Applied Soft Computing 15 (2014) 219–230

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

# The trading on the mutual funds by gene expression programming with Sortino ratio☆

Hung-Hsin Chen[a], Chang-Biau Yang[a,*], Yung-Hsing Peng[b]

[a] Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan
[b] Innovative DigiTech-Enabled Applications & Services Institute, Institute for Information Industry, Kaohsiung, Taiwan

ARTICLE INFO

*Article history:*
Received 23 May 2012
Received in revised form 21 June 2013
Accepted 17 September 2013
Available online 8 November 2013

*Keywords:*
Mutual fund
Gene expression programming
Sortino ratio
Mean variance model
Superior predictive ability test

ABSTRACT

The aim of this paper is to combine several techniques together to provide one systematic method for guiding the investment in mutual funds. Many researches focus on the prediction of a single asset time series, or focus on portfolio management to diversify the investment risk, but they do not generate explicit trading rules. Only a few researches combine these two concepts together, but they adjust trading rules manually. Our method combines the techniques for generating observable and profitable trading rules, managing portfolio and allocating capital. First, the buying timing and selling timing are decided by the trading rules generated by gene expression programming. The trading rules are suitable for the constantly changing market. Second, the funds with higher Sortino ratios are selected into the portfolio. Third, there are two models for capital allocation, one allocates the capital equally (EQ) and the other allocates the capital with the mean variance (MV) model. Also, we perform superior predictive ability test to ensure that our method can earn positive returns without data snooping. To evaluate the return performance of our method, we simulate the investment on mutual funds from January 1999 to September 2012. The training duration is from 1999/1/1 to 2003/12/31, while the testing duration is from 2004/1/1 to 2012/9/11. The best annualized return of our method with EQ and MV capital allocation models are 12.08% and 12.85%, respectively. The latter also lowers the investment risk. To compare with the method proposed by Tsai et al., we also perform testing from January 2004 to December 2008. The experimental results show that our method can earn annualized return 9.07% and 11.27%, which are better than the annualized return 6.89% of Tsai et al.

## 1. Introduction

Investors usually transact many kinds of financial products, including the mutual funds. However, the financial market is so complex with nonstationary and chaotic data series, thus it is usually of much risk. It is difficult for investors who lack comprehensive investment information and knowledge to manage the portfolio for earning return and decreasing the unsystematic risk in the market. To earn high return in the financial market, some researchers have applied the machine learning techniques and soft computing methods such as *support vector regression* (SVR) [14,36] and *artificial neural network* (ANN) [17,18] to the prediction of the value in the future, *support vector machine* (SVM) [13,14,24] to the prediction of the fluctuation in the next trading interval, *genetic algorithm* (GA) [17,35], *genetic programming* (GP) [25] and *gene expression programming* (GEP) [9] to the optimization of the trading functions, and so on. By using these methods, investors may be able to dig out the rules or the patterns hidden in the market. However, these methods usually do not consider how to manage the portfolio and some of them, such as SVR, ANN and SVM, are not able to generate the observable trading rules. In this paper, we propose a combined method for generating observable and profitable trading rules, managing portfolio and allocating capital.

The assumption of the *efficient market hypothesis* (EMH) is that the prices of traded assets already reflect all known information, therefore they reflect unbiased prospect of all investors in the future [8]. The EMH implies that the price reflects all relevant information so that it is impossible to earn excess return by using any information already available to the public. However, the hypothesis is still highly disputed, and it seems not to completely convince some of researchers. From statistical, behavioral and empirical [12,30] points of view, some researchers attempted to invalidate the EMH and showed evidence on the predictability of financial market by using technical analysis and machine learning techniques.

In behavioral finance view, some psychological factors may sometimes affect investors' investment decision. For example, investors are often affected by rumors and news. Endowment effect

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

* Corresponding author. Tel.: +886 7 5252000.
*E-mail address:* cbyang@cse.nsysu.edu.tw (C.-B. Yang).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.09.011

[15,33] describes that people often demand much more to give up an object than they are willing to pay to acquire it. Status quo bias [16,26] describes that the endowment effect causes people to have the tendency to like things staying relatively the same. Investors often give a higher value on the funds in their portfolio, but when suffering loss in the investment, they tend to refuse changing current portfolio or changing portfolio reluctantly. Such a behavior usually makes investors suffer more loss or miss the best time to reaction. Loss aversion [15] describes that giving up an object is greater than the utility associated with acquiring it. Sunk cost effect [33] says that if a cost has already been incurred, then it cannot be recovered. While investors suffer loss in investing in a fund, they tend not to realize the loss and they spend more capital in the fund to expect to get the lost capital back. Disposition effect [29] states the tendency of investors that they sell assets whose price has increased, but they keep assets which have dropped down in value. Systematic trading rules can help investors avoid these cognitive biases and make rational decisions for earning returns and reducing risk.

Technical analysis is a popular method for generating trading rules. However, Timmermann and Granger [34] state that once an anomaly has become public knowledge, it would be expected to disappear or to self-destruct in future samples. With this observation, the trading rules generated by statically profitable technical analysis, such as described in [32], may not be able to earn return persistently. Hence, we adopt an evolving method to generate the trading rules.

In this paper, we confine our investment to mutual funds. A fund pools money from many investors, and then the fund managers invest the money in stocks, bonds, securities, futures, or some combination of these financial products. To earn high return, we desire to study how to decide the trading timing, how to constitute the portfolio for each trading interval, and how to allocate the capital to the portfolio.

The appropriate timing for buying or selling funds may avoid the systemic risk. Tsai et al. [35] proposed the *global trend indicator* (GTI) to evaluate the trend of mutual funds in the market, and then they applied GA to the selection of funds. The return of their method is better than the *buy-and-hold* strategy and the *4433 rule*. In this paper, we assume that GTI can be bought and sold. To determine the trading (buying or selling) timing, our method applies *gene expression programming* (GEP) [9] and *superior predictive ability* (SPA) test [11] to the historical data series of GTI to generate effective trading functions without data snooping.

To constitute a good portfolio, we rank the available funds in the market by their *Sortino ratios* [31]. We choose the funds with higher Sortino ratios into our portfolio. The Sortino ratio, a modification of the *Sharpe ratio* [28], only penalizes the return that is negative or below the user-defined target value. The Sortino ratio has better performance than the Sharpe ratio as a measurement of portfolio risk when the distribution of excess return of the asset is skewed [3].

According to the portfolio theory, investors may reduce the unsystematic risk by investing their money in a wide variety of funds. After the portfolio is built, we perform the capital allocation by the equal principle (EQ model) or the *mean variance model* (MV model) [21]. The former divides the capital for the funds in the portfolio equally. The MV model for allocating the capital to funds can reduce the unsystematic risks as much as possible.

To evaluate the effectiveness of our method, we perform experiments on a dataset of mutual funds. The dataset contains only equity funds which can be bought and sold in Taiwan with U.S. dollars as their trading currency. The data were extracted from FundDJ [10] in the duration from 1999/1/1 to 2012/9/11. There are totally 112 funds on 1999/1/1 and 718 funds on 2012/9/11. The data from 1999/1/1 to 2003/12/31 are utilized as the training set for building the trading functions and related parameters, and the other data from 2004/1/1 to 2012/9/11 are utilized for testing the return performance.

The rest of this paper is organized as follows. In Section 2, we will present some background knowledge, including global trend indicator, gene expression programming, superior predictive ability test, Sortino ratio, and mean variance model. In Section 3, we will present the flowchart and our method for the investment. In Section 4, we will present some experimental results to show that our method can get stable returns by adopting effective trading signals generated by the gene expression programming. Finally, the conclusion will be given in Section 5.

## 2. Preliminaries

### 2.1. The global trend indicator and monitoring indicator

To evaluate the price change trend of mutual funds in the market, Tsai et al. [35] proposed the *global trend indicator* (GTI), whose equation is given as follows:

$$\begin{aligned} R_i(t) &= \frac{NAV_i(t)}{NAV_i(t-1)} - 1, \quad i \in F_t, \\ GTI(t) &= GTI(t-1) \times \left\{ 1 + \frac{\sum_{i \in F_t} R_i(t)}{\sum_{i \in F_t} |R_i(t)|} \right\}, \\ GTI(0) &= 100\,(\text{base value}), \end{aligned} \tag{1}$$

where $F_t$, $NAV_i(t)$, and $R_i(t)$ represent the funds that can be invested on day $t$, the net asset value of fund $i$ on day $t$, and the daily return of fund $i$ on day $t$, respectively. GTI is used to demonstrate the trend of the fund market. In this paper, we calculate GTI since 1995/1/1.

Furthermore, Tsai et al. [35] defined the *monitoring indicator* (MI), derived from GTI and its five simple moving average indicators, to measure whether the fund market is in the bull or bear state. The intervals of the five moving averages are 1, 3, 5, 7 and 9, denoted as $SMA_1(t)$, $SMA_3(t)$, $SMA_5(t)$, $SMA_7(t)$ and $SMA_9(t)$, respectively. The equation of MI is given as follows:

$$\begin{aligned} MI_{win}(t) &= \frac{\sum_{1 \le x < y \le 5} \{y - x | SMA_{2x-1}(t) > SMA_{2y-1}(t)\}}{20}, \\ MI_{lose}(t) &= \frac{-\sum_{1 \le x < y \le 5} \{y - x | SMA_{2x-1}(t) \le SMA_{2y-1}(t)\}}{20}, \\ MI(t) &= MI_{win}(t) + MI_{lose}(t). \end{aligned} \tag{2}$$

If $MI(t)$ is equal to 1 or close to 1, the current market is bull market (rising price), and if the $MI(t)$ is equal to $-1$ or close to $-1$, the current market is bear market (falling price).

