Expert Systems with Applications 36 (2009) 10097–10104

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques

Ritanjali Majhi [a], G. Panda [b,*], Babita Majhi [b], G. Sahoo [c]

[a] School of Management, National Institute of Technology, Warangal, India
[b] Dept. of Electronics and Communication Engineering, National Institute of Technology, Rourkela, Orissa 769008, India
[c] Dept. of Computer Science and Engineering, BIT, Mesra, India

A R T I C L E I N F O

Keywords:
Stock market forecasting
Bacterial foraging optimization
Adaptive bacterial foraging optimization
Genetic algorithm and particle swarm optimization

A B S T R A C T

The present paper introduces the use of BFO and ABFO techniques to develop an efficient forecasting model for prediction of various stock indices. The structure used in these forecasting models is a simple linear combiner. The connecting weights of the adaptive linear combiner based models are optimized using ABFO and BFO by minimizing its mean square error (MSE). The short and long term prediction performance of these models are evaluated with test data and the results obtained are compared with those obtained from the genetic algorithm (GA) and particle swarm optimization (PSO) based models. It is in general observed that the new models are computationally more efficient, prediction wise more accurate and show faster convergence compared to other evolutionary computing models such as GA and PSO based models.

1. Introduction

Financial forecasting or specifically Stock Market prediction is one of the hottest fields of research due to its commercial applications and the attractive benefits it offers. As more and more money is being invested in the stock market, investors get nervous and anxious of the future trends of the stock prices in the markets. The primary area of concern is to determine the appropriate time to buy, hold or sell. Unfortunately, stock market prediction is not an easy task, because of the fact that stock market indices are essentially dynamic, nonlinear, complicated, nonparametric, and chaotic in nature (Tan, Quek, and Ng, 2005). The time series of these processes are multi-stationary, noisy, random, and has frequent structural brakes (Oh and Kim, 2002; Wang, 2003). In addition, stock market's movements are affected by many macro-economical factors (Wang, 2002) such as political events, firms' policies, general economic conditions, investors' expectations, institutional investors' choices, movement of other stock market and psychology of investors.

Many research works have been reported in the field of stock market prediction across the globe. Generally there are three schools of thoughts regarding such prediction. The first school believes that no investor can achieve above average trading advantages based on historical and present information. The major theories include the random walk hypothesis and the efficient market hypothesis (Peters, 1996). Taylor (1986) in his paper has provided compelling evidence to reject the random walk hypothesis and therefore researchers have been encouraged to suggest better models for market price prediction. The second view is that of fundamental analysis. Analysts have undertaken in-depth studies into the various macro-economic factors and have looked into the financial conditions and results of the industry concerned to discover the extent of correlation that might exist with the changes in the stock prices. Technical analysts have presented the third view on market price prediction. They believe that there are recurring patterns in the market behavior, which can be identified and predicted. In the process they have used number of statistical parameters called technical indicators and charting patterns from historical data. However, these techniques have often yielded contradictory results due to heavy dependence on human expertise and justification.

The recent trend is to develop adaptive models for forecasting financial data. These models can be broadly divided into statistical models and soft-computing models. One of the well known statistical methods is the one based on autoregressive integrated moving average (ARIMA) (Ayeni Babatunde and Pilat, 1992). The recent advancement in the field of soft-computing has given new dimension to the field of financial forecasting. Most artificial neural network (ANN) based models use historical stock index data such as technical indicators (Kim, 2006) to predict future prices. Tools based on ANN have increasingly gained popularity due to their inherent capabilities to approximate any nonlinear function to a high degree of accuracy. Neural networks are less sensitive to error

* Corresponding author. Tel.: +91 661 2462455; mobile: +91 9437048906; fax: +91 661 2462963.
E-mail address: ganapati.panda@gmail.com (G. Panda).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.01.012

