solutions of GA are string structures with fixed length which need to be encoded and to be decoded for answer transformation. The tree structure of GP has dynamic extensibility. GP can parse the tree structure to get the corresponding solution. Therefore, GP is more suitable for searching trading strategies. One trading strategy is usually produced by predefined functions, constants and variables.

The evolutions in GP are summarized as follows.

- 1) *Initialization*: The initial population is composed of individuals, which are generated randomly. The structure of an individual is a tree.

*This research work was partially supported by the National Science Council of Taiwan under contract NSC99-2221-E-110-048.

[†]Corresponding author: cbyang@cse.nsysu.edu.tw

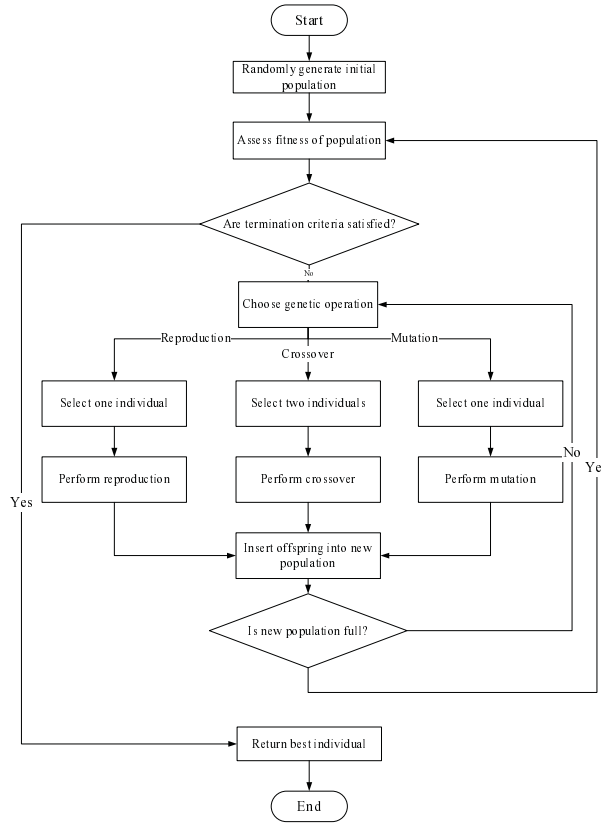


Figure 1. The flow chart of GP.

- 2) *Selection*: To select individuals with higher fitness values into the new population is usually accomplished by using the roulette-wheel method.
- 3) *Reproduction*: Reproduction is to copy the elitist individual into the new population.
- 4) *Crossover*: The most common approach for crossover in GP is the *subtree crossover*. The internal nodes are usually selected as crossover points. To perform the crossover between two trees, random subtrees are respectively selected from the father tree and the mother tree. Then, the father's subtree is replaced by the mother's subtree, and the new individual is the offspring.
- 5) *Mutation*: Mutation is applied to only one individual, and it increases the diversity of the population. The most common approach for mutation is the *subtree mutation*. In the subtree mutation, a mutation point (the root of a subtree) is randomly selected from the individual. To perform the mutation, the subtree connected to the mutation point is replaced by a randomly generated tree. The new individual is the result of mutation.

In GP, the final answer is represented by a formula rather than some parameters. In other words, the solution space of searching is flexible, and it also improves the encoding problem of GA. Figure 1 shows the flow chart of GP. A *function node* (or internal node) in the GP tree is

usually an arithmetic operator, a mathematical function, or a Boolean operator. Various problems can be solved with GP by designing suitable function nodes. Before we apply GP to the solution of a certain problem, we have to do the following works first.

- 1) **To define the set of terminal nodes and function nodes**: One problem may need some specific terminal nodes and function nodes, and the definitions of these nodes should satisfy the closure property when the trees are parsed. In other words, we have to check whether the trees can be parsed for evaluating or not. Some commonly used function nodes are given as follows.

- a) arithmetic operators : $+$, $-$, \times , \div , etc.
- b) Boolean operators : *AND*, *OR*, *NOT*, etc.
- c) comparison operators : $>$, \leq , \geq , etc.
- d) logical operators : *if - then - else*.

Each *terminal node* (leaf node) represents a constant or a variable for a specified problem.