### 2.2. Gene expression programming

The *gene expression programming* (GEP) [9] is an evolutionary algorithm which extends the *genetic programming* (GP) [19,20] by using linear structures to describe nonlinear structures such as trees. Similar to GP, GEP optimizes the population by the principles of natural selection, and it simulates that native creatures with the fittest to the environment would survive. Therefore, GEP can be applied to finding optimal solutions for various problems with some specific constraints. Some researches [2,4,7] have shown that GP has good performance in classification and regression problems. Zhou et al. [38] showed that GEP has superior accuracy in classification problems to GP. Also, in some researches [5,6,25], GP was utilized to generate the trading rules for trading financial assets. Sermpinis et al. [27] showed that the GEP method gets higher annualized return in the testing set than GP. According to these research

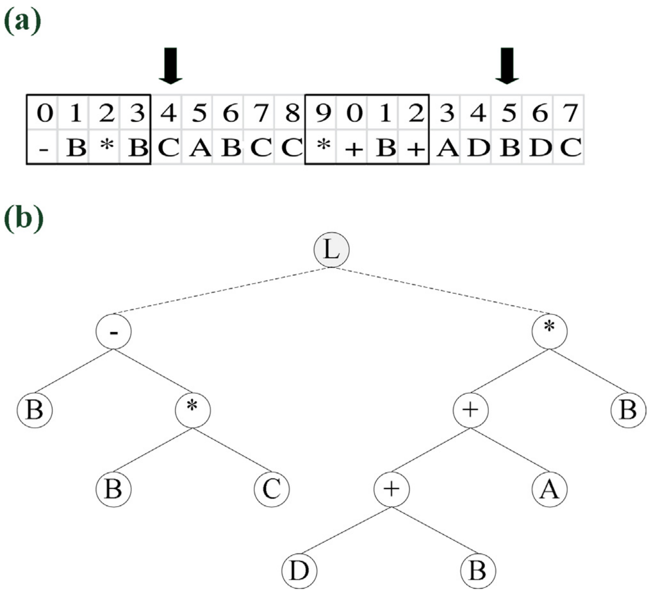


**Fig. 1.** (a) The chromosome with two genes. The head length of each gene is predefined as four, the allele in the rectangle represents the head part, and each black arrow points to the end of one gene which can be transformed into a valid expression tree. (b) The corresponding expression tree of the chromosome, where *L* denotes the user-specified linking function, which is not shown in the chromosome.

results, we adopt GEP as the optimization tool for generating trading rules.

In GEP, a chromosome may contain one or more genes of equal length, where each gene is composed of a head and a tail. Each allele of a gene is constructed from a predefined set of functions and variables. The head part of one gene may contain functions and variables, and its length is decided by the user, while the tail part may contain only variables to keep the gene valid. The length of the tail is decided by the head length and the maximum arguments of the predefined functions as shown in Eq. (3).

$$t = h \times (n - 1) + 1, \quad (3)$$

where *t*, *h*, and *n* represent the tail length, the head length, and the maximum number of the function arguments in the function set, respectively.

Once the head length and the set of functions have been set, the length of a gene is also decided accordingly. The gene of GEP can be transformed into an *expression tree* (ET) by the level-order traversal, and these genes within one chromosome are connected by a user-specified linking function, which can be further explained by the user. The chromosome in this paper will be decoded as a trading function.

For example, Fig. 1 illustrates a chromosome with two genes, where the head length of each gene is four, and the predefined functions are $\{+, -, \times, /\}$ and the variables are $\{A, B, C, D\}$. The left gene represents the left subtree, which is $B - (B \times C)$. The remaining part of the left gene is ignored since they are not valid. The two genes evolve within the chromosome simultaneously, while they can be interpreted as two trading rules joined by a user-specified linking function.

The user-specified fitness function of GEP is utilized to measure the goodness of a chromosome in the population. One chromosome with higher fitness score has higher opportunity to be chosen to progress step. Furthermore, to apply the GEP, users have to specify the parameters and the termination criteria. The parameters include the head length, the number of genes, the population size, the number of generations, the number of replications, the mutation rate, the transposition rate, the recombination rate, and the inversion rate. The termination criteria generally depend on the generations, and the convergence of the fitness scores. When the GEP terminates, the best function is represented by the chromosome with the best fitness score.

The flowchart of GEP is illustrated in Fig. 2, and the evolution steps of GEP are given as follows.

1. *Initialization*: It creates an initial population of chromosomes, which are generated randomly subject to the pre-specified population size.
2. *Elitists selection*: The higher fitness score a chromosome has, the more likely the chromosome is selected to produce offspring.
3. *Evolutionary operations*: The evolutionary operations include the following.
   - The *replication* operation is to copy a chromosome with high fitness score to the new population.
   - The *mutation* operation is applied to one chromosome, and it increases the diversity of the population. To ensure that the gene can be translated to a valid ET, the alleles in the head can be replaced by the elements in the function set or the terminal set, and the alleles in the tail can be replaced only by the elements in the terminal set.
   - In a *transposition of insertion sequence* (IS transposition) operation, a short fragment is inserted into a random position of the head of another randomly chosen gene, and then the fragment is inserted into a random position of the head of another randomly chosen gene except for the first position.
   - In a *root transposition* (RIS transposition) operation, a short fragment is randomly selected from the head of a randomly chosen gene, and then the fragment is inserted into the first position of the gene, where the fragment must start with a function symbol.
   - In a *gene transposition* operation, a randomly selected gene is entirely inserted into the root of the chromosome.
   - In a *1-point recombination* operation, two chromosomes randomly chosen from the population crossover at a randomly chosen point.
   - In a *2-point recombination* operation, two substrings are exchanged between two chromosomes.
   - In a *gene recombination* operation, two randomly chosen genes in two chromosomes are entirely exchanged to generated new offspring.
   - In an *inversion* operation, the alleles in the head of a randomly chosen gene in a chromosome will be re-allocated in the reverse order. These operations facilitate the generation of a variety of offspring.
4. *Termination criteria*: When the termination criteria are satisfied, the outcome is designated as the best solution for the asked problem. If the termination criteria are not fulfilled, GEP executes Steps 2 and 3 iteratively until the termination criteria are satisfied.

We apply the *python* software *pygep* [23] for generating the trading functions in our method.

### 2.3. Superior predictive ability test

Data snooping is the inappropriate use of data mining or statistics to induce misleading relationships in data, which usually occurs when a given set of data is used more than once for purposes of inference or model selection. When such data reuse occurs, there exists the possibility that any satisfactory results obtained may simply be due to chance rather than to any merit inherent in the method yielding the results [37].

White [37] proposed the *reality check* (RC) test to check if there exists at least one model which is able to beat the benchmark

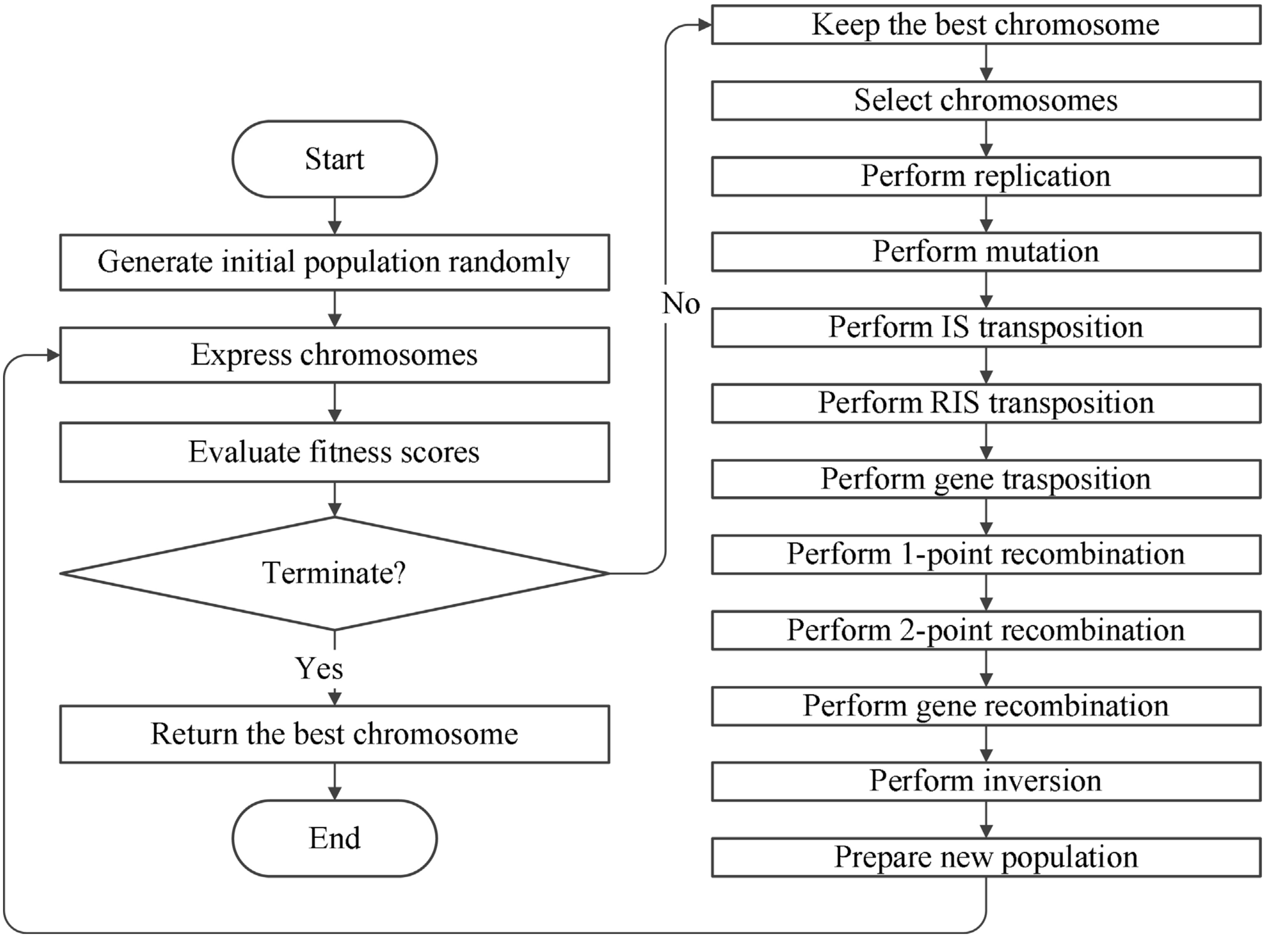


**Fig. 2.** The flowchart of GEP evolution.

without data snooping. Hansen [11] proposed the *superior predictive ability* (SPA) test that utilizes a sample-dependent null distribution to avoid the least favorite configuration in the RC test.