term assumptions and can tolerate noise, chaotic components, and heavy tails better than most other methods (Masters, 1993). The three most popular ANN tools for the task are radial basis function (RBF) (Hann and Kamber, 2001), recurrent neural network (RNN) (Saad, Prokhorov, and Wunsch, 1998) and multilayer perceptron (MLP). More recently, new models based on multi-branch neural networks (MBNN) (Yamashita, Hirasawa, and Hu, 2005), local linear wavelet neural networks (LLWNN) (Chen, Dong, and Zhao, 2005) among others have been reported. The genetic algorithm (GA) has recently been applied (Tan et al., 2005; Kim, 2006) for prediction. Existing literature reveals that very little work has been reported on the use of evolutionary computing tools in training the weights of forecasting of models. Recently a new evolutionary computing technique known as bacterial foraging optimization (BFO) has been reported (Passino, 2002) and successfully applied to many real world problems like harmonic estimation (Mishra, 2005), transmission loss reduction (Tripathy, Mishra, Lai, and Zhang, 2006), active power filter for load compensation (Mishra and Bhende, 2007), power network (Tripathy and Mishra, 2007), load forecasting (Ulagammai, Venkatesh, Kannan, and Padhy, 2007) and independent component analysis (Acharya, Panda, Mishra, and Lakhshmi, 2007). The conventional BFO employs constant run length unit (the step by which the bacteria run or tumble in one go) in updating the location of the bacteria. To improve the optimization performance Takagi–Sugeno fuzzy scheme has been used to adapt the run length unit (Mishra, 2005). However, in Fuzzy-BFO the performance is linked with choice of the membership function and the fuzzy rule parameters and no systematic approach exists to determine these parameters for a given problem. Hence, the Fuzzy-BFO presented in Mishra (2005) is not suitable for optimizing various complex problems. In Mishra and Bhende (2007), a modified BFO proposed in has been used to optimize the coefficients of PI controller for active power filters. This algorithm has been shown to outperform a conventional GA with respect to convergence speed. Tripathy and Mishra (2007) have recently proposed an improved BFO algorithm for simultaneous optimization of the real power losses and voltage stability limit of a mesh power network. Simulation results of their approach shows superior performance compared to the conventional BFO based method. In a recent communication (Ulagammai et al., 2007) the BFO has been applied to train a wavelet neural network (WNN) meant for identifying nonlinear characteristics of power system loads. Acharya et al. (2007) have used the BFO in independent component analysis and have reported that the proppseed method yields better separation performance compared to the constrained genetic algorithm based ICA.

To the best of our knowledge none of the existing work has applied the BFO and adaptive BFO algorithms in designing forecasting models for short and long term prediction of stock indices. The present work is a humble contribution in this direction.

The present paper has two main objectives. Firstly it aims to develop a new forecasting model for prediction of stock indices using an adaptive linear combiner as the basic structure of the model and the BFO, a promising evolutionary computing tool, for training the parameters of the model. The second objective is to introduce a newly developed simple adaptive BFO (ABFO) technique and apply the same to develop more efficient prediction model for the same purpose. The prediction performance of the new models have been evaluated for short and long term prediction of stock indices and have been compared with those obtained from models based on other evolutionary computing tools such as GA and PSO. The proposed ABFO learning rule provides adaptive runlength in the chemotaxis step which leads to faster convergence during training compared to its BFO counterpart.

The organization of the paper proceeds as follows. Section 2 deals with the basic principle of the BFO and ABFO tools employed for training the linear combiner of the models. The BFO and ABFO based model developments for stock market prediction are outlined in Section 3. To demonstrate the prediction performance of the proposed models the simulation study is carried out in Section 4. This section also provides the formulae of computing the technical indicators. The results of simulation are discussed in Section 5. Finally the conclusion of the investigation is provided in Section 6.

2. Basics of BFO and adaptive BFO

Bacterial foraging optimization (BFO) is a new evolutionary computation technique which has been proposed by Passino (Tan et al., 2005). It is inspired by the pattern exhibited by bacterial foraging behaviour. Bacteria have the tendency to gather to the nutrient-rich areas by an activity called chemotaxis. It is known that bacteria swim by rotating whip like flagella driven by a reversible motor embedded in the cell wall. *E. coli* has 8–10 flagella placed randomly on a cell body. When all flagella rotate counterclockwise, they form a compact, helically propelling the cell along a trajectory, which is called run. When the flagella rotate clockwise, they pull on the bacterium in different directions and causes the bacteria to tumble. The bacterial foraging system primarily consists of four sequential mechanisms namely chemotaxis, swarming, reproduction and elimination-dispersal. A brief outline of each of these processes is given in this section.

