Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 35 (2014) 137 – 146

Procedia
Computer Science

18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems - KES2014

Taiwan stock investment with gene expression programming

Cheng-Han Lee, Chang-Biau Yang*, Hung-Hsin Chen

Department of Computer Science and Engineering
National Sun Yat-sen University
70 Lienhai Rd., Kaohsiung 80424, Taiwan

Abstract

In this paper, we first find out some good trading strategies from the historical series and apply them in the future. The profitable strategies are trained out by the gene expression programming (GEP), which involves some well-known stock technical indicators as features. Our data set collects the 100 stocks with the top capital from the listed companies in the Taiwan stock market. Accordingly, we build a new series called portfolio index as the investment target. For each trading day, we search for some similar template intervals from the historical data and pick out the pertained trading strategies from the strategy pool. These strategies are validated by the return during a few days before the trading day to check whether each of them is suitable or not. Then these suitable strategies decide the buying or selling consensus signal with the majority vote on the trading day. The training period is from 1996/1/6 to 2012/12/28, and the testing period is from 2000/1/4 to 2012/12/28. Two simulation experiments are performed. In experiment 1, the best average accumulated return is 548.97% (average annualized return is 15.47%). In experiment 2, we increase the diversity of trading strategies with more training. The best average accumulated return is increased to 685.31% (average annualized return is 17.18%). These two results are much better than that of the buy-and-hold strategy, whose return is 287.00%.

Keywords: gene expression programming; stock investment; majority vote; technical indicator; strategy pool.

1. Introduction

Stock investors desire to obtain the best trading strategy for earning positive return in the stock market. However, it is very difficult to develop a stably profitable strategy. Many methods based upon artificial intelligence have been developed to predict the trend of stock price and stock selection. Huang[1] applied the hybrid method of support vector regression and genetic algorithm (GA)to the selection of invested stocks, and they utilized the buy-and-hold method to test the performance. Wen *et al.*[2] utilized the support vector machine (SVM) to predict the trend of stock prices, and a fixed strategy is used to trade the stocks. Ni *et al.*[3] utilized SVM and fraction feature selection method to predict the trend of stock prices. Tsai *et al.*[4] first defined the global trend indicator for evaluating the price change trend of the mutual funds, and then utilized GA to determine the weights in the scoring function for selecting the portfolio.

* Corresponding author. Tel.: +886-7-5252000 ; fax: +886-7-5254301.
E-mail address: cbyang@cse.nsysu.edu.tw

doi:10.1016/j.procs.2014.08.093

Huang *et al.*[5] applied the fuzzy-base method to select stocks, where the fuzzy membership functions are adjusted by GA. Potvin *et al.*[6] utilized the genetic programming (GP) to evolve dynamic trading strategies for stocks. Yan and Clark[7] proposed the robust scheme to train the stock-selection method with GP. Jhou *et al.*[8] utilized GP to generate multiple dynamic strategies for trading stocks. Hsu[9] combined GP and the self-organizing map to predict the stock prices. Huang *et al.*[10] utilized the *gene expression programming* (GEP) to generate trading signals for stocks. Chen *et al.*[11] utilized the Sortino ratio to select funds, and then GEP is applied to evolve dynamic trading strategies for funds. Wang *et al.*[12] utilized the particle swarm optimization to adjust the weights and parameters of technical indicators for trading stocks. Chang *et al.*[13] utilized the dynamic time warping to search similar historical price patterns, and the back-propagation neural network is applied to tune the weights for better turning points prediction. Zhang *et al.*[14] utilized the empirical mode decomposition, which is an important part of the Hilbert-Haung transformation[15], to predict the movement of stock index movement. Most of previous researches focused on the price trend prediction or stock selection. However, it not sufficient for investors to make decisions in the investment. Thus, some automatic trading strategies are required.

In the stock or fund market, once an anomaly has become public knowledge, we would expect it to disappear in future. The property is called the self-destruction of predictability and it was discussed by Timmermann and Granger in[16]. Also, the evolution of market makes that it is difficult to predict the stock return to earn profit[17]. To determine the buying or selling time, we need calculate fitness with an expression function. Hence, GEP is applied to generating dynamic strategies for trading in continuous-changing market.

In this paper, we assume that history will be repeated itself. Though this scenario cannot be proved theoretically, it is still effective for generating trading rules. Accordingly, we could find out good trading strategies from the experience of the historical series, and would apply them in the future. The main phases of our method are described as follows. (1) Trading strategies for the historical data are trained and the trained profitable strategies are saved in the strategy pool. (2) On the trading day, *dynamic time warping* (DTW) is applied to finding similar history intervals (template intervals) with some past days (leading interval). (3) Trading strategies are extracted from the template intervals and they are validated in recent days (validation interval). (4) These suitable strategies generate the trading consensus signal on the trading day by the majority vote scheme. In our voting scheme, the available threshold and voting threshold (introduced later) are used to control the effective voting result.