Given $m$ models, let $d_{k,t}$, $1 \le k \le m$ and $1 \le t \le n$, represent the performance measure of the $k$th model related to the benchmark model at time $t$, where $n$ represents the number of intervals in the time series.

Assume that for each model $k$, for all $t$, $E(d_{k,t}) = \mu_k$, where $E(\cdot)$ represents the expectation operator, and $\mu_k$ represents the mean of performance measure of the $k$th model, the null hypothesis is to determine whether the model outperforms the benchmark model. The null hypothesis is shown as follows:

$$H_0^k : \mu_k \le 0, \quad \text{where } 1 \le k \le m. \tag{4}$$

The RC statistics [37] is given as follows:

$$RC_n = \max_{1 \le k \le m} \sqrt{n}\overline{d_k}, \tag{5}$$

where $\overline{d_k} = \sum_{t=1}^{n} d_{k,t}/n$. White chose the least favorite configuration (LFC) $\mu = 0$ to obtain the null distribution. Usually, LFC is too conservative, and it is not able to describe data truthfully. Therefore, Hansen proposed the SPA test to solve this problem [11]. The SPA statistics is given as follows:

$$SPA_n = \max\left( \underset{1 \le k \le m}{max} \sqrt{n}\frac{\overline{d_k}}{\hat{\omega}_k}, 0 \right), \tag{6}$$

where $\hat{\omega}_k$ denotes the consistent estimator of the standard deviation of performance measure of the $k$th model. The main argument for the normalization is that it will improve the power. Since the test uses a data-dependent value for $\mu$ instead of $\mu = 0$, it is usually a more powerful test. In SPA test, $E(d_{k,t}) = \mu_k$ is estimated as follows:

$$\hat{\mu_k} = \overline{d_k} \cdot 1(\sqrt{n}\overline{d_k} \le -\hat{\omega}_k \sqrt{2 \log \log n}), \tag{7}$$

where $1\{\cdot\}$ is the indicator function.

### 2.4. Sortino ratio

We use the Sortino ratio [31] to rank the available funds. The Sortino ratio considers only the downside volatility as the unpleasant part of price fluctuation. The equation of the *Sortino ratio* is given as follows:

$$SV_i^2 = \frac{1}{N_b} \times \sum_{\substack{t=1, \\ R_{it} < R_{MAR}}}^{T_e} (R_{it} - R_{MAR})^2, \tag{8}$$

$$Sortino_i = \frac{E(R_{it}) - R_{MAR}}{SV_i}, \tag{9}$$

where $E(\cdot)$, $R_{MAR}$, $R_{it}$ and $N_b$ represent the expectation operator, the minimum acceptable return, the historical daily returns of the $i$th asset at time $t$ (obtained from the historical price series), and the total number of daily return of the $i$th asset in the past $T_e$ days whose values are below $R_{MAR}$, respectively. With the Sortino ratio, we can find out those funds whose performance is better in the past. These selected funds should have higher probability to earn return in the future, so they are put into our portfolio.

### 2.5. The mean variance model

Investment is a tradeoff between return and risk. The risk in the investment can be represented by the downside variance of the expected return or the average of negative return. Investors usually prefer to have their return as high as possible, but to have the risk as low as possible at the same time. However, an investment with high return usually accompanies with high risk.

The portfolio problem is to determine the capital allocation among a set of funds. Markowitz [21,22] proposed the *mean variance model* (MV model) to solve the problem. The principle of the MV model is to use the expected return of a portfolio as the investment return, and to use the covariance of returns of the portfolio as the investment risk. The MV model is based upon the foundation that the price fluctuation can be diversified by allocating capital among funds having opposite trends.

The definition of the MV model [21,22] is described as follows:

$$\begin{cases} \text{minimize} & \lambda\sum_{i=1}^{n}\sum_{j=1}^{n} w_i \cdot w_j \cdot \sigma_{ij} - (1-\lambda)\sum_{i=1}^{n} w_i \cdot E(R_{it}), \\ \text{subject to} & \sum_{i=1}^{n} w_i = 1 \text{ and } w_i \geq 0\,\forall i, \end{cases} \tag{10}$$

where $E(\cdot)$, $w_i$, $\sigma_{ij}$, $R_{it}$, $n$, and $\lambda$ $(0 \leq \lambda \leq 1)$ represent the expectation operator, the weight of capital allocated to the $i$th asset, the covariance of the returns between the $i$th and $j$th assets, the historical daily return of the $i$th asset, the size of the portfolio (number of assets), and the risk inclination of the investor, respectively.

In the extreme cases of the MV model, $\lambda = 1$ represents that the investor is risk-averse, which results in less expected return and lower risk of the portfolio. In contrast, $\lambda = 0$ represents that the investor is risk-willing, which results in more expected return and higher risk of the portfolio.

## 3. Our method

For the timing of buying and selling the portfolio, our method refers to the signals generated by the GEP. The funds which can be invested are ranked with the *Sortino ratio*, and some of these funds are selected as our portfolio. Then, we apply two capital allocation models to the portfolio. In the first model, the capital in the portfolio is divided equally (EQ model), and in the second model, the capital is allocated with the MV model. The unsystematic risk of the portfolio can be reduced with the MV model by allocating different weights of capital to the selected funds.

### 3.1. The flowchart of investment

The flowchart and corresponding pseudocode of our method are shown in Fig. 3 and Procedure *GEPInvestment*, respectively. In each trading interval (here we adopt one week), we check whether the fund market is open or not by checking the function *isMarketOpen*($t$) [35]. The function *isMarketOpen*($t$) returns “true” if the number of tradable funds in the market on day $t$ is greater than 10% of all funds in the market issued before day $t$. Otherwise, this function returns “false”. If *isMarketOpen*($t$) returns “true”, we will invoke the trading function evolved by GEP that generates signals for deciding the timing of buying or selling. If *isMarketOpen*($t$) returns “false”, we will skip day $t$ and check the result of the next trading day *isMarketOpen*($t+1$).

In bull market, we buy or replace the funds. Before investing, the ranks of funds are determined by the Sortino ratio. Then, according to the fund ranks, inferior funds in our portfolio will be replaced by superior ones. In bear market, all funds in our portfolio will be redeemed.

```
Procedure: GEPInvestment($N$, $t_d$, $\gamma$, buying rule, selling rule)
Input:
  $N$: the portfolio size (number of funds).
  $t_d$: training days.
  $\gamma$: the replacement threshold.
  buying rule: the buy rule trained from GEP.
  selling rule: the sell rule trained from GEP.
Output:
  $R_{cum}$: cumulative return of the portfolio.
  $\mu_{neg}$: risk of the portfolio.
begin
  $t \leftarrow$ the first trading day;
  $R_t^w \leftarrow 0$, $\mu_{neg} \leftarrow 0$, $signal(t) \leftarrow 0$, $signal(t-1) \leftarrow 0$;
  while (t in the investment period) do
    if (isMarketOpen($t$) = True) then
      /*Get the timing of buying or selling by the function
 trained from GEP*/
      $signal(t) \leftarrow$ TradingSignal ($\gamma$, buying rule, selling rule, $signal(t-1)$);
      /* Check current market situation  */
      $R_t^w \leftarrow$ MarketSituation ($N$, $t_d$, $\gamma$, $signal(t)$, $signal(t-1)$);
    end
    $t \leftarrow t+1$;
  end
end
Compute $R_{cum}$ and $\mu_{neg}$, and stop;
```

