ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220508486

Integrating a Piecewise Linear Representation Method and a Neural Network Model for Stock Trading Points Prediction

Article *in* IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews) · January 2009

DOI: 10.1109/TSMCC.2008.2007255 · Source: DBLP

CITATIONS
94

READS
1,341

3 authors:

Pei-Chann Chang
Yuan Ze University
228 PUBLICATIONS **6,043** CITATIONS

SEE PROFILE

Chin-Yuan Fan
National Applied Research Laboratories
69 PUBLICATIONS **1,477** CITATIONS

SEE PROFILE

Chen-Hao Liu
Kainan University
32 PUBLICATIONS **1,084** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Economics View project

Project IoT Patent Roadmap for Smart Logistic Service Provision in the Context of Industry 4.0. View project

All content following this page was uploaded by Pei-Chann Chang on 28 May 2014.

The user has requested enhancement of the downloaded file.

Integrating a Piecewise Linear Representation Method and a Neural Network Model for Stock Trading Points Prediction

Pei-Chann Chang, Chin-Yuan Fan, and Chen-Hao Liu

***Abstract*—Recently, the piecewise linear representation (PLR) method has been applied to the stock market for pattern matching. As such, similar patterns can be retrieved from historical data and future prices of the stock can be predicted according to the patterns retrieved. In this paper, a different approach is taken by applying PLR to decompose historical data into different segments. As a result, temporary turning points (trough or peak) of the historical stock data can be detected and inputted to the backpropagation neural network (BPN) for supervised training of the model. After this, a new set of test data can trigger the model when a buy or sell point is detected by BPN. An intelligent PLR (IPLR) model is further developed by integrating the genetic algorithm with the PLR to iteratively improve the threshold value of the PLR. Thus, it further increases the profitability of the model. The proposed system is tested on three different types of stocks, i.e., uptrend, steady, and downtrend. The experimental results show that the IPLR approach can make significant amounts of profit on stocks with different variations. In conclusion, the proposed system is very effective and encouraging in that it predicts the future trading points of a specific stock.**

***Index Terms*—Backpropagation neural network (BPN), financial time-series data, genetic algorithm (GA), piecewise linear representation (PLR), stock forecasting, trading points.**

I. Introduction

THE STOCK market is a highly nonlinear dynamic system. It is affected by many factors such as interest rates, inflation rates, economic environments, political issues, and many others. Although there are dependencies and correlations between these factors, the relationship of the stock price and these factors is rather difficult to model through mathematic formula.

Most recent research on stock forecasting has only been concerned with the prediction of price variation and with attempts to derive accurate models that are able to predict the future price of a stock movement, rather than the trading decision itself, such as buy/sell points regarding intelligent trading decision support. However, the trading decision plays a critical role in the stock market environment if an investor wants to make a profit. Up to now, there are only a few examples of research that deal with the turning points of a stock, and notably, this problem is still a large one as regards academic researchers and industrial practitioners [3].

Therefore, this research takes a different approach by applying piecewise linear representation (PLR) and backpropagation neural network (BPN) models to predict the turning point of a particular stock. Turning points of a target stock are more difficult to project than predicting the price variation of the stock itself. Since one does not have much information about the trading points (trough or peak) of a particular stock, it is necessary to rely on computational intelligence techniques applied to historical data of the stock to define this trading signal. In this research, PLR is applied to decompose historical data into different segments, and as a result, possible turning points of the historical stock data can be detected and inputted to the BPN to train the model. In addition, an intelligent PLR model is further developed by integrating the genetic algorithm (GA) with the PLR to improve the threshold value and to further increase the profitability of the model.

This paper studies the predictability and profitability of using PLR and BPN to predict the turning point of a stock price variation. The output from the neural network (NN) is transformed into a simple trading strategy, i.e., buy/hold/sell decisions. The proposed system consists of four steps: 1) input factors selection; 2) time-series data segmentation by PLR; 3) model building by training NNs using trading points identified by PLR; and 4) trading decision fine-tuned using IPLR by integrating GA with PLR for greater profits. Finally, financial time-series data from the Taiwan stock market and S&P500 are applied to demonstrate the effectiveness of the proposed system.

The rest of the paper is divided into five sections: Section II reviews the literature in the areas of stock forecasting; Section III describes the development of the IPLR model for stock trading point decision; Section IV examines the experimental tests conducted to determine the effectiveness of trading points generated by using the IPLR approach; lastly, conclusions and future directions of the research are provided.

II. Literature Survey

The prediction of stock price variation is a very difficult task as the price movement seems to behave in a very random manner. During the last decade, stock and future traders have largely been relying on various types of intelligent systems to make trading decisions. Recently, artificial NNs (ANNs) have been

Manuscript received August 6, 2007; revised December 20, 2007, March 7, 2008, and June 15, 2008. First published December 2, 2008; current version published December 22, 2008. This paper was recommended by Associate Editor M.-H. Lim.

P.-C. Chang is with the Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan (e-mail: iepchang@saturn.yzu.edu.tw).

C.-Y. Fan is with the Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan 32026, Taiwan (e-mail: S948906@mail.yzu.edu.tw).

C.-H. Liu is with the Department of Digital Technology, Kainan University, Taoyuan 33857, Taiwan (e-mail: chliu@mail.knu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2008.2007255

1094-6977/$25.00 © 2008 IEEE

Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.

applied to this area (refer to [4], [10], [14], [17], [19], [30], [31], [37], and [39]). However, limitations of these models are due to noise and complex dimensionality of stock price data. That is to say that the quantity of data itself and the input variables interfere with each other. Therefore, results were usually not as convincing.

Other soft computing (SC) methods are also applied in the prediction of stock prices. These SC approaches employ quantitative inputs, like technical indexes, and qualitative factors, like political effects, to automate stock market forecasting and trend analyses. A feedforward NN trained by a GA [34] is applied to forecast three-month U.S. Treasury Bill rates. They conclude that an NN can be used to accurately predict these rates. A neurofuzzy model [10] is applied to predict future values of Taiwan stock price variation. Thammano [44] conclude that the neurofuzzy architecture is able to recognize the general characteristics of the stock market faster and more accurately than the basic backpropagation algorithm. Tansel *et al.* [26] compare the ability of linear optimization, ANN, and GA to model time-series data using the criteria of modeling accuracy, convenience, and computational time. They find that linear optimization methods give the best estimates though the GA can provide the same values if the boundaries of the parameters and the resolution are selected appropriately, but NNs produce the worst estimations. However, they note that nonlinearity can be accommodated by both GA and NN because the latter requires minimal theoretical background. Baba *et al.* [5] employs NN and GA to construct an intelligent decision support system (DSS) for analyzing the Tokyo Stock Price Index (TOPIX). The essential feature of their DSS is that it projects the high and low TOPIX values four weeks into the future and suggests buy and sell decisions based on the average projected value and the current value of the TOPIX. Liao and Tsao [36] apply a fuzzy NN (FNN) combined with a chaos-search GA (CGA) and simulated annealing (SA) to short-term power system load forecasting as a sample test.

In addition, GA is also incorporated to improve the learning and generalizability of ANNs for stock market prediction [14]. Daily predictions are conducted and prediction accuracy is measured. A novel NN-based method for time-series forecasting is presented in [15] and they combine the optimal partition algorithm with a radial basis function NN. The method is applied to stock price prediction and the results of numerical simulations demonstrate the effectiveness of the method. A cluster-based combinatorial forecasting scheme is developed in [16] for a single-step-ahead prediction of the pound–dollar daily exchange rates and this hybrid method demonstrates an improvement over conventional linear and neural-based combinatorial schemes. Kimoto and Asakawa [17] present an improved NN and fuzzy models used for exchange rate prediction. The method has been compared to multilayer perceptrons (MLPs), radial basis functions, dynamic neural networks, and neurofuzzy systems. A hybridized SC technique for automated stock market forecasting and trend analysis is investigated in [18]. They use principal component analysis to preprocess the input data, an NN for one-day-ahead stock forecasting and a neurofuzzy system for analyzing the trend of the predicted stock values. Wang *et al.* [38] propose a GA-based fuzzy knowledge integration framework that can simultaneously integrate multiple fuzzy rules sets and their membership function sets. Results show that the fuzzy knowledge base derived using our approach performs better than every individual knowledge base. A Takagi–Sugeno–Kang (TSK)-type fuzzy-rule-based system is developed in [19] for stock price prediction. The TSK fuzzy model applies the technical index as the input variables and the consequence is a linear combination of the input variables. The fuzzy-rule-based model is tested on Taiwanese electronic shares from the Taiwan Stock Exchange (TSE). As shown through the intensive experimental tests, the model has successfully forecasted the price variation for stocks from different sectors with accuracy close to 97.6% in TSE index and 98.08% in MediaTek.