- 2) **To define the fitness function**: The fitness score of one chromosome is used to measure the goodness of the chromosome. We can eliminate the chromosomes with low scores and reserve the chromosomes with high scores.
- 3) **To set parameters for GP**: Some parameters of GP have to be set before they are operated. The parameters include population size, crossover rate, crossover mechanism, mutation rate, mutation mechanism, number of generations, and the maximum depth of the arithmetic tree.
- 4) **To determine the terminal condition**: The terminal condition generally depends on the generations, the types of operators, and the convergence of the fitness scores. When GP terminates, the solution is represented by the chromosome with the best fitness score in the final generation.

III. STOCK INVESTMENTS WITH GENETIC PROGRAMMING

In this section, we propose a model which is able to gain stable returns on trading stock by using the *genetic programming* to search the stable and profitable trading strategies.

A. Flow Chart of Investment

The flow chart for our model is shown in Figure 2. In our model, we use GP to train the stable and profitable trading strategy in 3 historical periods which are the most similar to the training period of the target stock, and then we trade the stock by using the trained strategy in the testing period. The historical period starts from 1995/1/1 to the day which is before the training period. The training period and the testing period are rolling updated until the end of the trading period. The sliding window scheme is shown in Figure 3.

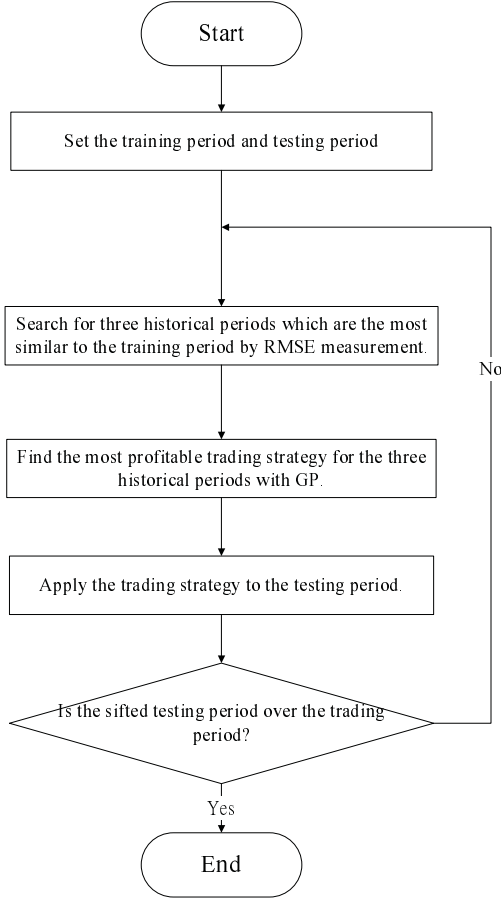
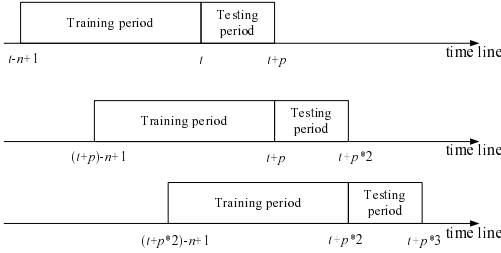


Figure 2. The flow chart for investing one stock.


 Figure 3. The training period and the testing period with the sliding window scheme. Here, n , t , and p represent the length (day) of the training period, the current date, and the length (day) of the testing period, respectively.

B. Training Interval Selection from Historical Data

Leu *et al.* [12] and Yu *et al.* [18] predict the stock close price by finding the historical period which is similar to the training period. In this paper, the *training period* is defined to be an interval just before the current *testing period*. The lengths of training and testing periods need not be the same. The *historical period* is selected from the interval starting from the very beginning to the day just before the training period. The goal of historical periods is to help the model construction of the training period. In our model, three historical periods which are the most

similar to the current training period are chosen from the historical data by the measurement of the root mean square error (RMSE). Here, all data series are formed by the daily close prices.

The equation of RMSE is given as follows:

$$RMSE(X_1, X_2) = \sqrt{\frac{\sum_{j=1}^n (x_{1,j} - x_{2,j})^2}{n}}, \quad (1)$$

where X_1 , X_2 , $x_{1,j}$, $x_{2,j}$, and n represent the data series of the first period, the data series of the second period, the j th element of the first period, the j th element of the second period, and the length of the period, respectively.