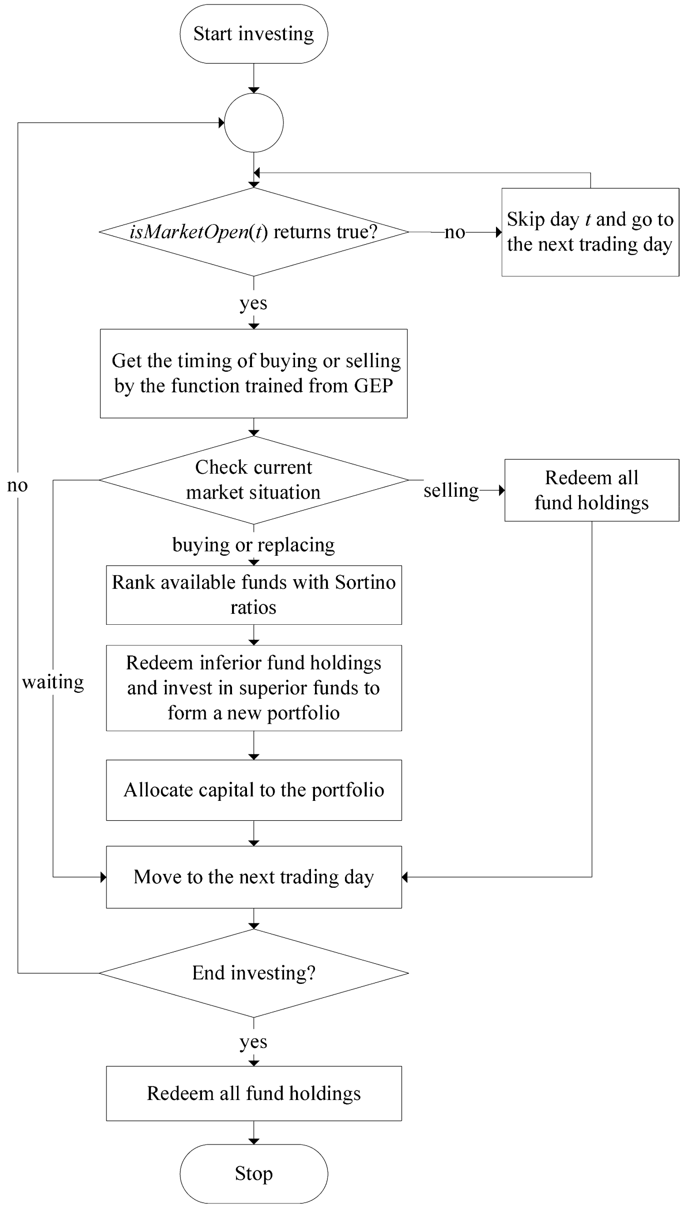

Start investing
isMarketOpen(t) returns true?
no
Skip day t and go to the next trading day
yes
Get the timing of buying or selling by the function trained from GEP
Check current market situation
no
selling
Redeem all fund holdings
buying or replacing
Rank available funds with Sortino ratios
waiting
Redeem inferior fund holdings and invest in superior funds to form a new portfolio
Allocate capital to the portfolio
Move to the next trading day
End investing?
yes
Redeem all fund holdings
Stop

**Fig. 3.** The flowchart for investing funds.

```
Procedure: TradingSignal($\gamma$, buying rule, selling rule, signal($t-1$))
Input:
  $\gamma$: the replacement threshold.
  buying rule: the buy rule trained from GEP.
  selling rule: the sell rule trained from GEP.
  signal($t-1$): signal of the previous trading day.
Output:
  signal($t$): signal of the current trading day.
begin
  /*Get the timing of buying or selling by the function
   trained from GEP*/
  T← the values of terminal variables in the GEP at day $t$;
  if (buying rule(T) > 0 and selling rule(T) ≤ 0) then
    signal($t$)← 1 ;
  else if (buying rule(T) ≤ 0 and selling rule(T) > 0) then
    signal($t$)← 0 ;
  else
    signal($t$)← signal($t-1$) ;
  else
  Output signal($t$) ;
else
```

```
Procedure: MarketSituation($N$, $t_d$, $\gamma$, signal($t$), signal($t-1$))
Input:
  $N$: the portfolio size.
  $t_d$: training days.
  $\gamma$: the replacement threshold.
  signal($t$): signal of the current trading day.
  signal($t-1$): signal of the previous trading day.
output:
  $R_t^w$: return of the portfolio at trading day $t$.
begin
  /* Check current market situation */
  if (signal(t) = 1 and signal(t-1) = 0) or (signal(t) = 1 and signal(t-1) = 1)) then
    /*buy or replace */
    Rank available funds with Sortino ratios using past $t_d$ days NAVs;
    Redeem inferior fund holdings according to the threshold $\gamma$, and invest
   in superior funds to form a new portfolio;
    Allocate capital to the portfolio;
  else if (signal(t) = 0 and signal(t-1) = 1) then
    /* sell */
    Redeem all fund holdings;
  else
    ;/* wait, do nothing */
  end
  Compute $R_t^w$ and output it;
end
```

### 3.2. Stage 1: Determining the timing with GEP

To earn more stable return, the buying timing and selling timing are determined by the trading functions trained by GEP with the global trend indicator (GTI). Once the buying signal appears, a portfolio (a subset of funds) is formed. However, when we determine to buy funds each time, the portfolio may change, so we cannot train the trading functions with the time series of a certain fund. Moreover, since the investment targets are all available funds, not a single fund, it is too risky if the trading function is decided by a single fund. GTI on day $t$ is composed of the average return of the funds which are tradable on day $t$. Thus, GTI represents the trend of the fund market. If GTI goes up, a large of these tradable funds also goes up. Since the selected funds in the portfolio are highly related to trend of GTI, after the buying timing is decided, the trading function of GTI can be applied to the portfolio.

To simplify the procedure of determining the trading timing, we assume that we can buy and sell GTI. Here, the transaction fee 3% in each transaction is involved. The fitness function of GEP is to maximize the return of GTI in the training period.

The function set of GEP contains the operators '>', '<', '=', '≥', '≤'. Our trading interval is one week, thus we put the weekly monitor indicator ($MI^W$) [35] in the terminal set of GEP, which can be viewed as the feature of market situation.

In the evolution, each chromosome consists of two genes as illustrated in Fig. 1, one for buying rule and the other for selling rule. Each gene is represented by an ET, thus we use a two-tree scheme in GEP. The trading function, combining the results from the two ETs by putting the variables at time $t$ to the corresponding location, will generate the trading signal.

If the buying rule and the selling rule are of opposite signals, which means that the two rules have the same opinions, then the value of the trading signal is either 1 or 0. Signal 1 means the timing for buying and signal 0 means the timing for selling. Otherwise the market situation is considered as ambiguous, and the current trading signal remains the same as the previous signal.

Next, the trading signals and their corresponding actions are determined in the following way.

- *buying*: If the trading signal of $t$ is 1 and the trading signal of $(t-1)$ is 0, then we select the funds with higher Sortino ratios as our portfolio.
- *selling*: If the trading signal of $t$ is 0 and the trading signal of $(t-1)$ is 1, then we redeem all the funds in our portfolio.
- *replacing*: If the trading signals of both $t$ and $(t-1)$ are 1, we replace the inferior funds with the superior ones according to Sortino ratio.
- *waiting*: If the trading signals of both $t$ and $(t-1)$ are 0, we do nothing, which means that we are waiting for the buying timing.

After getting the profitable trading functions, generated by GEP for GTI in the training duration, we apply the functions to our portfolio in the testing duration. This procedure will be performed every week.

### 3.3. Stage 2: Portfolio management

When a buying signal appears, we pick out superior funds according to the Sortino ratios from current available funds as our investment objects, and put them into the portfolio with size $N$. That is to say, exactly $N$ funds are invested. When a selling signal appears, the simple job is to redeem all held funds.

If we get a replacing signal, some inferior funds in the current portfolio are replaced by superior ones with the replacement method proposed by Tsai et al. [35]. Here we define a *replacement threshold*, denoted as $\gamma$, for removing inferior funds from the current portfolio. For example, suppose there are 120 available funds and $\gamma = 60\%$. Once the Sortino rank of a certain fund falls outside top 72 (= 120 × 60%), this fund will be replaced by a one with higher Sortino rank. The portfolio is updated every week if the replacing signal remains unchanged.

### 3.4. Stage 3: Capital allocation

We assume that the initial capital is fixed. Since the real investment targets are the available funds, we should allocate the capital among these funds. We apply two models to capital allocation in the portfolio. The first model, called the *EQ model*, distributes the capital to each fund equally. In the second model, the capital is allocated with the MV model.

To allocate our capital with the MV model, it is assumed that the returns of the funds of the portfolio in the future depend on their historical price series. The price fluctuation of the portfolio can be diversified by allocating capital among funds having opposite trends.

If $\lambda = 1.0$ is set in Eq. (10), it means that the inventor is very conservative. There should be higher return with $\lambda < 1$ because the higher risk accompanies the higher return. Hence the parameter $\lambda = 1.0$ serves the lower bound of the return in the MV model.

We can solve Eq. (10) by using *python* software *CVXOPT* [1] to get the best capital weights of the funds in the portfolio. The capital is reallocated when some funds are redeemed and some new funds are added into the portfolio.

## 4. Experimental results

### 4.1. Data collection

Our investment target is the mutual funds. Because the fund manager has diversified the money pool to many capital markets and the risk of investment is reduced, the fluctuation of funds is more stable than other financial assets and it is easier to predict the trend. However, there is still systematic risk, so we apply GEP to generating trading functions for solving this problem.

The dataset, which contains the profiles and the *net asset values* (NAV) of funds, was extracted from FundDJ [10]. The training duration for generating trading functions and related parameters is from 1999/1/1 to 2003/12/31, and the testing duration for obtaining the return performance is from 2004/1/1 to 2012/9/11. The training duration contains three curve patterns: rapidly going up, rapidly going down, and mixed, therefore it is beneficial for training profitable trading functions. The detailed information of our dataset is described as follows.

1. The dataset contains only the funds which can be bought and sold in Taiwan.
2. We consider only equity funds and the trading currency is the U.S. dollar.

The numbers of funds in our dataset are 112 on 1999/1/1 and 718 on 2012/9/11, respectively. The trading interval in our experiments is one week.

### 4.2. Experimental results of the buy and hold strategy

In our experiments, we search for the parameters in the training duration, including the portfolio size $N$, the replacement threshold $\gamma$, and the training days $t_d$ for ranking the Sortino ratio and for the MV model. Then the parameters which get better return in the training duration are applied to the testing duration.