To our knowledge, little research has focused on developing trading signals for effective stock trading. A stock trading method based on dynamic Bayesian networks is applied to model the dynamics of the trend of stock prices [41] and it is a three-level hierarchical hidden Markov model. The inferred probability distribution of first level is used as an indicator for the trading signal. A novel stock market model is developed [42] in which the orders of the chartists are not only based on past prices but also on past trading volume. This model has the potential to replicate some important stylized facts about stock markets, especially bubbles and crashes, excess volatility, fat tails, uncorrelated price increments, and volatility clustering. An intelligent stock trading decision support system that can forecast the buying and selling signals according to the prediction of short-term and long-term trends using rule-based NNs is developed in [43]. The NN component is composed of two rule-based NNs and they are used to predict the price trend of stocks. A hybrid neurogenetic approach [45] is developed for stock trading. A recurrent NN having one hidden layer is used for the prediction model. The input features are generated from a number of technical indicators being used by financial experts. The GA optimizes the NN's weights under a 2-D encoding and crossover. The model has been tested with 36 companies in New York Stock Exchange and National Association of Securities Dealers Automated Quotations for 13 years from 1992 to 2004. The neurogenetic hybrid shows notable improvement on the average over the buy-and-hold strategy and the context-based ensemble further improves the results.

Another approach based on the notion that trading strategies guided by forecasts of the direction of price movement is developed in [20]. The effectiveness of a financial time-series forecasting strategy is investigated in [40] by exploiting the multiresolution property of the wavelet transform. A financial series is decomposed into a complete, shift-invariant scale-related representation. In transform space, each individual wavelet series is modeled by a separate MLP. The probabilistic NN is used to forecast the direction of index return after it is trained based on the historical data.

As discussed earlier, a considerable amount of research has been conducted to study the behavior of a stock price movement. However, the investor is likely to be more interested in making profit by being provided with simple trading decisions such as buy/hold/sell from the system rather than predictions of the stock price itself. Therefore, this research focuses on developing

Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.

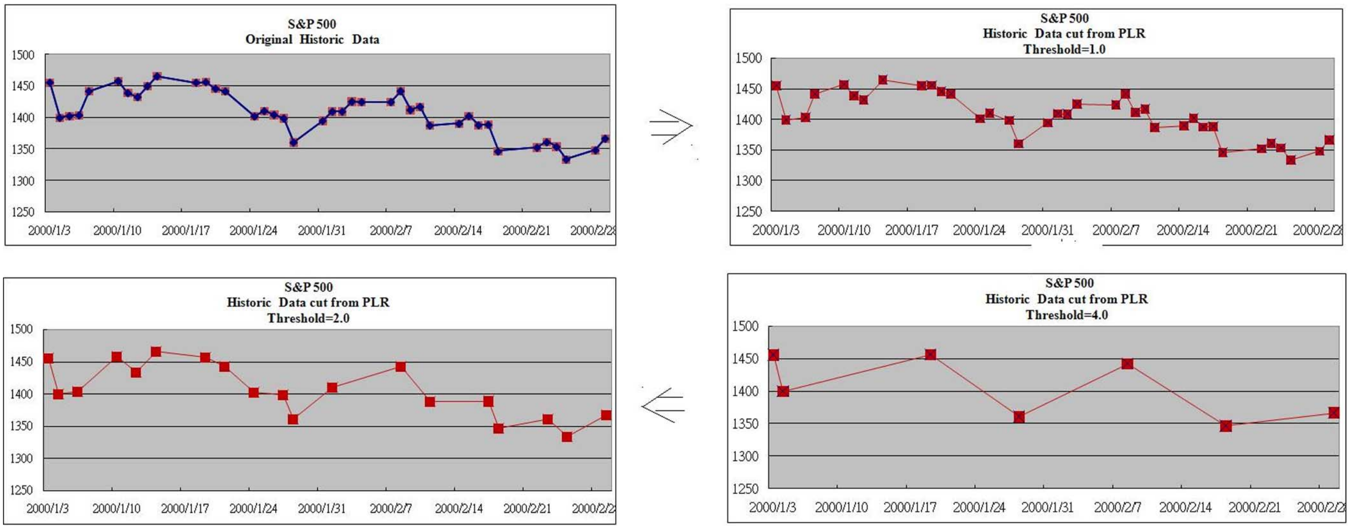


Fig. 1. Using PLR to generate possible trading points.

an intelligent stock trading DSS by providing trading signals to investors according to the stock price variations.

III. Development of an IPLR Model for Stock Trading Point Prediction

The evolution of stock prices over time can be seen as a short-term random oscillation on top of a long-term trend. The trading strategy is to ride on a general trend, and at the same time, enhance profits by capturing the likely short-term tendencies.

Traditionally, financial experts have proposed a set of trading rules for investors and the set of rules are based on moving average crosses, head and shoulders, range breakout, triangle breakout, etc. For example, for the buy signal, the stock has to show a long-term uptrend, the momentum indicators have to indicate that the stock is oversold, and the relative strength index (RSI) has to be low, indicating the starting signs of bounce-back. For the sell signal, the stock has to show a long-term downtrend. Momentum indicators have to signal that the stock is overbought, that the RSI is high, and that the price is close to the local peak and showing the first signs of decline.

However, these rules can be treated as buy/hold/sell strategies and the effectiveness of these rules is still quite limited as they may still miss out on a lot of trading opportunities. Recently, PLR has been applied to pattern matching [35]; this study takes a different approach by using PLR to decide the trough or peak of the historical data, and based on these trading points, a BPN model is built through supervised learning to forecast the future trading point of a specific stock.

A. Piecewise Linear Representation

From the previous study, trading points of stocks are observed from the variation of technical indexes. However, the variations are not necessarily easy to observe for every investor. Therefore, this study attempts to develop an intelligent trading point prediction system so that investors can implement a good trading strategy by applying this intelligent model. Recently, PLR has been applied for pattern matching [35]. The proposed model will adopt PLR to decide the trough or peak of the historical data, and based on these trading points, a BPN model is built to forecast the future trading point of a specific stock. The main procedures of PLR in predicting the trading point are described as follows:

1) Divide the Raw Data Into Subsegments: PLR can be applied as a sliding window to find the turning points of financial time-series data. This method requires the setup of the size of the sliding window, and the delineation of the relative maximum or minimum points of the data within the time window as the turning points. However, this may not be a very good approach, as the turning points depend upon what the sliding window size is. If the window size is not properly decided upon, the subsegments generated by PLR may lead to wrong decisions for future trading points. Therefore, the profitability of the method is questionable. Alternately, this research takes all selected historical data for piecewise representation, and then, the turning points decided upon will be more representative and can be applied for the prediction of future trading points. Another key factor to be decided is the threshold value of a piecewise representation that will be explained in the following.

2) Setup the Threshold Value (δ) of PLR: A larger threshold value creates longer trend patterns; contrarily, the patterns are very sensitive when the threshold value is very small. This research adopts GAs as fine-tuning operators for this key factor–threshold value (δ). Depending on the variation of each stock in the historical data, the threshold value is set up within the range of [0.01, 5.0] accordingly.

3) Trading Point Decision: PLR is developed for pattern matching; it cannot automatically generate the trading points of a stock for investors. The contribution of this research is to convert outputs from PLR to generate possible trading points. Take S&P 500 index for example where PLR generates a segmentation diagram according to the threshold value applied. As shown in Fig. 1, using PLR to generate possible trading points, the first graph shows the original time series while the rest of the graphs

Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.

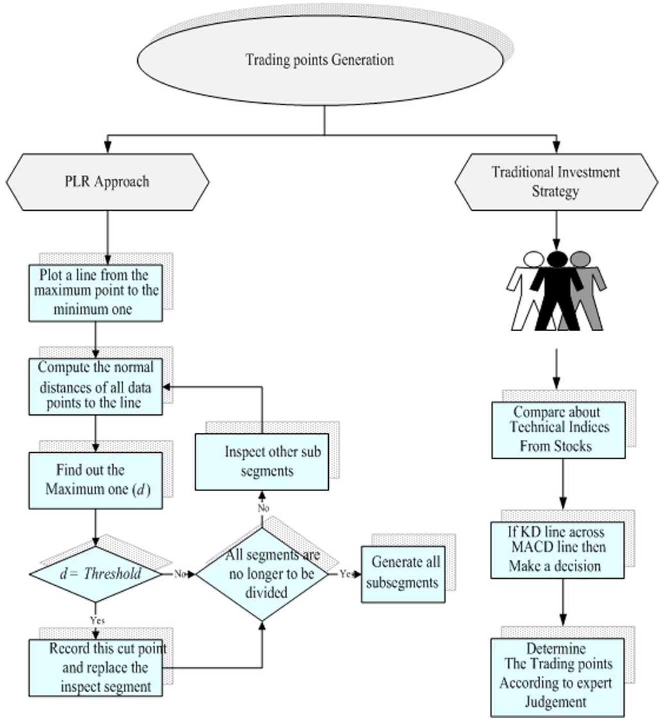


Fig. 2. Detailed processes of trading point's generation by IPLR and financial experts.

```
Procedure BuildTree (S).
Input: A financial time series S.
Let S be represented as x[1..n], y[1..n].
If (Max (y[1..n] == y[1] OR Max (y[1..n] == y[n])
OR Min (y[1..n] == y[1] OR Min (y[1..n] == y[n]))
Create a node in the hierarchy for this segment;
Draw a line between (x[1],y[1]) and (x(n), y(n));
Max d = maximum Euclidean distance of (x[i],y[i]) to the line;
    If (Max d < threshold ( δ ))
    This segment is good enough; no further work
    Else
Let (x[j],y[j]) be the point with maximum Euclidean distance to the line.
        Break the segment S into S1 and S2 at the point (x[j],y[j]);
        PARENT (S1) = S;
        PARENT (S2) = S;
        BuildTree (S1);
        BuildTree (S2);
    End If
  Else
```

Fig. 3. Virtual code of the PLR.

are generated by using different threshold values in PLR. As can be observed in the diagram, the higher the threshold value the smaller the number of segments generated. Each segment represents a local trough or peak, and these local extremes are transformed into trading signals. For a threshold value of 1.0, there are roughly 36 trading signals while there are only 7 trading signals for a threshold value of 4.0. These trading signals are inputted to BPN for supervised learning. However, there is no guarantee as regards which value will be better. That is why we set up five threshold values for PLR and choose the one that generates the most profit when applied to our data.