The data set for our experiments is the *portfolio index*, which is built from the accumulated average daily return of the close prices of 100 listed stocks selected by us in the Taiwan stock market. These 100 stocks are the ones with the largest market capital on 1995/1/5, and they have never been removed from the stock market list before 2012/12/28. We perform two experiments. As shown in the experimental results, the best template interval length is 90 days and the best validation interval length is 20 days. In experiment 1, the best average accumulated return is 548.97% (average annualized return is 15.47%), which appears when the voting threshold lies between 0.8 and 0.84, and the available threshold lies between 0.40 and 0.44. The second experiment is to increase the diversity of trading strategies with more training so that the performance gets further improved. In experiment 2, the best voting threshold is same as experiment 1 and the available threshold lies between 0.25 and 0.29. The best average accumulated return in experiment 2 is 685.31% (average annualized return is 17.18%).

The rest of this paper is organized as follows. In Section 2, we introduce gene expression programming and dynamic time warping. In Section 3, we present our method for deciding the trading signals. In Section 4, we present the first experimental results. To convince the stability of our experiments, we compare the similarity of trading signal sequences with near parameters in Section 5. Then, the second experimental results are presented in Section 6. Finally, the conclusion of this paper and possible future works are discussed in Section 7.

2. Preliminaries

In this section, we will introduce the gene expression programming and the dynamic time warping.

2.1. Gene expression programming

In 2001, Ferreira proposed the *gene expression programming* (GEP)[18], which was developed to improve the *genetic programming* (GP)[19]. Both GEP and GP models apply the biological evolution concept for finding the best solution to a certain problem. The main difference between GEP and GP is the implementation of chromosomes. The GP utilizes a tree structure to represent a gene. However, the maintenance of the tree structure is difficult in the implementation, and it takes a lot of time in evaluating and evolving the gene. The GEP uses two kinds of structures to represent a gene, the linear string (*genotype*) and the tree structure (*phenotype*). The steps of GEP are given as follows.

1. **Creating population**
   First, the chromosomes of a predefined size are randomly generated to form the initial population.
2. **Evaluating fitness**
   Each chromosome is expressed by the genotype and its fitness score is evaluated by the phenotype.
3. **Checking termination condition**
   Check whether the termination condition is satisfied or not. If it is satisfied, then the evolution stops.
4. **Preserving the best chromosome and selecting superior ones**
   The GEP keeps the best chromosome, and picks other chromosomes by elitist selection. Some new chromosomes are generated by a series of evolution operations with probability, which are the mutation, transposition and recombination.
5. **New population**
   After some of the above evolution operations have been applied, a new population is generated. Then, go to Step 2 for repeating the procedure.

2.2. Dynamic time warping

Dynamic time warping (DTW)[20] is a method for measuring the similarity between two data sequences of time series. The dynamic programming formula for solving DTW is described as follows.

$$\begin{aligned}
&DTW_{0,0} = 0,\\
&DTW_{0,j} = DTW_{i,0} = \infty,\\
&DTW_{i,j} = d(t_i, r_j) + min\left\{DTW_{i-1,j}, DTW_{i,j-1}, DTW_{i-1,j-1}\right\},\\
&1 \le i \le T, 1 \le j \le R.
\end{aligned}$$

where t_i, r_j, T, and R denote the ith point in series t, the jth point in series r, the length of series t, and the length of series R, respectively, and $d(\cdot,\cdot)$ represents the function for measuring the distance between two data points, usually the Euclidean distance.

3. Our method

3.1. Construction of the data set

We select 100 listed stocks in the Taiwan stock market as our investment target. These stocks are with the top 100 market capital on 1995/1/5 and they have never been removed from the market list before 2012/12/28. These 100 stocks constitute our data set, which is extracted from the Taiwan Economic Journal (TEJ) database[21]. Their names can be found in[22]. We adopt the adjusted stock price in order to exclude the effect of dividend and right.

We define *portfolio index* as the accumulated average daily return of the close prices of these 100 stocks, which is calculated as follows.

$$\begin{aligned}
R_i(t) &= \frac{C_i(t) - C_i(t-1)}{C_i(t-1)}, & PR(t) &= \frac{\sum_{i=1}^{100} R_i(t)}{100},\\
PI(t) &= PI(t-1) \times (1 + PR(t)), & PI(0) &= 100,
\end{aligned} \tag{1}$$

where $PI(t)$, $PR(t)$, $C_i(t)$ and $R_i(t)$ are the portfolio index, the portfolio rate, the close price of the ith stock on day t, and the return of the ith stock on day t, respectively. The initial value of PI is set to 100.