With the buy and hold strategy, in the end of the duration, all funds in the portfolio are redeemed, and the return of the duration is calculated. We consider various parameters with $N = \{5, 10, 15, 20\}$ for the portfolio size, $t_d = \{40, 60, 120\}$ for ranking the Sortino ratios and for the MV model, and $\gamma = \{50\%, 60\%, \ldots, 90\%\}$ for the replacement threshold in the experiments. To make our experiments more close to real scenario, if we switch the capital among the held funds, then 1% transaction fee is charged, and if we buy a new fund, then 3% transaction fee is charged.

To measure the downside volatility, we calculate the weekly average negative return ($\mu_{neg}$) of the portfolio as risk, given as follows:

$$R_t^w = \frac{P_t^w}{P_{t-1}^w} - 1, \qquad \mu_{neg} = \frac{1}{n_{neg}} \left| \sum_{R_t^w < 0} R_t^w \right|, \tag{11}$$

where $R_t^w$, $P_t^w$, $n_{neg}$ represent the weekly return of the portfolio on day $t$, the current value of the portfolio on day $t$, and the number of weekly returns $R_t^w$ whose values are negative, respectively. The smaller the average negative return is, the lower risk of the portfolio is.

Also, the superior predictive ability (SPA) test is applied to each parameter of our method to check if there exist some parameters which have the ability to earn positive return without data snooping. The number of samples for SPA is set to 5000 to get precise $p$-values. The performance measure for SPA test is described as follows.

$$\bar{d}_k = \frac{1}{n} \sum_{t=1}^{n} (R_{k,t}^w \times S_{k,t-1}), \tag{12}$$

where $\bar{d}_k$ represents the average performance measure of the $k$th model in the test, $n$ represents the number of trading intervals, $R_{k,t}^w$ represents the weekly portfolio return of the $k$th model on day $t$, and $S_{k,t-1} = \{0, 1\}$ corresponds to the trading signal taking neutral (value of 0) or long (value of 1) on day $t - 1$.

The cumulative returns for the EQ capital allocation model with various parameters in the training duration are shown in Fig. 4(a)–(c), where the values on the $x$-axis represent the values of $\gamma$. The results for the MV model ($\lambda = 1.0$, for the most conservative investors) are shown in Fig. 4(d)–(f). We find that the parameters for both models is able to earn positive returns when $t_d = \{60, 120\}$ and $\gamma \geq 60\%$. Furthermore, there is a higher return when $t_d = 120$ and $\gamma = 60\%$.

Since the returns of $t_d = 40$ are not so good in the training duration, we perform experiments with $t_d = \{60, 120\}$ in the testing duration. The cumulative returns are shown in Fig. 5. To compare the risk of the two capital allocation models, we calculate their average negative returns in the training duration, as shown in Table 1. It can be easily seen that the MV model ($\lambda = 1.0$) has lower risk in almost all cases.

To avoid the data snooping problem, we apply the SPA test. The $p$-values are shown in Table 2. From the results, we can see that the parameters $t_d = 120$ and $\lambda = 60\%$ have better $p$-values (smaller $p$-values), which means that there is higher probability to earn positive return with these parameters in the future.

From Fig. 5, we observe that our method could also earn positive returns in the testing duration, although the returns are not high enough. Hence, we analyze the variation of portfolio values in the testing duration, which are shown in Fig. 6. As we can see, all portfolio values fell in year 2008, when all capital markets suffered from financial crisis this year.

### 4.3. Experimental results of the trading timing with GEP

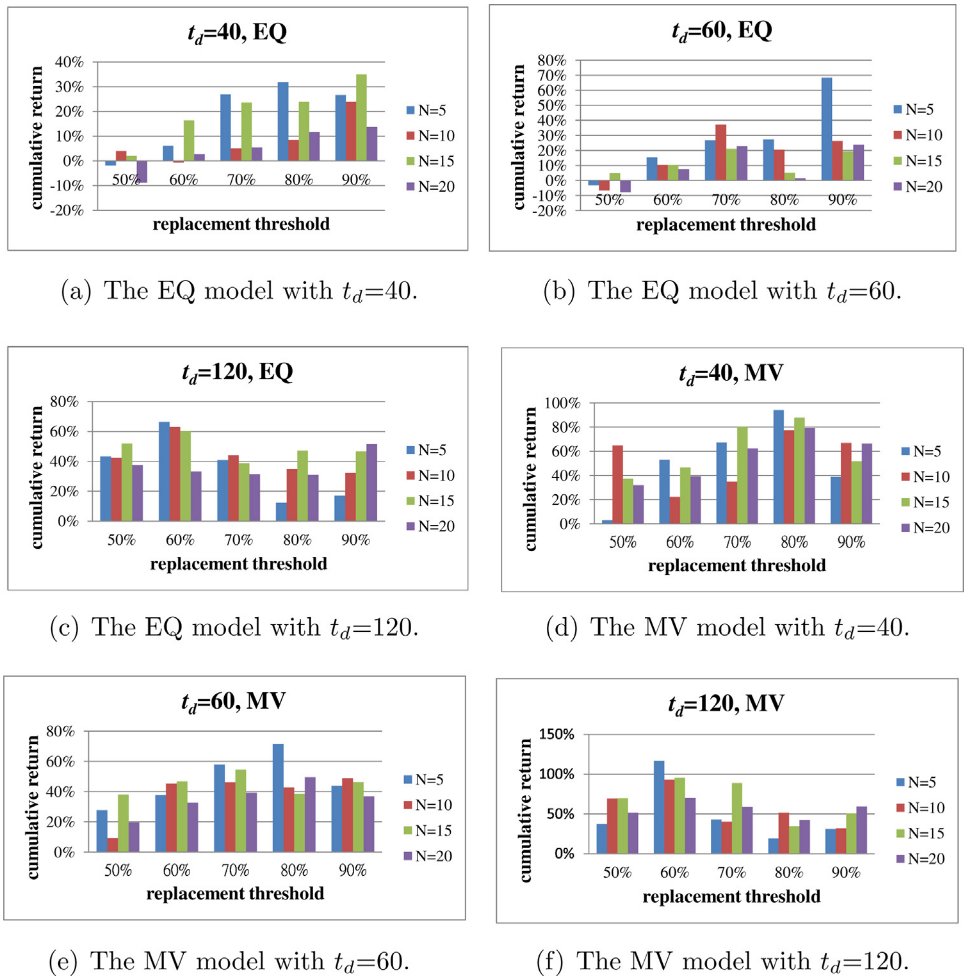
In these experiments, we adopt the trading signals generated by GEP for buying and selling the portfolio. In the training duration, we have obtained better parameters and the best trading functions that generates effective trading signals. These parameters and functions are then applied to the testing duration.

The elements in the function set and terminal set of GEP are $\{>, <, =, \geq, \leq, \neq\}$ and $\{-1.0, -0.9, \ldots, 0.9, 1.0, MI_t^w, MI_{t-1}^w, MI_{t-3}^w, MI_{t-5}^w, MI_{t-7}^w, MI_{t-9}^w\}$, respectively.

In the beginning of the training duration, GEP randomly generates many trading functions (chromosomes), where each function will generate one signal for the buying rule and the other signal for the selling rule. In each trading interval $t$, if the *isMarketOpen*($t$) returns "true", then the real values will be substituted to the terminal variables of each function, thus we can get two signals (values). Next, the trading signal and the corresponding action at $t$ will be determined accordingly. Iterate the above steps until the end of the training duration is reached. Then we can compute the GTI return of each function as its fitness score. These functions are evolved until the termination criteria of GEP are satisfied, finally we can get profitable trading functions for GTI.

The GTI in the training duration and testing duration are shown in Fig. 7. The parameters used by GEP are shown in Table 3.

We run the GEP 100 times for getting profitable trading functions for GTI, and then apply the top 10 functions to the testing



(a) The EQ model with $t_d$=40.

(b) The EQ model with $t_d$=60.

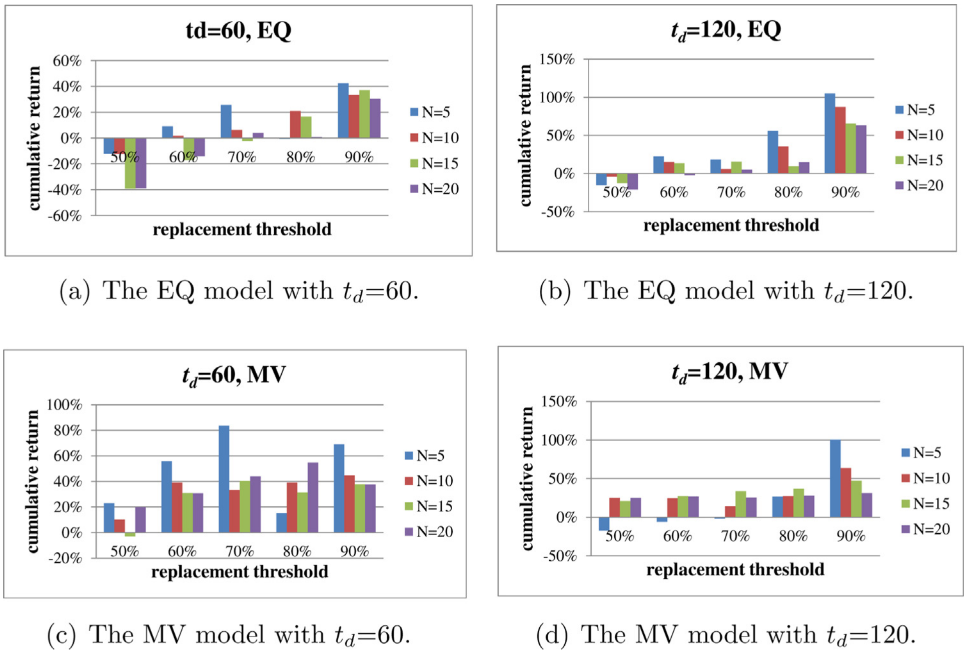
(c) The EQ model with $t_d$=120.