To compare the difference between experts' decisions and the PLR approach, the detailed processes of trading point's generation by PLR and financial experts are shown in Fig. 2.

In addition, the pseudocodes of trading point's generation by PLR are listed in Fig. 3.

The quality of the trading points generated by PLR is determined by the threshold value (δ). In traditional application, the judgment of a trading signal was decided by financial experts' opinions. Thus, the financial experts based their opinions regarding trading decisions on the technical analysis and in relation to the economic situation. Thereby, if the trading signal is not judged properly, the investor can make a mistake and lose money. However, in our research, these trading decisions can be directly linked to the threshold value (δ). Therefore, this study applies GA to optimize the threshold value of PLR instead, with the expectation to generate better subsegments, and to make more profits, as explained in Fig. 2.

B. Time-Series Data Segmentation by IPLR

An IPLR model is developed by integrating the GA with the PLR to improve the threshold value of PLR. The main purpose of adopting GA is to find the best threshold value in deciding the trading points by PLR so as to increase the accuracy of trading point predictions. As a result, the overall profitability of the model can be further improved. Detailed procedures of IPLR are shown in Fig. 4.

1) Candidate Stocks Screening: A set of candidate stocks is selected based on the following criteria: 1) capital size; 2) monthly sales; 3) earnings per share (EPS); 4) transaction volume per day; and 5) marginal cost of capital (MCC). It is a preferred situation when a stock has a small or medium size of capital, increasing monthly sales (best with historic high), high EPS, large volumes of stock transactions (the stock is very popular in the market), and if the marginal cost of capital is low. This is because in the Taiwanese stock market, the prices of large capital stock are always fluctuating very slowly. According to recent market statistics, a small- or medium-size market has a better profit margin due to the probability that the annual growth and stock price variation will be more significant during the year.

Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.

TABLE I
TECHNICAL INDEXES USED AS INPUT VARIABLES

Technical index	Technical Index (input in our system)	Explanation
Moving Average(MA)	5MA,6MA, 10MA, 20MA	Moving averages are used to emphasize the direction of a trend and smooth out price and volume fluctuations that can confuse interpretation.
Bias (BIAS)	5BIAS, 10BIAS	The difference between the closing value and moving average line, which uses the stock price nature of returning back to average price to analyze the stock market.
relative strength index (RSI)	6RSI ,12RSI	RSI compares the magnitude of recent ins to recent losses in an attempt to determine overbought and oversold conditions of an asset
nine days Stochastic line (K, D)	KD	The stochastic line K and line D are used to determine the signals of over-purchasing, over-selling, or deviation.
Moving Average Convergence and Divergence (MACD)	9 MACD	MACD shows the difference between a fast and slow exponential moving average (EMA) of closing prices. Fast means a short-period average, and slow means a long period one.
Williams %R (pronounced "percent R")	12W%R	Williams %R is usually plotted using netive values. For the purpose of analysis and discussion, simply ignore the netive symbols. it is best to wait for the security's price to change direction before placing your trades.
Transaction Volume	Transaction Volume	Transaction volume is a basic yet very important element of market timing strategy. Volume provides clues as to the intensity of a given price move.
Differences of technical index (Δ)	Δ5MA , Δ6MA, Δ10MA, Δ5BIAS , Δ10BIAS, Δ6RSI, Δ12RSI, Δ12W%R, Δ9K, Δ9D, Δ9 MACD	Differences of technical index between t day and t+1 day.

2) *Input Variables Selection:*

a) Generating an initial threshold: An initial threshold is generated randomly for PLR in time-series data segmentation.

b) Use PLR to segment the stock data: PLR is used to segment the selected stock data using the initial threshold generated in step 1. The stock data segmented are transformed into trading signals.

The segmentation algorithm uses a sliding window with varying sizes. The sliding window contains, at most, m points, beginning after the last identified end point and ending directly before the current point. If there are more than m points between the last end point and the current point, only the last m points are contained in the sliding window. The segmentation tries to find a possible upper or lower point in the current sliding window. An upper point is defined as follows (the definition of a lower point is symmetric and is omitted here):

Suppose the current point is $P_j(X_j, t_j)$. The upper point $P_i(X_i, t_i)$ is a point in the current sliding window that satisfies:

1) $X_i = Max$ (*X values of current sliding window*);
2) $X_i > X_j + \delta$;
3) $P_i(X_i,\ t_i)$ *is the last one satisfying the aforesaid conditions*.

where δ is the given threshold for segmentation.

c) Input variables selection using stepwise regression analysis (SRA): As shown in Table I, a set of technical indexes affecting the stock price movement have been identified by Chang *et al.* [10]. These input factors are further selected using the stepwise regression analysis (SRA) model. Stepwise regression analysis is applied to determine the set of independent variables that most closely affect the dependent variable. This is accomplished by repeating the process of variable selection. The step-by-step procedure of the SRA approach is explained in detail in the following:

Step 1: Calculate the correlation coefficient (r) of every input variable ($X_1, X_2, \ldots, X_n$), i.e., technical indexes, and output data (Y), i.e., trading signal. Set all numbers in a correlation matrix.

Step 2: Choose the largest r^2 value from the correlation matrix (suppose X_i is the largest one in the current stage), and derive a regression model, i.e., $\hat{Y} = f(X_i)$; then, consider the correlation of Y with other input data. (Assuming X_j is statistically significant; α value is applied to consider the significance of each input variable.)

Step 3: Calculate partial F value from other input data, as shown in (1), and choose the largest correlation coefficient among these input variables (assume it is X_j). Then, derive another regression model $\hat{Y} = f(X_i, X_j)$ again

$$\mathrm{SSR} = \sum \left(\hat{Y}_i - \bar{Y}\right)^2$$

$$\mathrm{SSE} = \sum \left(\hat{Y}_i - Y_i\right)^2 \tag{1}$$

$$F_j^* = \frac{\mathrm{MSR}(X_j \mid X_i)}{\mathrm{MSE}(X_j,\ X_i)} = \frac{\mathrm{SSR}/1(X_j \mid X_i)}{\mathrm{SSE}/(n-2)(X_j,\ X_i)}, \quad i \in I \tag{2}$$

Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.

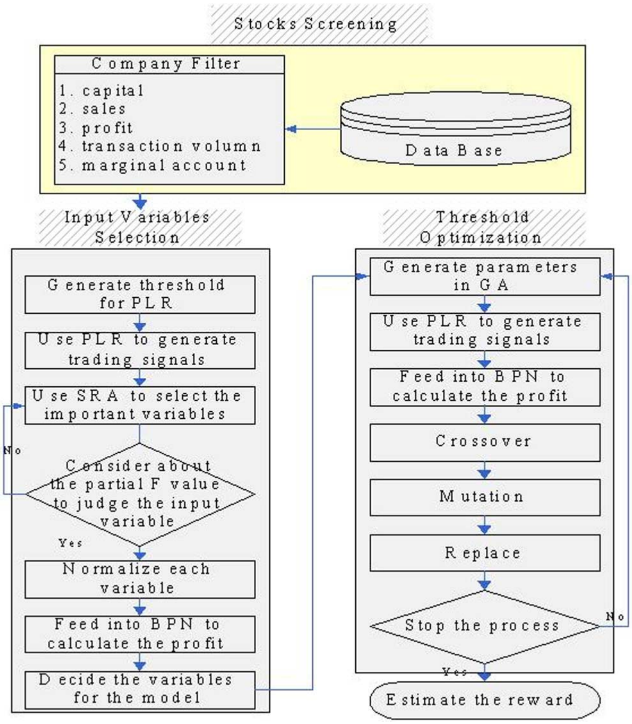


Fig. 4. Flowchart of the IPLR.

Step 4: Calculate the partial F value of the original data for input variable X_j. If the value is smaller than a user-defined threshold, it is removed from the model since X_j is not statistically significant for the output.