After we find the 3 historical periods with the lowest RMSE values compared to the training period, we extend these periods. That is, suppose that the lengths of the training and testing periods are $d1$ and $d2$, respectively. The length of the historical period we search is $d1$, and the length of the extended period is $d1 + d2$. Each strategy generated by GP is used to trade the stock in the 3 extended historical periods, then we can get 3 returns with their standard deviation. The higher the average return divided by its standard deviation is, the more stable and profitable the trading strategy is. After getting the best trading strategy, we apply it in the testing period.

C. Generating the Trading Strategy with the Genetic Programming

To utilize the genetic programming (GP), we have to define the function nodes and the terminal nodes first. The function nodes for our method consist of operators '>', '<', '=', '>=', '<=', 'logical and', 'logical or', '+', '-', '*', and 'x'. The terminal nodes involve 6 basic information and 25 technical indicators of one stock, which are open prices, close prices, highest prices, lowest prices, the trading volumes, the amount of prices, MTM^5 , OBV , DI , $volume^5$, $volume^{20}$, RSI^5 , RSI^{14} , MA^{10} , MA^{20} , $TAPI$, PSY^{14} , WMS^5 , WMS^9 , $BIAS^{10}$, $BIAS^{14}$, $BIAS^{20}$, OSC^5 , RSV^9 , K^3 , D^3 , EMA^{12} , EMA^{26} , DIF , and $MACD^9$. We denote these 31 features as *full features*. The constants for the terminal nodes are the numbers generated randomly between -1.0 and 1.0 . In addition, we modify the left subtree of the original arithmetic tree to a *buy-tree*, and the right subtree to a *sell-tree*. With this modification, the trading signal generated by two arithmetic trees on date t is given as follows.

```

IF (the value of the buy-tree on date t > 0)
  AND (the value of the sell-tree on date t < 0)
  THEN signal(t) ← TRUE
ELSE IF (the value of the buy-tree on date t < 0)
  AND (the value of the sell-tree on date t > 0)
  THEN signal(t) ← FALSE
ELSE
  signal(t) ← signal(t - 1)
END IF
    
```

(2)

According to the trading signal, the action we should take at date t is described as follows.

- **buying:** If the signal of date t is *TRUE* and the signal of date $(t-1)$ is *FALSE*, then we buy the stock.
- **selling:** If the signal of date t is *FALSE* and the signal of date $(t-1)$ is *TRUE* and we owned the stock, then we sell the stock.
- **holding:** If the signals of both dates $(t-1)$ and t are *TRUE*, then we hold the stock and wait the signal to turn to selling. Similarly, if the signals of dates t and $(t-1)$ are both *FALSE*, then we do nothing and wait the timing to buy.

When GP evolves a trading strategy in every generation, we can apply it to the 3 extended historical periods. Then, we get three returns, whose average and standard deviation can be calculated. Our fitness function of GP is to maximize the coefficient of return variation which is the average return divided by the standard deviation. In the end of evolution, we obtain a trading strategy which has the most stable and profitable return in the 3 extended historical periods. Then, we apply the strategy in the testing period. The training period and the testing period are rolling updated with the sliding window scheme.

IV. EXPERIMENTAL RESULTS

In this section, we will first introduce the benchmarks and the dataset, and then show our experimental results.

A. Benchmarks

In this section, we will introduce the measurements. Many measurements are able to evaluate the performance of a trading model, but they usually use different units or currencies. To get objective benchmark results, we use the *return on investment* (ROI), which is defined as $(\frac{\text{final capital}}{\text{initial capital}} - 1 \times 100\%)$. We also compare our results with the *buy-and-hold strategy* that the investor buys one stock at the start of the trading period, and sells the stock at the end of the trading period. A good trading model is able to earn high and stable return.

B. Data Collection and Preprocessing

Our dataset contains the basic information of *Taiwan Stock Exchange Capitalization Weighted Stock Index* (TAIEX), and the dataset is fetched from Taiwan Economic Journal (TEJ) database, which was established in April of 1990. It provides the stock information, including the prices, technical indicators, and fundamental analysis of securities of Taiwan stock market [15]. We assume that TAIEX is tradable. Our goal is to gain high and stable return by trading TAIEX with its daily close price being our trading target. The close prices of TAIEX from 1995/1/1 to 2011/3/1 are shown in Figure 4. In our experiments, the trading period is from 2000/9/14 to 2010/5/21. The reason we choose this period is that the starting day and the ending day has almost the same close price. In this period, it is fair to compare our results with the buy-and-hold strategy.