(d) The MV model with $t_d$=40.

(e) The MV model with $t_d$=60.

(f) The MV model with $t_d$=120.

**Fig. 4.** Cumulative returns of the buy and hold strategy in the training duration.



(a) The EQ model with $t_d$=60.

(b) The EQ model with $t_d$=120.

(c) The MV model with $t_d$=60.

(d) The MV model with $t_d$=120.

**Fig. 5.** Cumulative returns of the buy and hold strategy in the testing duration.

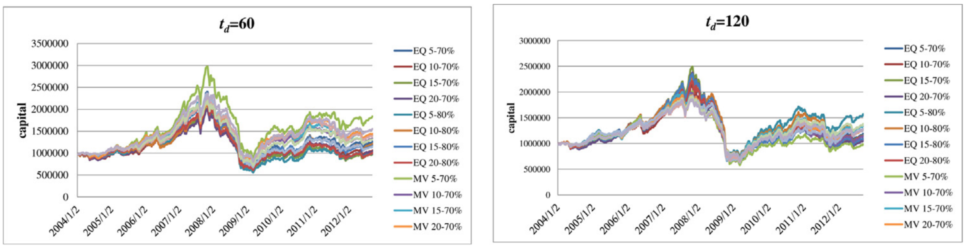
**Table 1**
The average negative returns with $N = \{5, 10, 15, 20\}$, $t_d = \{60, 120\}$, and $\gamma = \{60\%, 70\%, 80\%, 90\%\}$. Here, $\lambda = 1.0$ in the MV model, and $\mu_{neg1}$ and $\mu_{neg2}$ represent the average negative return in the training duration and testing duration, respectively.

| $td-\gamma$ | $N = 5$ | | 10 | | 15 | | 20 | |
|---|---|---|---|---|---|---|---|---|
| | $\mu_{neg1}$ (%) | $\mu_{neg2}$ (%) | $\mu_{neg1}$ (%) | $\mu_{neg2}$ (%) | $\mu_{neg1}$ (%) | $\mu_{neg2}$ (%) | $\mu_{neg1}$ (%) | $\mu_{neg2}$ (%) |
| 60–60% – EQ | 2.26 | 2.46 | 1.99 | 2.25 | 1.82 | 2.24 | 2.00 | 2.21 |
| 60–60% – MV | 2.04 | 2.28 | 1.81 | 2.02 | 1.69 | 2.07 | 1.67 | 2.00 |
| 60–70% – EQ | 2.00 | 2.51 | 2.05 | 2.33 | 1.92 | 2.44 | 2.00 | 2.33 |
| 60–70% – MV | 2.09 | 2.42 | 1.68 | 2.17 | 1.71 | 2.09 | 1.59 | 2.13 |
| 60–80% – EQ | 1.88 | 2.48 | 1.79 | 2.25 | 1.91 | 2.25 | 1.92 | 2.31 |
| 60–80% – MV | 1.89 | 2.09 | 1.59 | 2.07 | 1.54 | 2.05 | 1.53 | 2.13 |
| 60–90% – EQ | 1.98 | 2.37 | 1.84 | 2.25 | 1.87 | 2.27 | 1.83 | 2.32 |
| 60–90% – MV | 1.64 | 1.97 | 1.54 | 1.96 | 1.50 | 1.92 | 1.53 | 2.02 |
| 120–60% – EQ | 1.99 | 2.51 | 1.93 | 2.44 | 1.86 | 2.53 | 1.81 | 2.47 |
| 120–60% – MV | 1.79 | 2.57 | 1.67 | 2.31 | 1.69 | 2.25 | 1.75 | 2.21 |
| 120–70% – EQ | 2.20 | 2.33 | 1.96 | 2.42 | 1.99 | 2.43 | 1.97 | 2.39 |
| 120–70% – MV | 2.01 | 2.29 | 1.78 | 2.03 | 1.65 | 1.98 | 1.79 | 2.09 |
| 120–80% – EQ | 2.01 | 2.31 | 1.93 | 2.26 | 1.95 | 2.48 | 1.87 | 2.30 |
| 120–80% – MV | 1.79 | 2.09 | 1.65 | 2.04 | 1.75 | 2.02 | 1.67 | 2.11 |
| 120–90% – EQ | 1.98 | 2.37 | 1.87 | 2.36 | 1.93 | 2.39 | 1.90 | 2.26 |
| 120–90% – MV | 1.88 | 2.15 | 1.72 | 2.00 | 1.74 | 2.00 | 1.81 | 2.03 |

**Table 2**
The $p$-values of SPA test with $N = \{5, 10, 15, 20\}$, $t_d = \{60, 120\}$, and $\gamma = \{60\%, 70\%, 80\%, 90\%\}$. Here, $\lambda = 1.0$ in the MV model, $SPA_1$ and $SPA_2$ represent the $p$-value in the training duration and testing duration, respectively.

| $td-\gamma$ | $N = 5$ | | 10 | | 15 | | 20 | |
|---|---|---|---|---|---|---|---|---|
| | $SPA_1$ | $SPA_2$ | $SPA_1$ | $SPA_2$ | $SPA_1$ | $SPA_2$ | $SPA_1$ | $SPA_2$ |
| 60–60% – EQ | 0.298 | 0.310 | 0.348 | 0.362 | 0.332 | 0.507 | 0.347 | 0.465 |
| 60–60% – MV | 0.189 | 0.138 | 0.146 | 0.200 | 0.126 | 0.221 | 0.185 | 0.232 |
| 60–70% – EQ | 0.246 | 0.236 | 0.189 | 0.343 | 0.250 | 0.397 | 0.238 | 0.353 |
| 60–70% – MV | 0.115 | 0.109 | 0.133 | 0.232 | 0.093* | 0.207 | 0.157 | 0.179 |
| 60–80% – EQ | 0.244 | 0.394 | 0.265 | 0.275 | 0.397 | 0.299 | 0.407 | 0.387 |
| 60–80% – MV | 0.098* | 0.301 | 0.116 | 0.210 | 0.149 | 0.225 | 0.110 | 0.139 |
| 60–90% – EQ | 0.099* | 0.194 | 0.251 | 0.236 | 0.270 | 0.228 | 0.241 | 0.241 |
| 60–90% – MV | 0.157 | 0.111 | 0.106 | 0.159 | 0.121 | 0.196 | 0.163 | 0.188 |
| 120–60% – EQ | 0.101 | 0.274 | 0.102 | 0.293 | 0.098* | 0.308 | 0.213 | 0.361 |
| 120–60% – MV | 0.021** | 0.402 | 0.029** | 0.260 | 0.033** | 0.246 | 0.082* | 0.246 |
| 120–70% – EQ | 0.174 | 0.280 | 0.137 | 0.338 | 0.183 | 0.290 | 0.216 | 0.332 |
| 120–70% – MV | 0.171 | 0.405 | 0.167 | 0.304 | 0.041** | 0.220 | 0.089* | 0.237 |
| 120–80% – EQ | 0.320 | 0.151 | 0.191 | 0.226 | 0.148 | 0.340 | 0.221 | 0.297 |
| 120–80% – MV | 0.280 | 0.275 | 0.127 | 0.252 | 0.189 | 0.205 | 0.149 | 0.221 |
| 120–90% – EQ | 0.287 | 0 .073* | 0.224 | 0.099* | 0.161 | 0.149 | 0.128 | 0.141 |
| 120–90% – MV | 0.213 | 0.062* | 0.197 | 0.126 | 0.130 | 0.149 | 0.113 | 0.216 |

* Significant at 0.1 (10%).
** Significant at 0.05 (5%).



(a) Portfolio value variation with $t_d = 60$.

(b) Portfolio value variation with $t_d = 120$.

**Fig. 6.** Portfolio value variation in the testing duration.

**Table 3**
The parameters of gene expression programming.

| | |
|---|---|
| Head length | 6 |
| Number of genes | 2 |
| Population size | 500 |
| Number of generations | 150 |
| Number of replications | 1 |
| Selection method | Tournament |
| Mutation rate | 60% |
| Transposition rate | 80% |
| Recombination rate | 80% |
| Inversion rate | 80% |

duration with $N = \{5, 10, 15, 20\}$, $t_d = 120$, and $\gamma = 60\%$ because positive returns can be obtained by these parameters without data snooping. According to Table 2, there is higher confidence to earn positive returns with these parameters in the testing duration.