There are also two series, the *total trading volume* and the *total trading value*, which are obtained from the summation of the 100 stocks.

$$VOL_{sum}(t) = \sum_{i=1}^{100} VOL_i(t), \qquad VAL_{sum}(t) = \sum_{i=1}^{100} VAL_i(t), \tag{2}$$

where $VOL_i(t)$ and $VAL_i(t)$ denote the trading volume and the trading value of the ith stock on day t, respectively.

3.2. The overview of our method

1. **Strategy training**
   Every L days of the historical data form a *template interval*, where two neighboring template intervals have a distance of 10 days. In other words, if one template interval starts on day t with length L, then the next interval will start on day $t + 10$. For each template interval, we utilize the GEP to train some profitable trading strategies, which are saved in the strategy pool.
2. **Template search**
   Assume that the current trading day is t. The *leading interval* with length L is defined as the interval from day $(t - L + 1)$ to day t. On each trading day in the testing period, DTW is applied to finding some history intervals (template intervals) which is similar to the leading interval.
3. **Validation check**
   The *validation interval* with length L_v is defined as the interval from day $(t - L_v + 1)$ to day t. Trading strategies are extracted from the template intervals, and then they are validated in the validation interval to check whether they are suitable for the validation interval or not.
4. **Final trading consensus**
   These suitable strategies are gathered to decide the trading consensus (buying signal, selling signal, or holding signal) on the trading day t with the majority vote scheme.
5. **Final trading consensus**
   Repeat Step 2 through Step 4 until each trading day in the testing period has gotten its trading consensus signal. Finally, compute the return.

3.3. Trading strategy training

In the training phase, we utilize GEP to train each template interval, and save ten most profitable strategies in the *strategy pool*. For the GEP training, the function set is $\{>, \geq, <, \leq, =, and, or, +, -, \times, \div\}$, and the terminal set is $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ combined with 27 basic information and technical indicators. These 27 features are PI, PR, VOL, VAL, MA^5, MA^{10}, MA^{20}, EMA^5, EMA^{10}, EMA^{20}, VMA^5, VMA^{10}, VMA^{20}, OBV, RSI^{14}, CMO^{14}, MOM^5, MOM^{10}, MOM^{20}, $BIAS^5$, $BIAS^{10}$, $BIAS^{20}$, OSC^5, OSC^{10}, OSC^{20}, $TAPI$, and $MACD^{12,26}$, where the superscript numbers attached in the features are the parameters of the technical indicators. For example, $MA^5(t)$ denote average close prices in days $t, t-1, \cdots t-4$. The details and calculation formulas of these 27 features can be found in[22].

Before training, we use the standard z-score function to normalize each indicator sequence in each template interval. Next, for each indicator, we extract the normalized values in five different days as our features: days t, $t-2$, $t-4$, $t-9$ and $t-19$. The concept of days $t-4$, $t-9$ and $t-19$ are similar to last week, last double weeks and last month, respectively. Therefore, the total number of features extracted is $27 \times 5 = 135$.

In GEP evolving, a chromosome represents one trading strategy, and it consists of one buying gene for calculating the buying trend value R_{buy}, and one selling gene for computing the selling trend value R_{sell}. Then, we invoke Algorithm 1 to decide the trading signal of the strategy: BUY, SELL, or WAIT.

Once the trading signal has been determined, we buy stocks on the trading day if we receive BUY signal and we do not have any stock; we sell stocks if we receive SELL signal and we hold stocks; otherwise we do nothing, no matter we have stocks or not. Accordingly, we can compute the return of each trading strategy (chromosome) in one template interval. When the evolution of GEP finishes, the top ten profitable strategies of this template interval are picked out and saved in the strategy pool.

Algorithm 1 Signal decision of one trading strategy.

```
$R_{buy}$ ← calculated value of the gene 1
$R_{sell}$ ← calculated value of the gene 2
if $R_{buy} > 0$ and $R_{sell} \leq 0$ then return BUY
else if $R_{buy} \leq 0$ and $R_{sell} > 0$ then return SELL
else return WAIT
end if
```

3.4. Trading consensus signals in the testing period

We search for k template intervals which is the most similar to the leading interval. Here, we set $k = 3$. Then, these trained strategies are validated to examine their return performance in the *validation interval*. Only the strategies which are *suitable* during the validation interval can vote for deciding the buying or selling consensus signal on the trading day t. The details are described as follows.

Step 1: Setting the parameters

Set the values of four parameters L, L_v, γ_A, and γ_V,where L and L_v denotes the lengths of leading interval and validation interval, respectively, γ_A and γ_V are *available threshold* and *voting threshold*, respectively, which will be explained later.

Step 2: Searching similar template intervals

Compare the leading interval of the current trading day t against all template intervals with the measurement of DTW. After that, $k = 3$ similar template intervals are chosen.