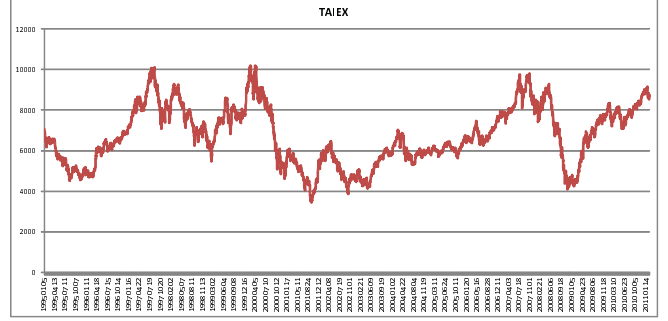


Figure 4. The close prices of TAIEX from 1995/1/1 to 2011/3/1.

C. Experimental results

In all experiments, short-term buying and selling is forbidden.

We perform five experiments described as follows.

- Experiment I: One arithmetic tree and the full feature set are used.
- Experiment II: Two arithmetic trees and the full feature set are used.
- Experiment III: We use two arithmetic trees and six selected features which are frequently used in the profitable strategies in Experiment II. These features are RSI^5 , PSY^{14} , WMS^5 , WMS^9 , DIF , and $MACD^9$.
- Experiment IV: We use two arithmetic trees and the full feature set with 12 macroeconomic indicators added, such as U.S. producer price index, U.S. annual changes in consumer price index, Taiwan unemployment rate, etc.
- Experiment V: Two arithmetic trees and the full feature set are used. We search the historical data to find the three periods which are the most similar to the current training period with the RMSE measurement. Then, the most profitable trading strategy for these three periods are trained by GP. And finally, we apply the strategy to the testing period.

In all experiments, it is assumed that TAIEX is tradable and only one unit is traded. Meanwhile, we can sell only when we have one unit in hand (bought before) The trading fee is assumed to be 0.6% (the current fee in Taiwan stock market).

We train the trading strategy by using GP with training periods of various lengths, including 90, 180, 270, 365, 455, 545, 635, and 730 days. And testing periods of various lengths, including 90, 180, and 365 days, with the sliding window scheme, are tested. Thus, totally 24 cases are performed in each experiment.

The function nodes include '>', '<', '=', '>=', '<=', 'logical and', 'logical or', '+', '-', and 'x'. The terminal nodes involve 6 basic information and 25 technical indicators, and the constants are random numbers between -1.0 and 1.0. The parameters of GP are shown in Table I.

In Experiment I, using only one arithmetic tree may generate trading signals which are very sensitive to the fluctuation of the market, and it may also cause many

Table I
THE PARAMETERS OF THE GENETIC PROGRAMMING.

Population size	100
Number of generations	200
Initial method	equal mix of Full and Grow method
Selection method	roulette wheel
Crossover rate	90%
Mutation rate	50%
Maximum depth	3

Table II
THE CUMULATIVE RETURNS FROM 2000/9/14 TO 2010/5/21 WITH TWO ARITHMETIC TREES FOR GP AND THREE HISTORICAL TRAINING PERIODS.

Training \ testing period	90	180	365	Avg.	Stdv.
90	61.27%	53.61%	68.44%	61.11%	7.41%
180	65.69%	66.76%	0.35%	44.27%	38.04%
270	-8.64%	-2.30%	109.28%	32.78%	66.33%
365	-19.52%	4.26%	56.52%	13.75%	38.90%
455	68.88%	27.06%	75.71%	57.22%	26.34%
545	60.41%	126.72%	166.57%	117.90%	53.62%
635	73.61%	38.70%	145.88%	85.96%	54.50%
730	21.66%	78.56%	61.72%	53.98%	29.23%
Avg.	40.42%	49.17%	85.52%		
Stdv.	37.30%	42.21%	53.17%		

unnecessary transactions, hence it did not earn high return. In Experiments II, III, and IV, two arithmetic trees with different feature sets are used for training trading strategies, but the training periods in these experiments are not highly related with the testing period, hence it is hard for GP to train a profitable trading strategies.