Step 5: Repeat step 3 to step 4. If every input variable's partial F value is greater than the user-defined threshold, then stop. It means that every input value should have a significant influence on the output value. According to [7] and [9], if F value of a specific variable is greater than the user-defined threshold, it is added to the model as a significant factor. When F value of a specific variable is smaller than a user-defined threshold, it is removed from the model. The statistical software Statistics Package for Social Science (SPSS) for Windows version 10.0 is applied for a stepwise regression analysis in this research.

d) Normalize the selected variables from SRA: The selected variables from SRA are normalized as follows:

$$X_i' = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}. \tag{3}$$

e) Normalized variables input to BPN for trading point prediction: The set of normalized variables and trading points are inputted to BPN for training the connection weight. Once the model is trained, it can be applied to future trading point predictions.

3) Threshold Optimization: During the variable optimizing process, PLR is preprocessed using five initial thresholds (5.0, 1.0, 0.5, 0.1, and 0.01). Different threshold values, when applied in PLR, will generate different patterns of trading signals. The threshold value with the largest profit among these five is selected, and the corresponding input variables are decided for IPLR simultaneously. The detailed procedures as regards IPLR are explained in the following.

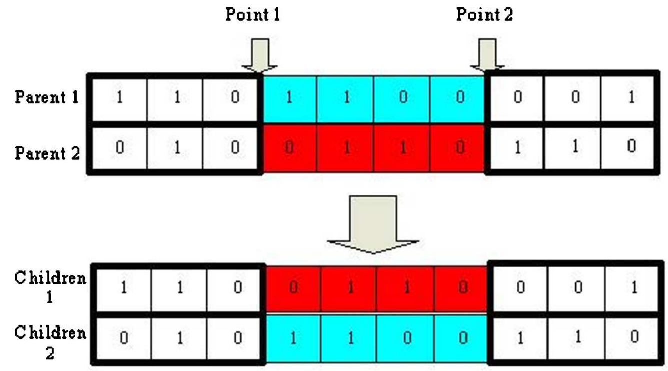


Fig. 5. Two point's crossover.

a) Step 1 (coding): Binary code is adopted here. To determine the degree of accuracy, chromosome length needs to be decided after the design of the experiments.

b) Step 2 (generate initial solutions): Initial solutions of the threshold values are generated by randomly assigning 0 or 1 to each gene of the chromosome, and in total, we will generate 50, i.e., the number of the population for each generation.

c) Step 3 (calculate the fitness value): For each chromosome, i.e., the threshold value, PLR is applied to segment the time-series data into subsegments. Then, related trading signals are generated and inputted to BPN for supervised training. After training, BPN is employed to generate future trading signals on these testing data. The total profit of each stock is calculated according to the formula as follows:

$$\text{Profits} = A \prod_{i=1}^{k} \left\{ \left[\frac{(1-a-b) \times C_{S_i} - (1+a) \times C_{B_i}}{(1+a) \times C_{B_i}} \right] \right\} \tag{4}$$

where A is the total amount of money to be invested at the beginning; a refers to the tax rate of the ith transaction (for the Taiwanese stock exchange market), b refers to the handling charge of the the ith transaction, k refers to the total number of transactions in the current chromosome, C_{S_i} is the selling price of the ith transaction, and C_{B_i} is the buying price of the ith transaction.

d) Step 4 (representation and selection): The tournament method is used in this study.

e) Step 5 (crossover): The two-point crossover method is applied here, and is represented as Fig. 5.

f) Step 6 (mutation): The two-point mutation method, as shown in Fig. 6, is adopted in the research.

g) Step 7 (replacement): If the fitness value of current chromosome is better than any one generated from the previous generation, replace it.

h) Step 8 (terminate): If the number of generations reaches the limit, then terminate the process and output the best result; otherwise go to step 3.

The most useful capability of GA in IPLR is to search for the best threshold value for PLR in order to generate more suitable

Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.

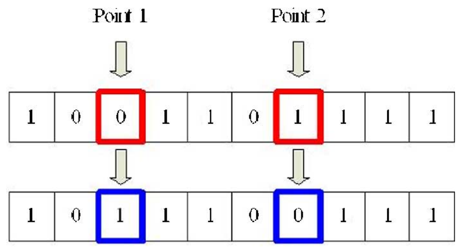


Fig. 6. Two point's mutation.

trading signals to feed into the BPN. The BPN will be described in the following section.

C. Backpropagation Network

After generating the subsegments through PLR, the trading signals need to be transformed into 0 or 1 before they are fed into the BPN. By adopting PLR, the subsegments are divided and the trends of time-series data are defined as follows:

$$\begin{aligned}&\text{If } C_i \geq C + \delta, \quad \text{trend} = \text{upward}\\&\text{If } C_i \leq C - \delta, \quad \text{trend} = \text{downward}\\&\text{If } C_i - \delta < C < C_i + \delta, \quad \text{trend} = \text{steady}\end{aligned} \tag{5}$$

where C_i is the stock price of the ith turning point; and C is the stock price of the current turning point; $\times$ is the threshold for judging the trend of the stock price and the trend is transferred as an output value of BPN. If the trend changes from up to down, the trading signal is changed from 0 to 1; if the trend changes from down to up, the trading signal is changed from 1 to 0; otherwise, the signal does not change. The trading signals of AU Optronics Corporation (AUO) are illustrated in a table.

However, the aforesaid definition of trading signals is not quite related to the price variation. A trading signal should be able to reflect the price variation and provide more detailed information for the investor to make a precise decision for stock trading. Therefore, we redefine the trading signals according to the tendency of the stock and this is shown in (6) and (7).

If a stock is on uptrend

$$t_i = \left[\frac{C_i - \min\{C_i, C_{i+1}, C_{i+2}\}}{\max\{C_i, C_{i+1}, C_{i+2}\} - \min\{C_i, C_{i+1}, C_{i+2}\}}\right] \times 0.5. \tag{6}$$

If a stock is on downtrend

$$t_i = \left(\left[\frac{C_i - \min\{C_i, C_{i+1}, C_{i+2}\}}{\max\{C_i, C_{i+1}, C_{i+2}\} - \min\{C_i, C_{i+1}, C_{i+2}\}}\right] \times 0.5\right) + 0.5 \tag{7}$$

where C_i means the stock price on the ith transaction day. This new definition of trading signals, we believe, properly represents the momentum of a stock price. The trading signals in Table II are recalculated and they are shown in Table III. Instead of 0 or 1, this new trading signal is more informative, and as it is in the range of 0 to 1, it can provide more insightful information related to the movement of the stock price.

TABLE II
TRADING SIGNALS OF AUO

Time series	Stock price	Turning point	Trend	Trading point	Trading signal
1	46.2	•	Up	Buy	0
2	49.1				0
3	48.2	•	Up	Buy	0
4	54.3				0
5	56.6	•	Down	Sell	1
6	53.5				1
7	54.6				1
8	50.2				1
9	48.8	•	Up	Buy	0
10	51.5	•			0

TABLE III
TRANSFORMATION OF t_i

Time series	Stock price	Trading signal	t_i
1	46.2	0	0
2	49.1	0	0.07377
3	48.2	0	0
4	54.3	0	0.129032
5	56.6	1	1
6	53.5	1	0.875
7	54.6	1	1
8	50.2	1	0.759259
9	48.8	0	-
10	51.5	0	-

All t_i values and related input variables, i.e., technical indexes, are fed into the BPN for training the connection weights.

The BPN and the supervised learning, i.e., learned by samples, are chosen to train the forecasting process. After learning (or training), the trained weight can be employed for the prediction of future occurrences. The structure of the BPN consists of three layers: input, hidden, and output layers, as shown in Fig. 7. Each layer contains I, J, and K nodes denoted, respectively, by circles. The node is also called neuron or unit. The circles are connected by links, denoted by arrows, each of which represents a numerical weight. w_{ij} denotes numerical weights between input and hidden layers, and so does w_{jk}, between hidden and output layers, as also shown in Fig. 7. The processing or computation is performed in each node in the hidden and output layers. As for the number of layers and number of nodes, they will be further decided in relation to the design of the experiment.

The backpropagation learning algorithm consists of two procedures: 1) a feedforward step and 2) a backpropagation weight training step.

During the training process, input variables to the BPN are selected by the stepwise regression model and so each stock has different input variables. The input technical index is selected from a pool of 23 indexes in combination, as shown in Table I.

After the training process, the output of the BPN for a particular set of inputs, i.e., the technical index is automatically

Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.

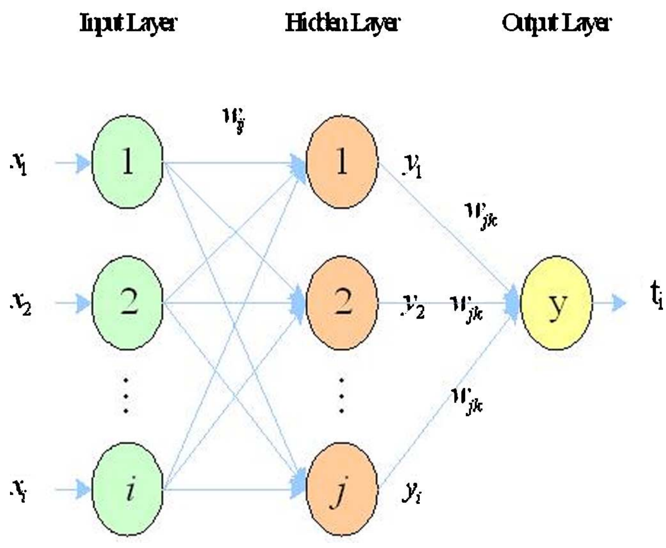


Fig. 7. Structure of backpropagation neural network.

transformed into a trading signal according to (6) or (7). However, in real-world application, since the technical index for the current day is not available until the closing of the day, i.e., 1:30 P.M. Taiwan time, once a trading signal is generated, the stock market has been closed already. As such, it is not possible to buy or sell stock at this moment. To avoid this problem, the buy or sell price for a trading signal is decided based on the next day's opening price instead. There is only a minor difference between the current day's closing price and the following day's opening price.