Step 3: Trading strategy validation

Since each template interval preserves ten profitable trading strategies in the strategy pool and three similar template intervals are selected, totally 30 strategies are extracted. Apply each strategy to the validation interval, and then the return R_i of strategy i during the validation interval is obtained.

Next, we compare R_i with the return R_{BH} of buy-and-hold strategy during the validation interval to decide if strategy i is suitable.

Step 4: Trading consensus signal voting

If a strategy is suitable, it will be applied to the trading day to decide the trading signal. If the signal is BUY, the buying vote (V_B^+) is increased by one; if the signal is SELL, the selling vote (V_S^+) is increased by one; otherwise (i.e. WAIT), the unavailable vote (V^-) is increased by one. If a strategy is not suitable, only V^- is added by one.

After voting, the trading signal on day t is decided according to Algorithm 2. Only when the percentage of effective signals (total number of BUY and SELL signals) exceeds the *available threshold*, denoted as γ_A, we launch the decision of the final trading consensus signal by the majority vote. Next, if one of the BUY or SELL percentages is higher than the *voting threshold*, denoted as γ_V, then the Buying consensus or Selling consensus is determined accordingly. For other cases, the decision is Confusion, which means nothing to do.

Step 5: Return calculation

Now we invest on PI for each day in the testing period according to the trading consensus with the following procedure.

- **Buying consensus**: If we currently have no stock, we spend all money to buy PI.
- **Selling consensus**: If we currently have some stocks, we sell all held stocks.
- **Confusion** :The state is the same as the previous one.

Algorithm 2 Trading consensus signal voting.

```
$V^+ \leftarrow V_B^+ + V_S^+$
$V \leftarrow V^+ + V^-$
if $\frac{V^+}{V} > \gamma_A$ then
    if $\frac{V_B^+}{V^+} > \gamma_V$ then Buying consensus
    else if $\frac{V_S^+}{V^+} > \gamma_V$ then Selling consensus
    else Confusion
    end if
else Confusion
end if
```

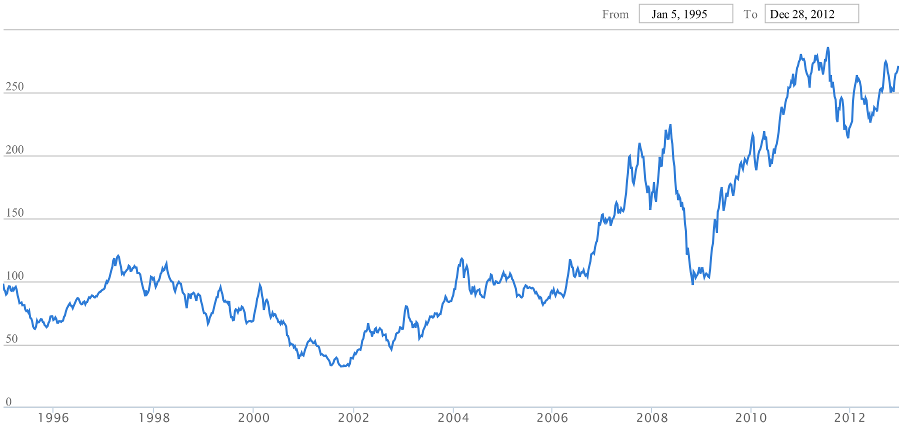


Fig. 1: The portfolio index from 1995/1/5 to 2012/12/28.

4. Results of the first experiment

Assume that in each trading day, we can always buy and sell on portfolio index (PI) in after-hour trading. We set first training day on beginning of 1996 (not 1995) because some indicators need the values from past days. In our experiments, the training period is from 1996/1/6 to 2012/12/28, and the testing period is from 2000/1/4 to 2012/12/28. The PI series in the training period, which contains the testing period, is shown in Figure 1. The return of the buy-and-hold strategy, with buying on 2000/1/4 and selling on 2012/12/28, is 287.00%. We deduct the total transaction fee as 0.6% when stocks are sold.

Our program is written in Python, and we utilize *PyGEP*[23] to implement the GEP. The parameters of the GEP used in this paper are shown in Table 1.

In the first experiment, we try to get proper parameter values. We have tried various values of parameters, template interval length $L = \{60, 90, 120, 180\}$, validation interval length $L_v = \{0, 10, 20, 30, 40, 50\}$, available threshold $\gamma_A = \{0.00, 0.01, 0.02, \cdots, 0.69\}$ and voting threshold $\gamma_V = \{0.50, 0.51, 0.52, \cdots, 0.89\}$. There are totally $4 \times 5 \times 70 \times 40 = 56000$ combinations. Every simulation in the first experiment and second experiment are executed 10 times, and then the average of the ten simulations is obtained. In order to simplify the experimental results, we group the results in a block table by gathering continuous five values of γ_A and five values of γ_V and then averaging them. So there are totally 250 experimental results in each block in the following tables.