Among the five experiments, the fifth is the most profitable. The cumulative returns of the experiment are shown in Table II. As one can see, the cumulative returns of all cases are almost positive, it means that our method has only small probability to lose money. In the experiment, when the 545-day training period pairs with 365-day testing period, we can get the highest cumulative return 166.57%, which is higher than the return of the buy-and-hold strategy 1.19%. We show the trading record in Figure 5 in 2001 and trading strategies in Table III in the ten years. In Figure 5, we can buy TAIEX at relatively low price and sell it at relatively high price in the bull market, such as the 2nd buying time, 7th buying time, 2nd selling time, and 7th selling time. In the bear market, we can sell the stock before the stock crashes, such as 2nd, 3rd, and 6th selling time. It reveals the profitability of our model.

V. CONCLUSION

In this paper, we propose a novel model for investment in Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX). Our trading strategies are generated by two arithmetic trees in GP. When 545-day training period pairs with 365-day testing period, the cumulative return is 166.57%, higher than the buy-and-hold strategy.

In the future, we will try to find more suitable strategies by searching more periods from the historical data, and modify the evaluating method for finding similar patterns. The longest common subsequence or dynamic time warping is one of the possible ways to find the most similar subsequences in our training data for GP. We will also apply the risk management and the portfolio to the stock investment.

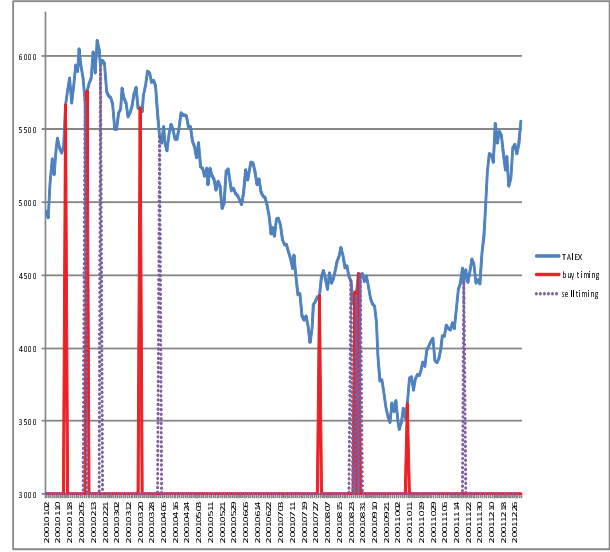


Figure 5. The trading record from 2001/1/1 to 2001/12/31 with 545-day training period and 365-day testing period in TAIEX. Here, a vertical solid line represents buying time and a vertical dotted line represents selling time.

Table III
THE TRADING STRATEGIES FROM 2000/9/14 TO 2010/5/21 WITH 545-DAY TRAINING PERIOD AND 365-DAY TESTING PERIOD IN TAIEX.

testing period	return	trading count	arithmetic trees	trading strategies
000914*010913	16.61%	9	buy rule sell rule	RSV^9 $(0.5 \geq PSY^{14}) \wedge ND(-0.1 < BIAS^{14})$
010914*020913	3.96%	11	buy rule sell rule	$(highp - (-0.3)) - (MA^{14})$ $MACD^9 > ((D^3)OR(MACD^9))$
020914*030913	29.89%	18	buy rule sell rule	$MA^{10} \times ((0.6)OR(highp))$ $RSV^9 < 0.3 \times OSC^9$
030914*040912	12.33%	3	buy rule sell rule	$(BIAS^{14} > MACD^9) \geq MACD^9$ $(OBV - avg.volume^{20}) < avg.volume^{20}$
040913*050912	1.57%	20	buy rule sell rule	$(RSI^5 - BIAS^{20}) \leq 0.7$ $((MA^{14})OR(MA^{14})) \leq MA^{20}$
050913*060912	1.12%	16	buy rule sell rule	DIF $(BIAS^{20} - RSI^5) \geq (PSY^{14} + PSY^{14})$
060913*070912	16.60%	21	buy rule sell rule	$-0.6 \leq (amoutp \geq -0.1)$ $(RSV^9 < MACD^9) > PSY^{14}$
070913*080911	-6.11%	5	buy rule sell rule	$BIAS^{14}$ $BIAS^{20}$
080912*090911	34.01%	17	buy rule sell rule	$(0.9 \geq RSI^{14}) + BIAS^{14}$ $-1.0 - (highp > avg.volume^{20})$
090912*100521	4.48%	14	buy rule sell rule	$D^3 > WMS^5$ $(DIF - MTM^5) \leq (0.6 \times WMS^5)$