To make a trading decision, there should be a metarule to track the trend and decide when to trade. In this study, the average of t_i in the training stage is regarded as a boundary condition when making a judgment about the trading decision. For example, 0.508 is the boundary condition in this case. A simple example from Green Point Corporation (GP) is applied to demonstrate this working principle. For a financial time series with ten stock data from GP, the threshold is set up as 0.01, 0.1, 0.5, 1.0, and 5.0 and according to SRA input variables selected for each threshold value. This is shown in Table IV.

According to the inputs 12RSI, Δ6MA, Δ9K, and Δ12RSI, when the threshold value is 1.0, the corresponding output is a transformed trading signal. This dataset is inputted to the BPN for supervised training. After the training process, a new set of test data is inputted to the BPN to generate a set of outputs, i.e., trading signals, as shown in Table V. They are the outputs from the BPN including the stock price and the corresponding trading signals.

These outputs in Table V can be further transformed into a figure by showing the relationship between trading signals and output values from the BPN and this is shown in Fig. 8.

As shown in Fig. 8, there are six turning points detected in Table V. The trading decision is made once a change of the trading signal passes through the boundary value, i.e., 0.508 in this case. If the change is upward, a selling decision is made. On the contrary, if the change is downward, a buying decision is made. The profitability of these trading decisions is calculated according to (4) and the profit of each transaction is accumulated to provide a calculation of final total profits.

TABLE IV
INPUT VARIABLES FOR EACH THRESHOLD VALUE IN GP STOCK

Threshold (δ) value	Input Variables by SRA to be inputted to BPN x_i value
5.0	x_1 =**5BIAS**
	x_2 =**10BIAS**
	x_3 =**12RSI**
	x_4 =**Δ5BIAS**
	x_5 =**9MACD**
	x_6 =**12W%R**
	x_7 =**Δ9MACD**
1.0	x_1 =**12RSI**
	x_2 =**Δ6MA**
	x_3 =**Δ9K**
	x_4 =**Δ12RSI**
0.5	x_1 =**9K**
	x_2 =**12RSI**
	x_3 =**Δ12RSI**
	x_4 = **12W%R**
0.1	x_1 =**Δ6MA**
	x_2 =**Δ12RSI**
0.01	x_1 =**Δ6MA**
	x_2 =**Δ12RSI**

TABLE V
FINAL TRADING DECISION FROM BPN

Time series	Stock price	t_i from BPN	Trading decision
1	81.8	0.647	Buy
2	82.9	0.408	Sell
3	82.3	0.616	Buy
4	85.5	0.436	
5	88.4	0.486	Sell
6	87.5	0.558	
7	86.3	0.691	
8	87.6	0.676	Buy
9	89.3	0.507	
10	90.6	0.501	Sell

IV. NUMERICAL EXAMPLES

In this section, nine different stocks are selected for performance comparisons of the IPLR model. Three of them are on the uptrend (AUO, Epistar Corporation (EPISTAR), GP, three are on the downtrend (Silicon Integrated System Corporation (SiS), SENAO International Corporation (SENAO), D-Link Corporation (D-LINK), and others are steady (Foxlink Corporation (FOXLINK), Compal Corporation (COMPAL), UMC

Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.

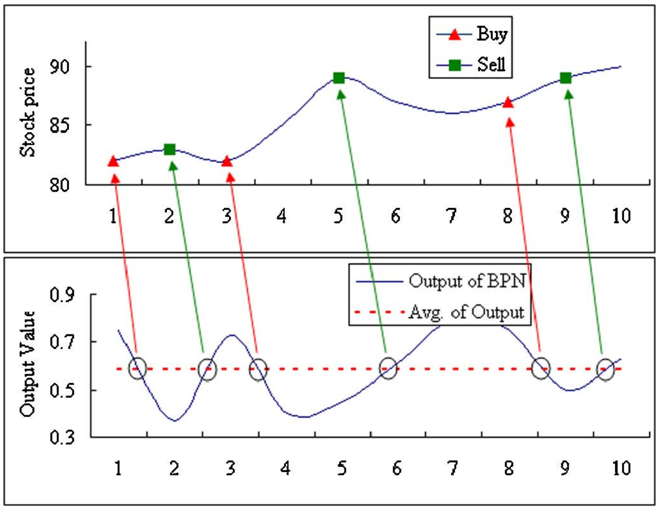


Fig. 8. Relation between trading signals and output values from BPN.

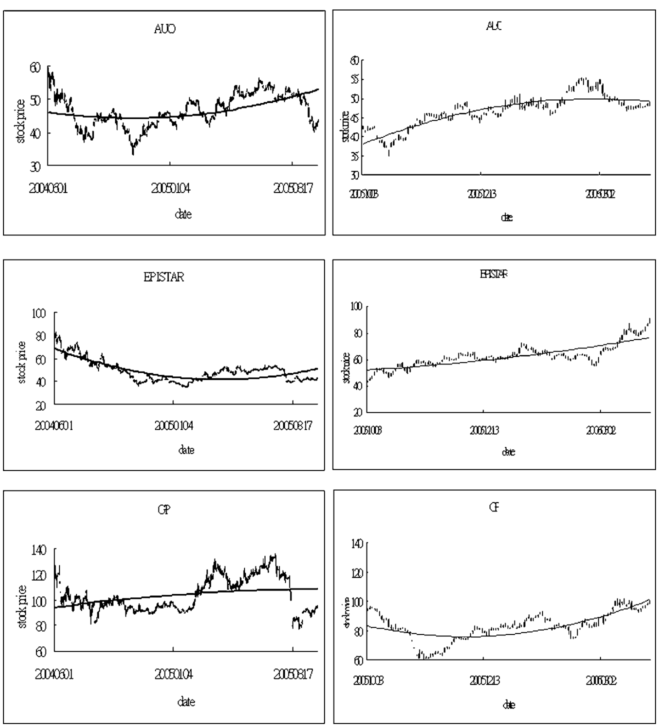


Fig. 9. Three stocks in up-trend.

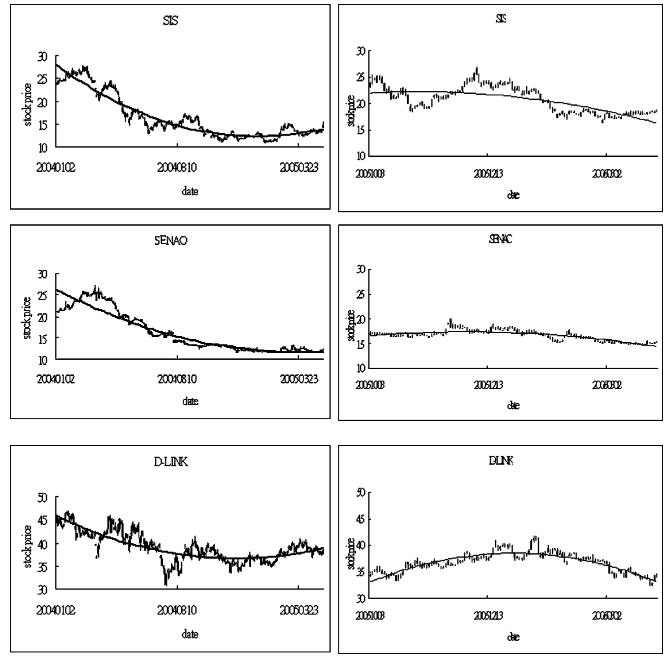


Fig. 10. Three stocks in down-trend.

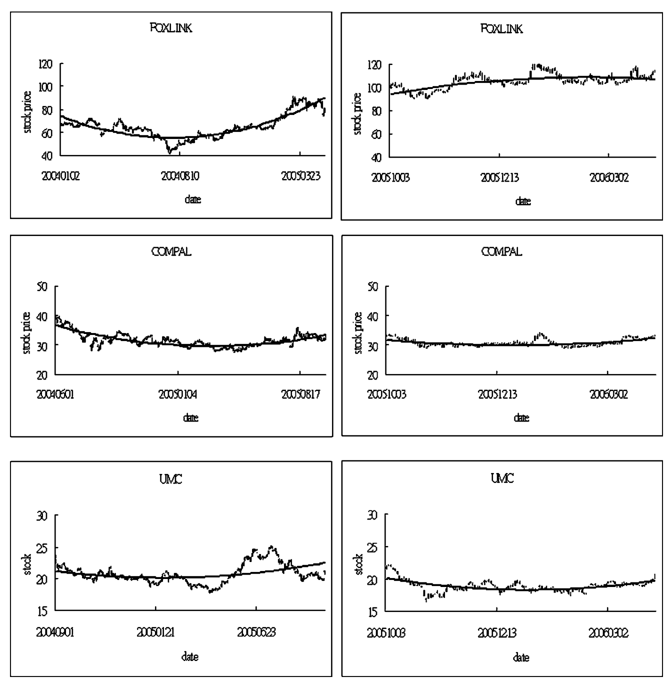


Fig. 11. Three stocks in steady state.