In the training phase of the first experiment, the trading strategies of each template interval are trained once, and the best ten strategies are saved in the strategy pool. Three template intervals, most similar to the leading interval, are selected, so 30 strategies are collected to vote the trading consensus signal of each trading day.

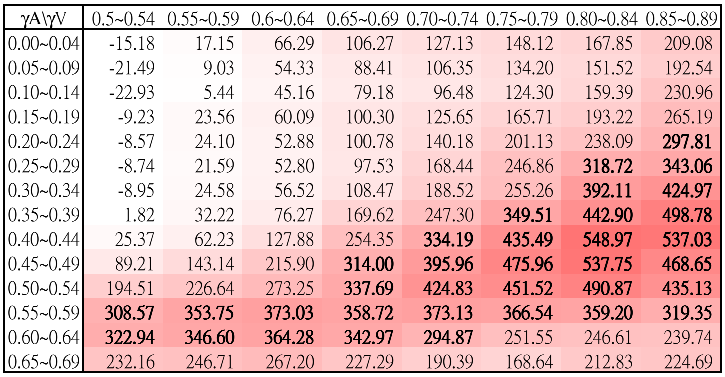
Table 1: The parameters of the gene expression programming.

Number of genes	2	IS transposition rate	30%
The length of head	5	RIS transposition rate	30%
Population size	1000	Gene transposition rate	30%
Number of generations	200	1-point recombination rate	50%
Selection method	roulette wheel	2-point recombination rate	50%
Mutation rate	18.18%	Gene recombination rate	30%

Table 2: Number of blocks exceeding the return of the buy-and-hold strategy

$L\backslash L_v$	0	10	20	30	40	50
60	0	5	14	12	6	14
90	19	34	35	29	27	9
120	0	11	4	0	0	1
180	0	0	0	0	0	0

Table 3: The average return (%) when $L = 90$ and $L_v = 20$. There are 35 blocks exceeding the return of the buy-and-hold strategy.



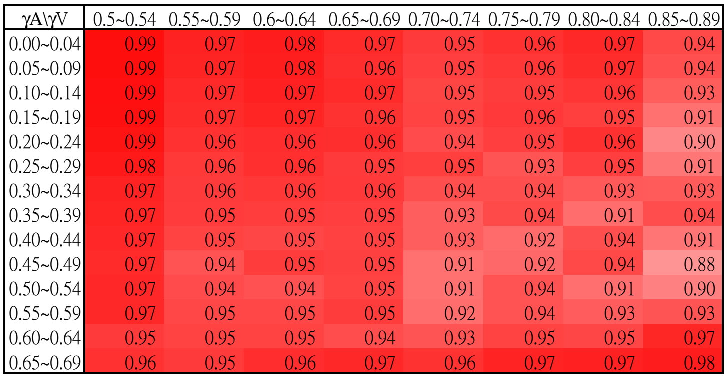
γA\γV	0.5~0.54	0.55~0.59	0.6~0.64	0.65~0.69	0.70~0.74	0.75~0.79	0.80~0.84	0.85~0.89
0.00~0.04	-15.18	17.15	66.29	106.27	127.13	148.12	167.85	209.08
0.05~0.09	-21.49	9.03	54.33	88.41	106.35	134.20	151.52	192.54
0.10~0.14	-22.93	5.44	45.16	79.18	96.48	124.30	159.39	230.96
0.15~0.19	-9.23	23.56	60.09	100.30	125.65	165.71	193.22	265.19
0.20~0.24	-8.57	24.10	52.88	100.78	140.18	201.13	238.09	**297.81**
0.25~0.29	-8.74	21.59	52.80	97.53	168.44	246.86	**318.72**	**343.06**
0.30~0.34	-8.95	24.58	56.52	108.47	188.52	255.26	**392.11**	**424.97**
0.35~0.39	1.82	32.22	76.27	169.62	247.30	**349.51**	**442.90**	**498.78**
0.40~0.44	25.37	62.23	127.88	254.35	**334.19**	**435.49**	**548.97**	**537.03**
0.45~0.49	89.21	143.14	215.90	**314.00**	**395.96**	**475.96**	**537.75**	**468.65**
0.50~0.54	194.51	226.64	273.25	**337.69**	**424.83**	**451.52**	**490.87**	**435.13**
0.55~0.59	**308.57**	**353.75**	**373.03**	**358.72**	**373.13**	**366.54**	**359.20**	**319.35**
0.60~0.64	**322.94**	**346.60**	**364.28**	**342.97**	**294.87**	251.55	246.61	239.74
0.65~0.69	232.16	246.71	267.20	227.29	190.39	168.64	212.83	224.69

In Table 2, we collect the results with various combinations of template interval lengths and validation interval lengths. We can find that the best combination is $L = 90$ and $L_v = 20$. Table 3 shows the details of the best combination. We can see that the best return is 548.97% when γ_V lies between 0.8 and 0.84, and γ_A lies between 0.40 and 0.44.