The trading functions first are generated trading signals in the testing duration, and the corresponding actions are determined. Then we follow the action to determined buying, selling, replacing or waiting at interval $t$. In the end of the testing duration, the return and risk of our portfolio is computed. The results with the two capital allocation models are shown in Tables 4 and 5. Because the efficiency of GEP, several trading functions generate similar

**Table 4**
The cumulative returns and the average negative returns of the best 10 profitable trading functions with the EQ model in the testing duration, where $t_d = 120$, and $\gamma = 60\%$.

| Func. | $N = 5$ | | 10 | | 15 | | 20 | |
|---|---|---|---|---|---|---|---|---|
| | $R_{cum}$ (%) | $\mu_{neg}$ (%) | $R_{cum}$ (%) | $\mu_{neg}$ (%) | $R_{cum}$ (%) | $\mu_{neg}$ (%) | $R_{cum}$ (%) | $\mu_{neg}$ (%) |
| 1 | 143.14 | 2.23 | 90.48 | 2.12 | 70.74 | 2.30 | 66.37 | 2.19 |
| 2 | 171.23 | 2.24 | 117.03 | 2.11 | 96.88 | 2.28 | 92.33 | 2.16 |
| 3 | 171.23 | 2.24 | 117.03 | 2.11 | 96.88 | 2.28 | 92.33 | 2.16 |
| 4 | 135.82 | 2.23 | 101.70 | 2.11 | 83.44 | 2.30 | 79.29 | 2.23 |
| 5 | 171.23 | 2.24 | 117.03 | 2.11 | 96.88 | 2.28 | 92.33 | 2.16 |
| 6 | 171.23 | 2.24 | 117.03 | 2.11 | 96.88 | 2.28 | 92.33 | 2.16 |
| 7 | 141.61 | 2.24 | 106.84 | 2.12 | 87.12 | 2.31 | 82.87 | 2.24 |
| 8 | 171.23 | 2.24 | 117.03 | 2.11 | 96.88 | 2.28 | 92.33 | 2.16 |
| 9 | 143.14 | 2.23 | 90.48 | 2.12 | 70.74 | 2.30 | 66.37 | 2.19 |
| 10 | 141.61 | 2.24 | 106.84 | 2.12 | 87.12 | 2.31 | 82.87 | 2.24 |

**Table 5**
The cumulative returns and the average negative returns of the best 10 profitable trading functions with the MV model in the testing duration, where $t_d = 120$, and $\gamma = 60\%$.

| Func. | $N = 5$ | | 10 | | 15 | | 20 | |
|---|---|---|---|---|---|---|---|---|
| | $R_{cum}$ (%) | $\mu_{neg}$ (%) | $R_{cum}$ (%) | $\mu_{neg}$ (%) | $R_{cum}$ (%) | $\mu_{neg}$ (%) | $R_{cum}$ (%) | $\mu_{neg}$ (%) |
| 1 | 155.28 | 2.11 | 63.64 | 2.03 | 68.53 | 1.88 | 29.67 | 1.90 |
| 2 | 188.09 | 2.10 | 84.93 | 2.01 | 91.47 | 1.85 | 49.65 | 1.87 |
| 3 | 188.09 | 2.10 | 84.93 | 2.01 | 91.47 | 1.85 | 49.65 | 1.87 |
| 4 | 150.27 | 2.03 | 82.70 | 1.98 | 66.51 | 1.85 | 45.00 | 1.86 |
| 5 | 188.09 | 2.10 | 84.93 | 2.01 | 91.47 | 1.85 | 49.65 | 1.87 |
| 6 | 188.09 | 2.10 | 84.93 | 2.01 | 91.47 | 1.85 | 49.65 | 1.87 |
| 7 | 157.50 | 2.03 | 87.62 | 1.99 | 70.51 | 1.85 | 48.52 | 1.86 |
| 8 | 188.09 | 2.10 | 84.93 | 2.01 | 91.47 | 1.85 | 49.65 | 1.87 |
| 9 | 155.28 | 2.11 | 63.64 | 2.03 | 68.53 | 1.88 | 29.67 | 1.90 |
| 10 | 157.50 | 2.03 | 87.62 | 1.99 | 70.51 | 1.85 | 48.52 | 1.86 |

**Table 6**
The $p$-values of SPA test on the results produced by the best trading functions in the testing duration, where $t_d = 120$, and $\gamma = 60\%$.

| Func. | $N = 5$ | | 10 | | 15 | | 20 | |
|---|---|---|---|---|---|---|---|---|
| | $SPA_{EQ}$ | $SPA_{MV}$ | $SPA_{EQ}$ | $SPA_{MV}$ | $SPA_{EQ}$ | $SPA_{MV}$ | $SPA_{EQ}$ | $SPA_{MV}$ |
| 1 | $0.0110^{**}$ | $0.0040^{***}$ | $0.0170^{**}$ | $0.0330^{**}$ | $0.0340^{**}$ | $0.0210^{**}$ | $0.0335^{**}$ | $0.0815^{*}$ |
| 2 | $0.0055^{***}$ | $0.0040^{***}$ | $0.0125^{**}$ | $0.0185^{**}$ | $0.0185^{**}$ | $0.0135^{**}$ | $0.0170^{**}$ | $0.0415^{**}$ |
| 3 | $0.0025^{***}$ | $0.0020^{***}$ | $0.0100^{**}$ | $0.0250^{**}$ | $0.0135^{**}$ | $0.0110^{**}$ | $0.0190^{**}$ | $0.0410^{**}$ |
| 4 | $0.0075^{***}$ | $0.0060^{***}$ | $0.0155^{**}$ | $0.0180^{**}$ | $0.0260^{**}$ | $0.0225^{**}$ | $0.0185^{**}$ | $0.0375^{**}$ |
| 5 | $0.0050^{***}$ | $0.0035^{***}$ | $0.0095^{***}$ | $0.0215^{**}$ | $0.0160^{**}$ | $0.0180^{**}$ | $0.0165^{**}$ | $0.0430^{**}$ |
| 6 | $0.0015^{***}$ | $0.0020^{***}$ | $0.0120^{**}$ | $0.0190^{**}$ | $0.0250^{**}$ | $0.0120^{**}$ | $0.0180^{**}$ | $0.0440^{**}$ |
| 7 | $0.0060^{***}$ | $0.0035^{***}$ | $0.0165^{**}$ | $0.0195^{**}$ | $0.0255^{**}$ | $0.0195^{**}$ | $0.0235^{**}$ | $0.0480^{**}$ |
| 8 | $0.0055^{***}$ | $0.0010^{**}$ | $0.0065^{***}$ | $0.0200^{**}$ | $0.0215^{**}$ | $0.0195^{**}$ | $0.0200^{**}$ | $0.0445^{**}$ |
| 9 | $0.0055^{***}$ | $0.0060^{***}$ | $0.0180^{**}$ | $0.0335^{**}$ | $0.0345^{**}$ | $0.0285^{**}$ | $0.0440^{**}$ | $0.0820^{*}$ |
| 10 | $0.0085^{***}$ | $0.0035^{***}$ | $0.0210^{**}$ | $0.0160^{**}$ | $0.0235^{**}$ | $0.0205^{**}$ | $0.0305^{**}$ | $0.0510^{*}$ |

[*] Significant at 0.1 (10%).
[**] Significant at 0.05 (5%).
[***] Significant at 0.01 (1%).

signals, hence, they earn the same return, and have the same risk. It can be seen that all of these trading functions generated by GEP can earn positive return in the testing duration. This is because GTI is composed of average return of the tradable funds, and these funds in our portfolio are highly related to the trend of GTI. Hence the trading functions which can earn high return in trading GTI usually also earn high return in trading the portfolio. In addition, the MV model has lower risk because it allocates the capital with the most conservative way, but it still able to earn positive return.

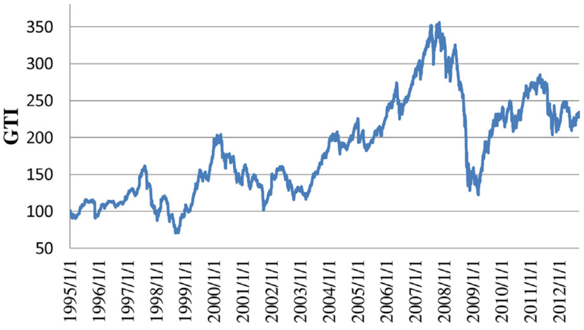


**Fig. 7.** GTI in the training duration and testing duration.

To ensure that our models earn returns without data snooping, we apply the SPA test to our experimental results of the 10 best trading functions. The $p$-values are shown in Table 6. As the results show, our method with the 10 best trading functions have statistical significance for earning return without data snooping. The buying and selling rules of the best trading function (functions 2, 3, 5, 6, and 8, we show function 5 here) generated by GEP are shown in Table 7. The variation of the portfolio values resulted from the best trading function in the testing duration is illustrated in Fig. 8. The highest annualized returns of the EQ and MV models are 12.08% and

**Table 7**
The reduced buying and selling rules generated by GEP for getting the best return in the training duration.

| | |
|---|---|
| Buying rule | $((MI^{w}_{t-9} > 0.3)$ and $(MI^{w}_{t-1} = 1))$ or<br>$((MI^{w}_{t-9} \leq 0.3)$ and $(MI^{w}_{t-1} = 0))$ |
| Selling rule | $MI^{w}_{t-9} = MI^{w}_{t-1}$ |

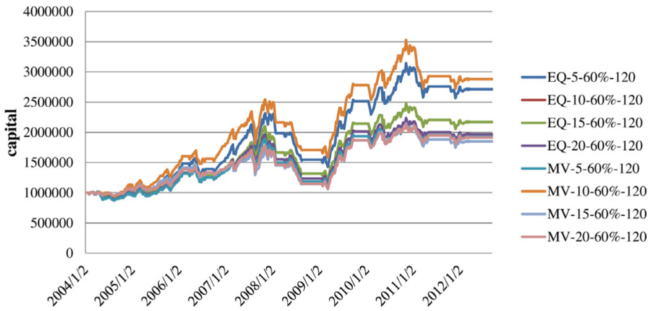


**Fig. 8.** The portfolio values of the best trading function in the testing duration.

**Table 8**
Return comparison of various methods from 2004/1/1 to 2008/12/31, where GA and GP represent the genetic algorithm and the genetic programming, respectively.

| Model name | $R_{cum}$ (%) | $AR_{cum}$ (%) |
|---|---|---|
| MSCI world price index | −11.21 | −2.35 |
| S&P 500 composite price index | −18.51 | −4.01 |
| Our method with EQ<br>(buy-and-hold strategy, $N = 5$, $\gamma = 90\%$, $t_d = 120$) | −0.74 | −0.14 |
| Our method with MV<br>(buy-and-hold strategy, $N = 5$, $\gamma = 90\%$, $t_d = 120$, $\lambda = 1.0$) | −3.13 | −0.63 |
| Our method with EQ<br>(best function, $N = 5$, $\gamma = 60\%$, $t_d = 120$) | 54.39 | 9.07 |
| Our method with MV<br>(best function, $N = 5$, $\gamma = 60\%$, $t_d = 120$, $\lambda = 1.0$) | 70.57 | 11.27 |
| Our method using GP with EQ<br>(best function, $N = 5$, $\gamma = 60\%$, $t_d = 120$) | 37.51 | 6.58 |
| Our method using GP with MV<br>(best function, $N = 5$, $\gamma = 60\%$, $t_d = 120$, $\lambda = 1.0$) | 38.69 | 6.76 |
| Tsai's method by GA<br>(constant weight, $N = 35$, $\gamma = 70\%$) | 39.53 | 6.89 |
| Tsai's method by GA<br>(constant weight, $N = 45$, $\gamma = 70\%$) | 37.35 | 6.55 |
| Tsai's method by GA<br>(constant weight, $N = 55$, $\gamma = 70\%$) | 33.78 | 5.99 |

12.85%, respectively, in the testing duration, where $N = 5$, $t_d = 120$, and $\gamma = 60\%$.