Corporation (UMC)). In the following, these stocks in different tendencies are shown in Figs. 9–11. The historic data covers the financial time-series data from 2004/01/02 to 2006/04/12 and the training data is based on the data from 2004/01/02 to 2005/9/30. As for the test data, it is drawn from 2005/10/01 to 2006/04/12.

The rule applied to decide whether a stock is in an uptrend or downtrend state is made according to the tendency of the moving average. For an uptrend stock, its 30-day moving average will cross over its 90-day moving average. On the contrary, as regards a downtrend stock, its 30-day moving average will cross under its 90-day moving average. There is no major tendency of 30-day moving average with 90-day moving average for a steady stock.

The main purpose of instance selection is to identify whether there are major performance differences among these three types of stocks in our IPLR model. However, even if the downtrend stock is selected; investors can also make good trading decisions if the proposed model is good enough.

A. Input Variables for IPLR and Parameter Setting of BPN

Input variables for the IPLR are based on the best profit from the five threshold values, i.e., 0.01 up to 5.0. If threshold 1.0

Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.

TABLE VI
PARAMETER SETUP FOR BPN BY DOE (DESIGN OF EXPERIMENTAL)

PARAMETER	BEST
# of neuron in hidden layer	7
Transfer function	Sigmoid
Learning rule	Delta Rule
Learning rate	0.5
Momentum	0.5
# of learning times	1000

TABLE VII
INPUT VARIABLES FOR DIFFERENT THRESHOLD VALUES BY SRA FOR PLR

Initial threshold (δ)	Selected factors	Profits
5.0	5BIAS, 10BIAS, 12RSI, Δ5BIAS, 9MACD, 12W%R, Δ9MACD	95,157
1.0	12RSI, Δ6MA, Δ9K, Δ12RSI	35,165
0.5	9K, 12RSI, Δ12RSI, 12W%R	2,211
0.1	Δ6MA, Δ12RSI	3,201
0.01	Δ6MA, Δ12RSI	3,388

TABLE VIII
TREATMENT LEVEL OF EACH FACTOR IN GA

Factors	Levels
# of generation	50, 100
# of population	5, 10
Crossover rate (Pc)	0.8, 1.0
Mutation rate (Pm)	0.1, 0.05

generates the maximum profit among them, the input variables to the BPN are then selected for the IPLR. To simplify the process, these input variables remain the same throughout the whole evolutionary process since it is very troublesome if another variable selection process is to be restarted again.

The parameter setting of the BPN, including the number of hidden layer, transfer function, and learning rule, etc., will be decided by the design of experimental (DOE). These parameters are important because they will affect the system performance if they are not properly adjusted. The final setup of the parameters for the BPN is listed in Table VI after the DOE experimental tests.

For our case sample in GP stock, the best threshold value is 5.0 with maximum profit and the input variables selected are 5BIAS, 10BIAS, 12RSI, Δ5BIAS, 9MACD, 12W%R, and Δ9MACD, as shown in Table VII. These input variables are applied in IPLR procedures and they are not changed at all throughout the whole evolutionary processes.

B. Best Parameters Setting of GA in IPLR

To further improve the performance of the PLR, the genetic algorithm is applied to further fine-tune the threshold value. However, there are several parameters to be set up for the approach; the number of generations, the number of population, crossover rate, and the mutation rate. The experiments are repeated ten times for each different combination. To analyze the impact of each factor, the variable analysis is undertaken for each factor in order to identify the significant factors. The final results are shown in Table VIII.

TABLE IX
ANALYSIS OF VARIABLES

Source	Degree Of freedom	Sum Square	Mean of Sum squre	F value	P value
Generation	1	5.17E+08	5.17E+08	1.06	0.305
Population	1	5.39E+09	5.39E+09	11.03	0.001
Pc	1	9.17E+08	9.17E+08	1.88	0.173
Pm	1	4.58E+08	4.58E+08	0.94	0.335
Error	155	7.57E+10	4.89E+08		
Total	159	8.3E+10			

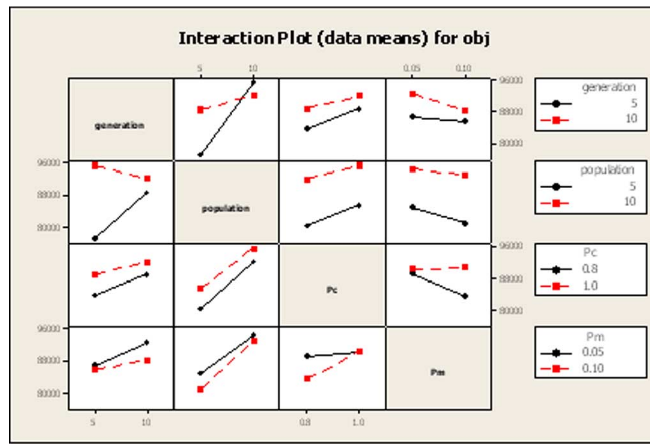


Fig. 12. Interaction plot of different factors.

F-value and P-value (see Table IX) are two statistically observed significance levels. A parameter is of significant influence when its F-value is greater than $F(k-1, n-k)$ where n is the total number of factors investigated and k is the number of sources. P-value is the smallest fixed level at which the null hypothesis can be rejected. If the P value is less than 0.05, it would reject the null hypothesis. As shown in Fig. 12 (the interaction plot of different factors), the best combination of these factors is generation number (50), population number (10), crossover rate (1.0), and mutation rate (0.05), respectively, and the profit is the highest among these combinations.

C. Comparisons of Different Prediction Models

After setting up the parameters of the experiments, we compared the proposed IPLR method with three existing methods. These three different algorithms to be compared are the rule-based BPNs by Chou *et al.* [43], a Trading solutions software package [48], and the PLR-BPN approach. Through a series of experimental tests, we find that IPLR consistently generates the highest profit among all and PLR-BPN comes second for those nine different stocks. The overall comparisons of these three different models on these nine different stocks in terms of rate of return are shown in Table X.

In addition, S&P 500 index is also applied for comparing the performance of these three different approaches [43], [48]. The S&P 500 index data include four-year data, i.e., 2000–2003, for

Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.

TABLE X
OVERALL COMPARISON OF RULE-BASED BPN, PLR, AND IPLR ON THESE NINE DIFFERENT STOCKS IN TERMS OF RATE OF RETURN

Stocks	Input Technical Index	Rule Based BPN	PLR	IPLR	Best Threshold for IPLR
GP	5BIAS, 10BIAS, 12RSI, Δ5BIAS, 9MACD, 12W%R, Δ9MACD	94%	95%	123%	4.50
AUO	5BIAS, 12RSI	90%	47%	146%	4.97
EPISTAR	5BIAS	282%	406%	408%	0.15
COMPAL	5MA, 20MA, 12W%R	38%	43%	67%	1.48
UMC	20MA, Δ12W%R	78%	74%	95%	1.50
FOXLINK	Δ10BIAS, Δ6RSI	73%	147%	150%	0.68
SENAO	9MACD, 12W%R	47%	39%	46%	0.19
SIS	Δ9K, 12W%R	65%	139%	162%	0.82
D-LINK	Δ9K, Δ9MACD	51%	50%	79%	0.21

TABLE XI
OVERALL COMPARISONS OF RULE-BASED BPN, TRADING SOLUTION [48], AND IPLR ON S&P500

S&P 500 Index	Rule-Based BPN	Trading Solutions[48]	IPLR
Total number of Trades signals	16	6	32
Total number of Trades (includes buy & sell)	8	3	16
Number of profitable Trades	3	3	23
Percentage of profitable Trade	37.5%	100%	65.7%
Annual Rate of per profit	>1%	11%	35.7%

training, and 9-month-period data, i.e., January to September 2004, for testing. The profit of each model is shown in Table XI. Once again, IPLR generates the highest profit among all. Trading points generated by each model are shown in Figs. 13–15. We also take into consideration the trading fee, i.e., 40 U.S. dollars per transaction.

D. Discussion

The GA is applied to search for the best threshold value during the evolutionary process; however, the computational time to evaluate fitness is significant. That is why the GA evolving process is designed with only 50 generations. Another reason is that the output from the PLR has been pretty good already; therefore, the evolutionary processes can quickly converge after certain iterations. The reason for applying GA is because of its good global search ability. Of course, we also compare the GA approach with the simple multistart search and the results are shown in Table XII. S&P500 index data is applied for training and testing. As observed in Table XII, GA still performs better than the multistart search approach in terms of profit making with only a small loss of computational time.