5. Trading similarity test of near parameters

To test the stability of our experiments, we calculate the difference of the trading decisions between every two neighboring cells, since their difference on either γ_A or γ_V is only 0.01. For each combination of parameter (one simulation experiment), we collect the days with the buying consensus signal into a buying-day sequence and with the selling consensus signal into a selling-day sequence. Then the two sequences are calculated to test the similarity of trading behaviors of two neighboring cells. Remember that each block is the average of 250 cells. The similarity of two cells is calculated by Equation 3 as follows.

Table 4: The average similarity of trading behavior with left cell (fixed γ_A)when $L = 90$ and $L_v = 20$.



γA\γV	0.5~0.54	0.55~0.59	0.6~0.64	0.65~0.69	0.70~0.74	0.75~0.79	0.80~0.84	0.85~0.89
0.00~0.04	0.99	0.97	0.98	0.97	0.95	0.96	0.97	0.94
0.05~0.09	0.99	0.97	0.98	0.96	0.95	0.96	0.97	0.94
0.10~0.14	0.99	0.97	0.97	0.97	0.95	0.95	0.96	0.93
0.15~0.19	0.99	0.97	0.97	0.96	0.95	0.96	0.95	0.91
0.20~0.24	0.99	0.96	0.96	0.96	0.94	0.95	0.96	0.90
0.25~0.29	0.98	0.96	0.96	0.95	0.95	0.93	0.95	0.91
0.30~0.34	0.97	0.96	0.96	0.96	0.94	0.94	0.93	0.93
0.35~0.39	0.97	0.95	0.95	0.95	0.93	0.94	0.91	0.94
0.40~0.44	0.97	0.95	0.95	0.95	0.93	0.92	0.94	0.91
0.45~0.49	0.97	0.94	0.95	0.95	0.91	0.92	0.94	0.88
0.50~0.54	0.97	0.94	0.94	0.95	0.91	0.94	0.91	0.90
0.55~0.59	0.97	0.95	0.95	0.95	0.92	0.94	0.93	0.93
0.60~0.64	0.95	0.95	0.95	0.94	0.93	0.95	0.95	0.97
0.65~0.69	0.96	0.95	0.96	0.97	0.96	0.97	0.97	0.98

$$
\begin{aligned}
c(x_i, y_j) &= \begin{cases} \dfrac{\mid x_i - y_j \mid^2}{100} & \text{, if } \mid x_i - y_j \mid < 10 \\ 1 & \text{, otherwise} \end{cases}, \\
p(x_i, y_j) &= \begin{cases} \dfrac{\mid x_i - y_j \mid^2}{1000} & \text{, if } \mid x_i - y_j \mid < 10 \\ 0, \text{otherwise} & \end{cases}, \\
d(X_i, Y_j) &= c(x_i, y_j) + min \begin{cases} d(X_{i-1}, Y_j) + p(x_i, y_j) \\ d(X_i, Y_{j-1}) + p(x_i, y_j) \\ d(X_{i-1}, Y_{j-1}) \end{cases}, \\
\mu &= 1 - \frac{d(B_1, B_2) + d(S_1, S_2)}{max(l(B_1) + l(S_1), l(B_2) + l(S_2))},
\end{aligned}
\tag{3}
$$

where $X = x_1 x_2 \cdots x_{|X|}$, $Y = y_1 y_2 \cdots y_{|Y|}$, $X_i = x_1 x_2 \cdots x_i$ and $Y_j = y_1 y_2 \cdots y_j$, B_1 and B_2 denote the buying-day sequences of two adjacent cells , S_1 and S_2 denote the selling-day sequences of two adjacent cells, $c(\cdot,\cdot)$, $p(\cdot,\cdot)$, $d(\cdot,\cdot)$ and $l(\cdot)$ are cost function, penalty function, distance function, and the length of input sequence, respectively, and μ denotes the similarity. One can see that the sum of cost and penalty does not exceed 1 when x_i and y_j try to map together. The value of μ ranges from 0 to 1. The higher μ is, the more similarity between the two sequences in two days is. Here, we think that x_i and y_j are *matched* if $\mid i - j \mid$ is less than ten trading days, but give a small punishment for this case.

Table 4 shows the results of comparing neighboring cells with left/right direction (same γ_A, but different γ_V). Each block shows the average of 250 cells. We can see that all average similarities are higher than 0.80, where the two trading behaviors are identical if the similarity is equal to 1.