To compare the return performance with the method proposed by Tsai et al. [35], we adopt 2004/1/1 to 2008/12/31 as another testing duration, since they only showed the performance in this duration. $R_{cum}$ (cumulated return) and $AR_{cum}$ (annualized return) are compared in Table 8.

One can see that in Table 8, our methods, with EQ or MV model, earn higher return among these methods. In addition, we also employ the GP method, instead of GEP, in the two models by using the same parameters as Table 3, except that the population size is decreased to 150 because of the performance issue. The methods with GP get similar cumulative returns as Tsai's method. The MSCI world price index, S&P 500 composite price index, and the buy and hold strategy got loss in the duration. The main reason is that in 2008, all capital markets suffered from financial crisis, and almost all funds in the world got enormous loss. In contrast, GEP can be used to generate profitable trading functions, so our methods can still earn positive returns.

## 5. Conclusion

In this paper, we propose one effective method for investment in mutual funds. Our method consists of three stages: (1) determining the buying and selling timing with the trading functions generated by GEP; (2) constructing the portfolio formed by the funds with higher Sortino ratios; and (3) allocating the available capital among the selected funds with the equal (EQ) model or the mean variance (MV) model. Our method can effectively avoid the data snooping problem.

Investors may be conservative and they do not like to take high risk, hence, we adopt $\lambda = 1.0$ for testing the MV model. As shown in Tables 1, 4 and 5, we can see that our method with the MV model has lower risk than that with the EQ model in almost all cases. As the results shown in Tables 4–6, our method can statistically significantly earn positive return without data snooping in related parameters.

Compared with the method proposed by Tsai et al. [35], the annualized returns of our method with the EQ and MV models are 9.07% and 11.27%, respectively, from 2004/1/1 to 2008/12/31, which is better than the annualized return 6.89% achieved by Tsai et al. Our method with the MV model not only lowers the investment risk, but also gains the higher return than that of Tsai's method. Hence, our method with the MV model is suitable for the risk-averse investors.

In the future work, it is worth finding a more suitable way to allocate the capital, because investors may prefer some funds to others in their portfolio, and they may like to invest more capital in the preferred funds. In addition, the performance of the trading function may be raised by improving the function and terminal sets used in GEP.

## References

[1] M.S. Andersen, J. Dahl, L. Vandenberghe, CVXOPT: Python package for convex optimization, 2012 http://abel.ee.ucla.edu/cvxopt/
[2] A. Borrelli, I.D. Falco, A.D. Cioppa, M. Nicodemi, G. Trautteur, Performance of genetic programming to extract the trend in noisy data series, Physica A: Statistical Mechanics and its Applications 370 (2006) 104–108.
[3] A. Chaudhry, H.L. Johnson, The efficacy of the Sortino ratio and other benchmarked performance measures under skewed return distributions, Australian Journal of Management 32 (3) (2008) 485–502.
[4] B.-C. Chen, J.-Y. Lin, T.-P. Hong, Learning discriminant functions with fuzzy attributes for classification using genetic programming, Expert Systems with Application 23 (2002) 31–37.
[5] J. Chen, C. Chang, J. Hou, Y. Lin, Dynamic proportion portfolio insurance using genetic programming with principal component analysis, Expert Systems with Application 35 (2008) 273–278.
[6] M.A.H. Dempster, C.M. Jones, A real-time adaptive trading system using genetic programming, Quantiative Finance 1 (2001) 397–413.
[7] I.D. Falco, A.D. Cioppa, E. Tarantino, Discovering interesting classification rules with genetic programming, Expert Systems with Application 1 (2002) 257–269.
[8] E.F. Fama, Efficient capital markets: a review of theory and empirical work, The Journal of Finance 25 (1970) 383–417.
[9] C. Ferreira, Gene expression programming: a new adaptive algorithm for solving problems, Complex Systems 13 (2001) 87–129.
[10] FundDJ Co., Ltd., FundDJ, 2000 http://www.funddj.com/
[11] P.R. Hansen, A test for superior predictive ability, Journal of Business and Economic Statistics 23 (2005) 365–380.
[12] H. Huang, M. Pasquier, C. Quek, Financial market trading system with a hierarchical coevolutionary fuzzy predictive model, IEEE Transactions on Evolutionary Computation 13 (2009) 445–462.
[13] W. Huang, Y. Nakamori, S.-Y. Wang, Forecasting stock market movement direction with support vector machine, Computers & Operations Research 32 (2005) 2513–2522.
[14] H. Ince, T.B. Trafalis, Kernel principal component analysis and support vector machines for stock price prediction, in: IEEE International Joint Conference on Neural Networks Vol. 3, Budapest, Hungary, 2004 July, pp. 2053–2058.
[15] D. Kahneman, J.L. Knetsch, R.H. Thaler, Experimental tests of the endowment effect and the coase theorem, Journal of Political Economy 98 (1990) 1325–1348.
[16] D. Kahneman, J.L. Knetsch, R.H. Thaler, Anomalies: The endowment effect, loss aversion, and status quo bias, Journal of Economic Perspectives 5 (1991) 193–206.
[17] K.-J. Kim, I. Han, Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, Expert Systems with Application 19 (2000) 125–132.
[18] T. Kimoto, K. Asakawa, Stock market prediction system with modular neural networks, in: Proceedings of the International Joint Conference on Neural Networks, Vol. 1, Washington, USA, 1990 June, pp. 1–6.
[19] J. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, Massachusetts, USA, 1992.
[20] J. Koza, Genetic Programming II: Automatic Discovery of Reusable Programs, MIT Press, Cambridge, Massachusetts, USA, 1994.
[21] H.M. Markowitz, Portfolio selection, Journal of Finance 7 (1952) 77–91.
[22] H.M. Markowitz, Portfolio Selection: Efficient Diversification of Investments, John Wiley and Sons, New York, USA, 1959.

[23] R. O'Neil, pygep. Gene expression programming for python, 2007 http://code.google.com/p/pygep/

[24] P.-F. Pai, C.-S. Lin, A hybrid ARIMA and support vector machines model in stock price forecasting, OMEGA: The International Journal of Management Science 33 (2005) 497–505.

[25] J.-Y. Potvin, P. Soriano, M. Vallee, Generating trading rules on the stock markets with genetic programming, Computers & Operations Research 31 (2004) 1033–1047.

[26] W. Samuelson, R. Zeckhauser, Status quo bias in decision making, Journal of Risk and Uncertainty 1 (1988) 7–59.

[27] G. Sermpinis, J. Laws, A. Karathanasopoulos, C.L. Dunis, Forecasting and trading the EUR/USD exchange rate with gene expression and Psi Sigma neural networks, Expert Systems with Applications 39 (2012) 8865–8877.

[28] W.F. Sharpe, The sharpe ratio, Journal of Portfolio Management 21 (1994) 49–58.

[29] H. Shefrin, M. Statman, The disposition to sell winners too early and ride losers too long: theory and evidence, The Journal of Finance 40 (1988) 777–790.

[30] A. Shleifer, Inefficient Markets: An Introduction to Behavioral Finance, Oxford University Press, Oxford, GB, 2000.

[31] F.A. Sortino, L.N. Price, Performance measurement in a downside risk framework, The Journal of Investing 3 (1994) 59–64.

[32] R. Sullivan, A. Timmermann, H. White, Data-snooping, technical trading rule performance, and the bootstrap, Journal of Finance 54 (1999) 1647–1692.

[33] R.H. Thaler, Toward a positive theory of consumer choice, Journal of Economic Behavior & Organization 1 (1980) 39–60.

[34] A. Timmermann, C.W.J. Granger, Efficient market hypothesis and forecasting, International Journal of Forecasting 20 (2004) 15–27.

[35] T.J. Tsai, C.B. Yang, Y.H. Peng, Genetic algorithms for the investment of the mutual fund with global trend indicator, Expert Systems with Application 38 (2011) 1697–1701.

[36] Q. Wen, Z. Yang, Y. Song, P. Jia, Automatic stock decision support system based on box theory and SVM algorithm, Expert Systems with Application 37 (2010) 1015–1022.

[37] H. White, A reality check for data snooping, Econometrica 68 (2000) 1097–1126.

[38] C. Zhou, W. Xiao, T.M. Tirpak, P.C. Nelson, Evolving accurate and compact classification rules with gene expression programming, IEEE Transactions on Evolutionary Computation 7 (2003) 519–531.