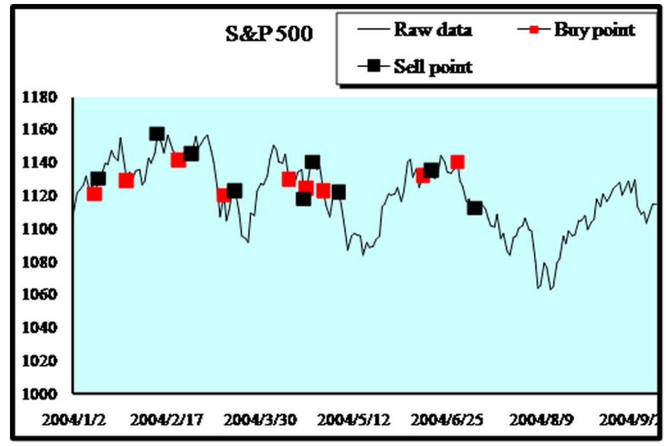


Fig. 13. Trading points from rule-based BPN.

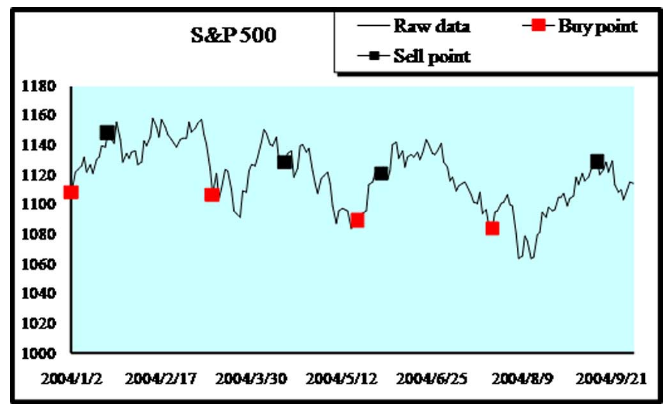


Fig. 14. Trading points from trading solutions.

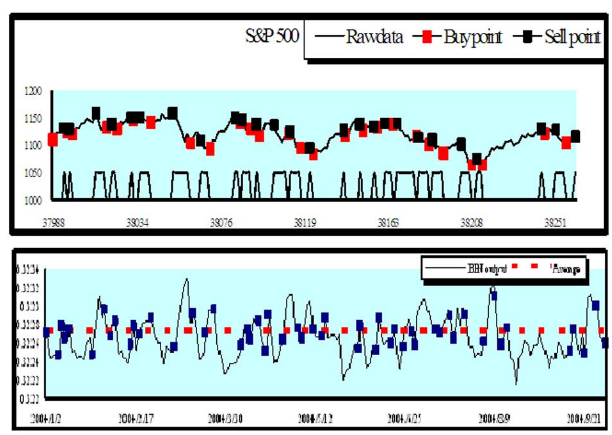


Fig. 15. Trading points from IPLR.

The comparison between GA and a simple multistart search shows that GA can further improve the profit by 16.5% in S&P 500 over the multistart approach. Of course, GA uses more computational time in searching for a better solution. However, time is spent off-line in training the BPN. The training process is only needed when the neural network is applied for the first time. It may take a couple of days to perform a retraining when needed.

Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.

TABLE XII
COMPARISONS OF GA AND SIMPLE MULTISTART SEARCH FOR S&P500 INDEX

S&P 500 Index	Multi Start	G.A
Total number of generations generation	100	100
Total time to find the Best solution (minutes)	150	180
Threshold	3.5	3.542
The best solution (Profit%)	30%	35.7%
Improve Rate		16.2%

TABLE XIII
COMPARISONS OF GA AND SIMPLE MULTISTART SEARCH FOR AUO AND SIS AND UMC

	Multi-Start			G.A		
	AUO	SIS	UMC	AUO	SIS	UMC
Total number of generations generation	100	100	100	100	100	100
Total time to find the Best solution (minutes)	157	132	132	168	179	160
Threshold	1.0	1.0	2.0	4.97	0.82	1.5
The best solution (Profit%)	44%	139%	74%	146%	162%	95%
Improve Rate				97%	16.5%	28.3%

Therefore, the computational time can be justified by making more profits (average 47.27% for uptrend-AUO, steady-UMC, and downtrend-SIS stocks), as shown in Table XIII, where the training time is off-line.

V. CONCLUSION

A considerable amount of research has been conducted to study the behavior of stock price movement. However, the investor is more interested in making profit by being provided simple trading decisions such as buy/hold/sell from the system rather than determining the stock price itself. Therefore, a different approach is made by applying PLR to decompose the historical data into different segments. As a result, turning points (trough or peak) of the historical stock data can be detected and then be input to the BPN to train the connection weight of the model. Then, a new set of input data can trigger the model when a buy or sell point is detected by the BPN. An intelligent PLR model is further developed by integrating the GA with the PLR. This improves the threshold value of PLR to further increase the profitability of the model.

The IPLR model is tested on three different types of stocks, i.e., uptrend, steady, and downtrend. The experimental results show that the IPLR approach can make a significant amount of profit especially on uptrend and downtrend states rather than steady state. In summary, the proposed system is very effective and encouraging in its predictions regarding the future trading points of a specific stock. However, there is one issue to be further discussed and that is the price variation of the stock. It is observed that if the price variation of the current stock is to be forecasted either in an uptrend or a downtrend, then it is better that we train our BPN with a similar pattern, i.e., either in a similar uptrend or downtrend period.

In the future, the proposed system can be further investigated by incorporating other SC techniques or by providing a better forecasting model other than BPN. They are listed as follows:

1) *Clustering of financial time-series data*: Data preprocessing is one of the features that can be applied in financial time-series data processing. Effective clustering of time-series data can further improve the forecasting accuracy of the forecasting system. But how these data are to be clustered and what input factors are needed are interesting issues to be further investigated?
2) *A different forecasting model*: There are numerous forecasting models other than the BPN model in the literature. It is important to study the behavior of these models when applied to predictions of the stock's trading point. Different input factors and different forecasting models, such as support vector machine, FNN, and case-based reasoning (CBR) are possible candidates for improving the accuracy of the proposed model.
3) *A similar training pattern*: The performance of BPN is quite sensitive to the set of training data. It is more appropriate if one can retrieve the set of data with a similar pattern to the testing data from the historical data. The profitability of the system should be further improved. Therefore, a CBR technique can be applied in retrieving the set of training data with similar patterns to the set of data to be forecasted.

REFERENCES

[1] A. Abraham, N. Baikunth, and P. K. Mahanti, "Hybrid intelligent systems for stock market analysis," in *Proc. Int. Conf. Comput. Sci.*, 2001, pp. 337–345.
[2] A. Abraham, N. S. Philip, and P. Saratchandran, "Modeling chaotic behavior of stock indices using intelligent paradigms," *Int. J. Neural, Parallel Sci. Comput.*, vol. 11, pp. 143–160, 2003.
[3] Y. S. Abu-Mostafa and A. F. Atiya, "Introduction to financial forecasting," *Appl. Intell.*, vol. 6, pp. 205–213, 1996.
[4] M. Aiken and M. Bsat, "Forecasting market trends with neural networks," *Inf. Syst. Manage.*, vol. 16, no. 4, pp. 42–48, 1994.
[5] N. Baba, N. Inoue, and H. Asakawa, "Utilization of neural networks and GAs for constructing reliable decision support systems to deal stocks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2000, vol. 5, pp. 111–116.
[6] D. Brownstone, "Using percentage accuracy to measure neural network predictions in stock market movements," *Neurocomputing*, vol. 10, pp. 237–250, 1996.
[7] P. C. Chang and T. W. Liao, "Combing SOM and fuzzy rule base for flow time prediction in semiconductor manufacturing factory," *Appl. Softw. Comput.*, vol. 6, no. 2, pp. 198–206, 2006.
[8] P. C. Chang and Y. W. Wang, "Fuzzy Delphi and back-propagation model for sales forecasting in PCB industry," *Expert Syst. Appl.*, vol. 30, no. 4, pp. 715–726, 2006.
[9] P. C. Chang, C. H. Liu, and Y. W. Wang, "A hybrid model by clustering and evolving fuzzy rules for sale forecasting in printed circuit board industry," *Decis. Support Syst.*, vol. 42, no. 3, pp. 1254–1269, 2006.
[10] P. C. Chang, Y. W. Wang, and W. N. Yang, "An investigation of the hybrid forecasting models for stock price variation in Taiwan," *J. Chin. Inst. Ind. Eng.*, vol. 21, no. 4, pp. 358–368, 2004.
[11] P. C. Chang and C. H. Liu, "A TSK type fuzzy rule based system for stock price prediction," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 135–144, 2008.
[12] A. S. Chen, M. T. Leung, and H. Daouk, "Application of neural networks to an emerging financial market: Forecasting and trading the taiwan stock index," *Comput. Oper. Res.*, vol. 30, pp. 901–923, 2003.

Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.

[13] M. Y. Chen and D. A. Linkens, “Rule-base self-generation and simplification for data-driven fuzzy models,” *Fuzzy Sets Syst.*, vol. 142, pp. 243–265, 2004.