6. Results of the second experiment

After the first experiment, the better values for the template interval and validation interval lengths are set as $L = 90$ and $L_v = 20$. Thus, to increase the diversity, in this experiment, we train each template interval for 10 times with $L = 90$ and $L_v = 20$. In each training process, we save the best 10 strategies in the strategy pool, hence 100 strategies are obtained for one template interval. In the testing period, we also select three template intervals similar to each leading interval. Therefore, 300 strategies are gathered to decide the trading consensus signal in each trading day. The second experiment also repeats 10 times and the results are averaged as shown in Table 5. The best return is 685.31% when γ_V lies between 0.8 and 0.84, and γ_A lies between 0.25 and 0.29.

Table 5: The average return (%) in experiment 2, where $L = 90$ and $L_v = 20$. The bold font in one block means that its return is higher than the return of the buy-and-hold strategy.

γA\γV	0.5~0.54	0.55~0.59	0.6~0.64	0.65~0.69	0.70~0.74	0.75~0.79	0.80~0.84	0.85~0.89
0.00~0.04	66.44	120.22	153.68	171.08	200.53	270.12	**328.82**	206.66
0.05~0.09	75.39	129.60	161.53	170.20	193.30	274.35	**350.31**	228.61
0.10~0.14	93.52	151.06	183.87	206.41	263.86	**404.67**	**473.65**	281.86
0.15~0.19	97.89	161.63	194.36	230.47	**302.75**	**467.64**	**531.22**	**342.50**
0.20~0.24	80.58	128.47	163.55	235.33	**324.84**	**523.61**	**600.71**	**445.79**
0.25~0.29	56.08	93.18	132.01	217.11	**362.67**	**573.86**	**685.31**	**467.28**
0.30~0.34	40.69	76.66	111.18	214.22	**400.35**	**593.36**	**633.90**	**366.98**
0.35~0.39	39.68	87.03	113.82	203.45	**367.96**	**530.61**	**553.38**	**406.82**
0.40~0.44	61.62	114.19	162.06	274.46	**453.91**	**578.96**	**549.61**	**456.20**
0.45~0.49	90.10	120.51	145.35	271.10	**506.09**	**562.62**	**507.22**	**401.42**
0.50~0.54	146.15	147.05	147.19	245.94	**393.83**	**363.97**	246.05	253.85
0.55~0.59	46.74	47.66	39.55	38.12	33.09	17.59	48.31	130.40
0.60~0.64	5.44	4.07	4.58	9.02	14.08	20.73	28.63	36.94
0.65~0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

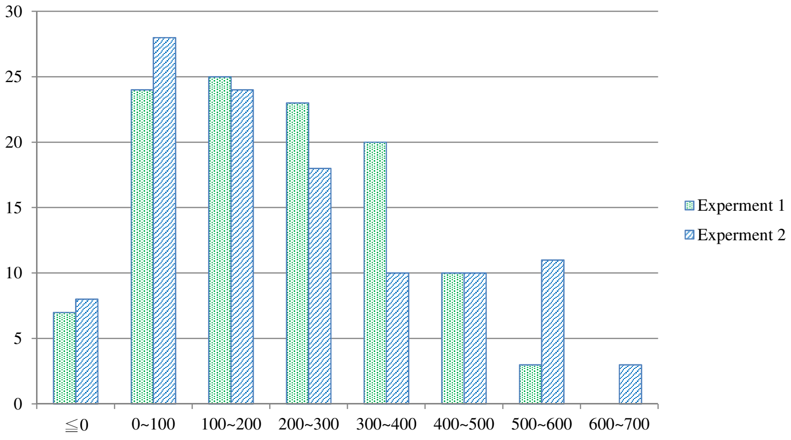


Fig. 2: The histogram of different return ranges in experiment 1 and experiment 2.

Figure 2 illustrates the histogram of various return ranges. We can observe that in experiment 2, the number of blocks with return exceeding 500% is much more than experiment 1. As a conclusion, a larger return will be gotten if a template interval is trained more times. Note that for a specific parameter setting, a template interval is trained ten times and 100 strategies are collected in experiment 2, while it is trained once and 10 strategies are collected in experiment 1.

7. Conclusion

According to some previous researches[6,8,10,11], the evolution way to generate dynamic trading strategies is profitable. Some good trading strategies are learned from the historical series and they can be applied in the future. In this paper, we use the GEP to train out profitable strategies, where some good stock technical indicators are involved as features. These strategies will generate buying, selling or holding signals, and the final trading consensus signal is decided by majority vote.

As the experimental results show, the returns of our methods are much better than the buy-and-hold strategy if the parameters are properly set. In experiment 1, we find out that the better template interval and validation interval lengths are 90 and 20 days, respectively. The best average accumulated return is 548.97%, which appears when γ_V lies between 0.8 and 0.84 and γ_A lies between 0.40 and 0.44. In experiment 2, we perform more training to increase the diversity of trading strategies. The returns are better than experiment 1 and the best average accumulated return is about 685.31%. We also find that the best γ_V is the same as experiment 1 and γ_A lies between 0.25 and 0.29.