REFERENCES

- [1] P. F. Blanco, D. Sagi, and J.I.Hdalgo, "Technical market indicators optimization using evolutionary algorithm," *Proc. of the 2008 GECCO conference companion on Genetic and evolutionary computation*, Atlanta, USA, July 2008.
- [2] M. A. Boyacioglu and D. Avci, "An adaptive network-based fuzzy inference system ANFIS for the prediction of stock market return: The case of the istanbul stock exchange," *Expert Systems with Applications*, Vol. 37, pp. 7908–7912, 2010.
- [3] T. J. Chang, S. C. Yang, and K. J. Chang, "Portfolio optimization problems in different risk measures using genetic algorithm," *Expert Systems with Applications*, Vol. 36, pp. 10529–10537, 2009.
- [4] J. S. Chen, J. L. Hou, S. M. Wu, and Y. W. C. Chien, "Constructing investment strategy portfolios by combina-

- tion genetic algorithms,” *Expert Systems with Applications*, Vol. 36, pp. 3824–3828, 2009.
- [5] D. Y. Chiu and P. J. Chen, “Dynamically exploring internal mechanism of stock market by fuzzy-based support vector machines with high dimension input space and genetic algorithm,” *Expert Systems with Applications*, Vol. 36, pp. 1240–1248, 2009.
- [6] R. Choudhry and K. Garg, “A hybrid machine learning system for stock machine market forecasting,” *Proc. of World Academy of Science, Engineering and Technology*, Vol. 39, pp. 315–318, Mar. 2008.
- [7] M. R. Hassan, B. Nath, and M. Kirley, “A fusion model of hmm, ann and ga for stock market forecasting,” *Expert System with Application*, Vol. 33, pp. 171–180, 2007.
- [8] J. H. Holland, *Adaptation in natural and artificial systems: an introduction analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975.
- [9] W. Huang, Y. Nakamori, and S. Y. Wang, “Forecasting stock market movement direction with support vector machine,” *Computers & Operations Research*, Vol. 32, pp. 2513–2522, 2005.
- [10] H. Ince and T. B. Trafalis, “Short term forecasting with support vector machines and application to stock price prediction,” *International Journal of General Systems*, Vol. 37, pp. 677–687, 2008.
- [11] K. Kwon and B. R. Moon, “A hybrid neurogenetic approach for stock forecasting,” *IEEE Transactions on Neural Networks*, Vol. 18(3), pp. 851–864, 2007.
- [12] Y. Leu, C. P. Lee, and C. C. Hung, “A fuzzy time series-based neural network approach to option price forecasting,” *the 2nd Asian Conference on Intelligent Information and Database Systems*, Vol. 5990, pp. 360–369, 2010.
- [13] J.-Y. Potvin, P. Soriano, and M. Vallee, “Generating trading rules on the stock markets with genetic programming,” *Computers & Operations Research*, Vol. 31, pp. 1033–1047, 2004.
- [14] N. Powell, S. Y. Foo, and M. Weatherspoon, “Supervised and unsupervised methods for stock trend forecasting,” *Proc. the 40th Southeastern Symposium on System Theory*, New Orleans, LA, USA, Mar. 2008.
- [15] Taiwan Economic Journal Co., Ltd, “TEJ.” [http://www.tej.com.tw/twsite/ Default.aspx](http://www.tej.com.tw/twsite/Default.aspx), 1991.
- [16] C. F. Tsai and Y. C. Hsiao, “Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches,” *Decision Support Systems*, Vol. 50, pp. 258–269, 2010.
- [17] T. J. Tsai, C. B. Yang, and Y. H. Peng, “Genetic algorithms for the investment of the mutual fund with global trend indicator,” *Expert Systems with Applications*, Vol. 38(3), pp. 1697–1701, 2011.
- [18] T. H. K. Yu and K. H. Huang, “A neural network-based fuzzy time series model to improve forecasting,” *Expert Systems with Applications*, Vol. 37, pp. 3366–3372, 2010.
- [19] M. Zhang and W. Smart, “Using gaussian distribution to construct fitness functions in genetic programming for multiclass object classification,” *Pattern Recognition Letters*, Vol. 27, pp. 1266–1274, 2006.