[14] S. C. Chi, H. P. Chen, and C. H. Cheng, “A forecasting approach for stock index future using grey theory and neural networks,” in *Proc. IEEE Int. Joint. Conf. Neural Netw.*, Washington, DC, Jul. 1999, vol. 6, pp. 3850–3855.

[15] K. Izumi and K. Ueda, “Analysis of exchange rate scenarios using an artificial market approach,” in *Proc. Int. Conf. Artif. Intell.*, 1999, pp. 360–366.

[16] K. J. Kim and I. Han, “Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index,” *Expert Syst. Appl.*, vol. 19, pp. 125–132, 2000.

[17] T. Kimoto and K. Asakawa, “Stock market prediction system with modular neural network,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, San Diego, CA, 1990, vol. 1, pp. 1–6.

[18] H. L. Larsen and R. R. Yager, “A framework for fuzzy recognition technology,” *IEEE Trans. Syst. Man, Cybern. C, Appl., Rev.*, vol. 30, no. 1, pp. 65–76, Feb. 2000.

[19] J. W. Lee, “Stock price prediction using reinforcement learning,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Pusan, Korea, 2001, vol. 1, pp. 690–695.

[20] K. Papagiannaki, N. Taft, Z. L. Zhang, and C. Diot, “Long-term forecasting of Internet backbone traffic,” *IEEE Trans. Neural Netw.*, vol. 16, no. 5, pp. 1110–1124, Sep. 2005.

[21] K. Parasuraman and A. Elshorbagy, “A wavelet networks: An alternative to classical neural networks,” in *Proc. IEEE Int. Join. Conf. Neural Netw.*, Jul. 31–Aug. 4, 2005, vol. 5, pp. 2674–2679.

[22] A. Popoola, S. Ahmad, and K. Ahmad, “Multi-scale wavelet preprocessing for fuzzy systems,” in *Proc. Int. Congr. Comput. Intell. Methods Appl.*, Dec. 15–17, 2005, pp. 15–17.

[23] R. Sitte and J. Sitte, “Analysis of the predictive ability of time delay neural networks applied to the S&P 500 time series,” *IEEE Trans. Syst. Man, Cybern. C, Appl. Rev.*, vol. 30, no. 4, pp. 568–572, Nov. 1999.

[24] M. C. Su, C. W. Liu, and S. S. Tsay, “Neural-network-based fuzzy model and its application to transient stability prediction in power systems,” *IEEE Trans. Syst. Man, Cybern. C, Appl. Rev.*, vol. 29, no. 1, pp. 149–157, Feb. 1999.

[25] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its application to modeling and control,” *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.

[26] I. N. Tansel, S. Y. Yang, A. Venkataraman, W. Y. Bao, and N. Mahendrakar, “Modeling time series data by using neural netwroks and genetic algorithms,” in *Proc. Artif. Neural Netw. Eng. Conf.*, 1999, pp. 1055–1060.

[27] A. Thammano, “Neuro-fuzzy model for stock market prediction,” in *Proc. Artif. Neural Netw. Eng. Conf.*, 1999, pp. 587–591.

[28] L. X. Wang and J. M. Mendel, “Generating fuzzy rules by learning form examples,” *IEEE Trans. Syst. Man, Cybern.*, vol. 22, no. 6, pp. 1414–1427, Nov./Dec. 1992.

[29] S. Wang and N. P. Archer, “A neural network based fuzzy set model for organizational decision making,” *IEEE Trans. Syst. Man, Cybern. C, Appl. Rev.*, vol. 28, no. 2, pp. 194–203, May 1998.

[30] J. Yao and H. L. Poh, “Forecasting the KLSE index using neural networks,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Perth, WA, Australia, Nov./Dec.1995, vol. 2, pp. 1012–1017.

[31] Y. Yoon and J. Swales, “Prediction stock price performance: A neural network approach,” in *Proc. 24th Annu. Int. Conf. Syst. Sci.*, Kauai, HI, Jan. 8–11, 1991, vol. 2, pp. 156–162.

[32] Y. Lixin and Y.-Q. Zhang, “Evolutionary fuzzy neural networks for hybrid financial prediction,” *IEEE Trans. Syst. Man, Cybern. C, Appl. Rev.*, vol. 35, no. 2, pp. 244–249, May 2005.

[33] G. P. Zhang, “Avoiding pitfalls in neural network research,” *IEEE Trans. Syst. Man, Cybern. C, Appl. Rev.*, vol. 37, no. 1, pp. 3–16, Jan. 2007.

[34] R. Graham, “Neural networks for real-time path finding in computer games,” *ITB J.*, vol. 9, pp. 223–230, May 2004.

[35] H. Wu, B. Salzberg, and D. Zhang, “Online event-driven subsequence matching over financial data streams,” in *Proc. SIGMOD, Stream Manag.*, Jun. 2004, pp. 23–34.

[36] G.-C. Liao and T.-P. Tsao, “Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting,” *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 330–340, Jun. 2006.

[37] P. G. Harrald and M. Kamstra, “Evolving artificial neural networks to combine financial forecasts,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 40–52, Apr. 1997.

[38] C. H. Wang, T. P. Hong, and S. S. Tseng, “Integrating fuzzy knowledge by genetic algorithms,” *IEEE Trans. Evol. Comput.*, vol. 2, no. 4, pp. 138–149, Nov. 1998.

[39] L. J. Cao and F. E. H. Tay, “Support vector machine with adaptive parameters in financial time series forecasting,” *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1506–1518, Nov. 2003.

[40] B. L. Zhang, R. Coggins, M. A. Jabri, D. Dersch, and B. Flower, “Multiresolution forecasting for futures trading using wavelet decompositions,” *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 1506–1518, Jul. 2001.

[41] O. Jangmin, J.-W. Lee, S.-B. Park, and B.-T. Zhang, “Stock trading by modelling price trend with dynamic Bayesian networks,” in *Intelligent Data Engineering and Automated Learning.* Lecture Notes in Computer Science, vol. 3177. Berlin, Germany: Springer-Verlag, 2004, pp. 794–799.

[42] F. H. Westerhoff, “Technical analysis based on price-volume signals and the power of trading breaks,” *Int. J. Theory Appl. Finance*, vol. 9, no. 2, pp. 1–18, 2006.

[43] S.-C. T. Chou, C.-C. Yang, C.-H. Chan, and F. Lai, “A rule-based neural stock trading decision support system,” in *Proc. IEEE. Conf. Comput. Intell. Finan. Eng.*, New York, Mar. 24–26, 1996, pp. 148–154.

[44] A. Thammano, “Neuro-fuzzy model for stock market prediction,” in *Proc. Artif. Neural Netw. Eng. Conf.*, New York, 1999, pp. 587–591.

[45] Y.-K. Kwon and B.-R. Moon, “A hybrid neurogenetic approach for stock forecasting,” *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 851–864, May 2007.

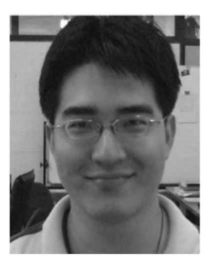
[46] K. V. Nesbitt and S. Barrass, “Finding trading patterns in stock market data,” *IEEE Comput. Graph. Appl.*, vol. 24, no. 5, pp. 45–55, Sep. 2004.

[47] D. Whitely, T. Starkweather, and D. Fuquay, “Scheduling problems and traveling salesman: The genetic edge recombination operator,” in *Proc. 3rd. Int. Conf. Genet. Algorithms*, 1989, pp. 133–140.

[48] NeuroDimension, Inc. (2008). Trading solution company [Online]. Available: http://www.tradingsolutions.com/

Pei-Chann Chang received the M.S. and Ph.D. degrees from Lehigh University, Bethlehem, PA, in 1985 and 1989, respectively.

He is currently a Professor at the Department of Information Management, Yuan-Ze University, Taoyuan, Taiwan. He is the Senior Editor of the *Journal of the Chinese Institute of Industrial Engineering*. He is the author or coauthor of more than 60 publications in international journals. His current research interests include financial time-series forecasting, evolutionary computation, fuzzy neural applications, production scheduling, forecasting, case-based reasoning, and applications of soft computing.



Chin-Yuan Fan received the B.S. degree in management from the Chinese Culture University, Taipei, Taiwan, in 2001, and the M.S. degree in management from Da-Yeh University, Chang-Hwa, Taiwan, 2003. Currently, he is working toward the Ph.D. degree at the Department of Industrial Engineering and Management,Yuan-Ze University, Taoyuan, Taiwan.

His research interests include applications of soft computing, financial time-series forecasting, multiobjective optimization problems, and multicriteria decision making.

Chen-Hao Liu received the Ph.D. degree from Yuan-Ze University, Taoyuan, Taiwan, in 2007.

He is currently an Assistant Professor at the Department of Digital Technology, Kainan University, Taoyuan. His research interests include production scheduling, evolutionary algorithms, multiobjective optimization problems, and financial time-series forecasting.

View publication stats Authorized licensed use limited to: Hong. Downloaded on February 7, 2009 at 23:52 from IEEE Xplore. Restrictions apply.