In the future, we may apply some adaptive methods to automatically select the values of L and L_v in various situations. For doing this, it may help us to reduce the experiment time.

Acknowledgements

This research work was partially supported by the National Science Council of Taiwan under contract NSC101-2221-E-110-020-MY3.

References

1. Huang, C.F.. A hybrid stock selection model using genetic algorithms and support vector regression. *Applied Soft Computing* 2012; **12**:807–818.
2. Wen, Q., Yang, Z., Song, Y., Jia, P.. Automatic stock decision support system based on box theory and SVM algorithm. *Expert Systems with Applications* 2010;**37**(2):1015–1022.
3. Ni, L.P., Ni, Z.W., Gao, Y.Z.. Stock trend prediction based on fractal feature selection and support vector machine. *Expert Systems with Applications* 2011;**38**:5569–5576.
4. Tsai, T.J., Yang, C.B., Peng, Y.H.. Genetic algorithms for the investment of the mutual fund with global trend indicator. *Expert Systems with Applications* 2011;**38(3)**:1697–1701.
5. Huang, C.F., Chang, B.R., Cheng, D.W., Chang, C.H.. Feature selection and parameter optimization of a fuzzy-based stock selection model using genetic algorithms. *International Journal of Fuzzy Systems* 2012;**14**(1):65–75.
6. Potvin, J.Y., Soriano, P., Vallée, M.. Generating trading rules on the stock markets with genetic programming. *Computers & Operations Research* 2004;**31**(7):1033–1047.
7. Yan, W., Clack, C.D.. Evolving robust GP solutions for hedge fund stock selection in emerging markets. *Soft Computing* 2011;**15**(1):37–50.
8. Jhou, S.M., Yang, C.B., Chen, H.H.. Taiwan stock forecasting with the genetic programming. In: *Proc. of the 16th Conference on Artificial Intelligence and Application (Domestic Track)*. Chungli, Taiwan; 2011, p. 151–157.
9. Hsu, C.M.. A hybrid procedure for stock price prediction by integrating self-organizing map and genetic programming. *Expert Systems with Applications* 2011;**38**:14026–14036.
10. Huang, C.H., Yang, C.B., Chen, H.H.. Trading strategy mining with gene expression programming. In: *Proc. of the 2013 International Conference on Applied Mathematics and Computational Methods in Engineering*. Rhodes Island, Greece; 2013, p. 37–42.
11. Chen, H.H., Yang, C.B., Peng, Y.H.. The trading on the mutual funds by gene expression programming with sortino ratio. *Applied Soft Computing* 2014;**15**:219–230.
12. Wang, F., Yu, P.L., Cheung, D.W.. Complex stock trading strategy based on particle swarm optimization. In: *IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*. New York, USA; 2012, p. 1–6.
13. Chang, P.C., Fan, C.Y., Lin, J.J.. Integrating a piecewise linear representation method with dynamic time warping system for stock trading decision making. In: *Fourth International Conference on Natural Computation*; vol. 2. Jinan, China; 2008, p. 434–438.
14. Zhang, L., Liu, N., Yu, P.. A novel instantaneous frequency algorithm and its application in stock index movement prediction. *IEEE Journal of Selected Topics in Signal Processing* 2012;**6**(4):311–318.
15. Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., et al. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences* 1998;**454**(1971):903–995.
16. Timmermann, A., Granger, C.W.. Efficient market hypothesis and forecasting. *International Journal of Forecasting* 2004;**20**(1):15–27.
17. Lim, K.P., Brooks, R.. The evolution of stock market efficiency over time: a survey of the empirical literature. *Journal of Economic Surveys* 2011;**25**(1):69–108.
18. Ferreira, C.. Gene expression programming: A new adaptive algorithm for solving problems. *Complex Systems* 2001;**13**:87–129.
19. Koza, J.R.. *Genetic Programming: On the programming of computers by means of natural selection*; vol. 1. Cambridge, USA, MA: MIT press; 1992.
20. Berndt, D.J., Clifford, J.. Using dynamic time warping to find patterns in time series. In: *Association for the Advancement of Artificial Intelligence Technical Report*. 1994, p. 395–370.
21. Taiwan Economic Journal Co., Ltd, . TEJ. http://www.tej.com.tw/twsite/; 1991.
22. Lee, C.H.. *Stock Investment with Gene Expression Programming*. Master Thesis, National Sun Yat-sen University, Kaohsiung, Taiwan; 2013.
23. Ryan J. O'Neil, . Gene Expression Programming for Python. https://code.google.com/p/pygep/; 2007.