(1) *Chemotaxis:* An *E. coli* bacterium can move in two different ways: it can run (swim for a period of time) or it can tumble, and alternate between these two modes of operation in the entire lifetime. In the BFO, a unit walk with random direction represents a tumble and a unit walk in the same direction indicates a run. In computational chemotaxis, the movement of the *i*th bacterium after one step is represented as

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(j) \quad (1)$$

where $\theta^i(j,k,l)$ denotes the location of *i*th bacterium at *j*th chemotactic, *k*th reproductive and *l*th elimination and dispersal step. $C(i)$ is the length of unit walk, which is a constant in basic BFO and $\phi(j)$ is the direction angle of the *j*th step. When its activity is run, $\phi(j)$ is same as $\phi(j-1)$, otherwise, $\phi(j)$ is a random angle directed within a range of $[0,2\pi]$.If the cost at $\theta^i(j+1,k,l)$ is better than the cost at $\theta^i(j,k,l)$ then the bacterium takes another step of size $C(i)$ in that direction otherwise it is allowed to tumble. This process is continued until the number of steps taken is greater than the number of chemotactic loop, N_c.

(2) *Swarming:* The bacteria in times of stresses release attractants to signal bacteria to swarm together. Each bacterium also releases a repellant to signal others to be at a minimum distance from it. Thus all of them will have a cell to cell attraction via attractant and cell to cell repulsion via repellant. The cell to cell signalling in *E. coli* swarm may be mathematically represented as

$$\begin{aligned} J_{cc}(\theta, P(j,k,l)) &= \sum_{i=1}^{S} J_{cc}(\theta, \theta^i(j,k,l)) \\ &= \sum_{i=1}^{S}\left[-d_a \exp\left(-w_a \sum_{m=1}^{p}(\theta_m - \theta_m^i)^2\right)\right] \\ &\quad + \sum_{i=1}^{S}\left[h_r \exp\left(-w_r \sum_{m=1}^{p}(\theta_m - \theta_m^i)^2\right)\right] \end{aligned} \quad (2)$$

where $J_{cc}(\theta,P(j,k,l))$ represents the objective function value to be added to the actual objective function, S is the total number of bacteria, p is the number of variables to be optimized and $\theta = [\theta_1, \theta_2, \ldots, \theta_p]^T$ is a point in the p-dimensional search domain. d_a, w_a, h_r and w_r are coefficients to be chosen properly.

(3) *Reproduction:* After all N_c chemotactic steps have been covered, a reproduction step takes place. The fitness values of the bacteria are sorted in ascending order. The lower half of the bacteria having higher fitness die and the remaining $S_r = S/2$ bacteria are allowed to spilt into two identical ones. Thus the population size after reproduction is maintained constant.

(4) Elimination and dispersal: Since bacteria may stuck around the initial or local optima positions, it is required to diversify the bacteria either gradually or suddenly so that the possibility of being trapped into local minima is eliminated. The dispersion operation takes place after a certain number of reproduction process. A bacterium is chosen, according to a preset probability p_{ed}, to be dispersed and moved to another position within the environment. These events may help to prevent the local minima trapping effectively, but unexpectedly disturb the optimization process. The detailed of this concept is presented in Tan et al. (2005). The flow chart of the complete algorithm is shown in Fig. 1.

2.1. Adaptive bacterial foraging optimization (ABFO)

Chemotaxis is a foraging strategy that implements a type of local optimization where the bacteria try to climb up the nutrient concentration, avoid noxious substance and search for ways out of neutral media. A chemotactic step size varying as the function of the current fitness value, is expected to provide better convergence behaviour as compared to a fixed step size. A simple adaptation scheme for the step size for ith bacterium given in (3) is employed to develop efficient model for prediction:

$$C(i) = \frac{\left|J^i(\theta)\right|}{\left|J^i(\theta)\right| + \alpha} = \frac{1}{1 + \frac{\alpha}{\left|J^i(\theta)\right|}} \tag{3}$$

where α is a positive constant

$J^i(\theta)$ = cost function of the ith bacterium
$C(i)$ = variable runlength unit of ith bacterium

If $J^i(\theta) \to 0$, then $C(i) \to 0$ and when $J^i(\theta) \to$ large, $C(i) \to 1$. This implies that the bacterium which is in the vicinity of noxious substance associates with higher cost function. Hence it takes larger steps to migrate to a place with higher nutrient concentration. Use of Eq. (3) in Eq. (1) is expected to offer improved convergence performance compared to fixed step size due to the logical explanation given above.

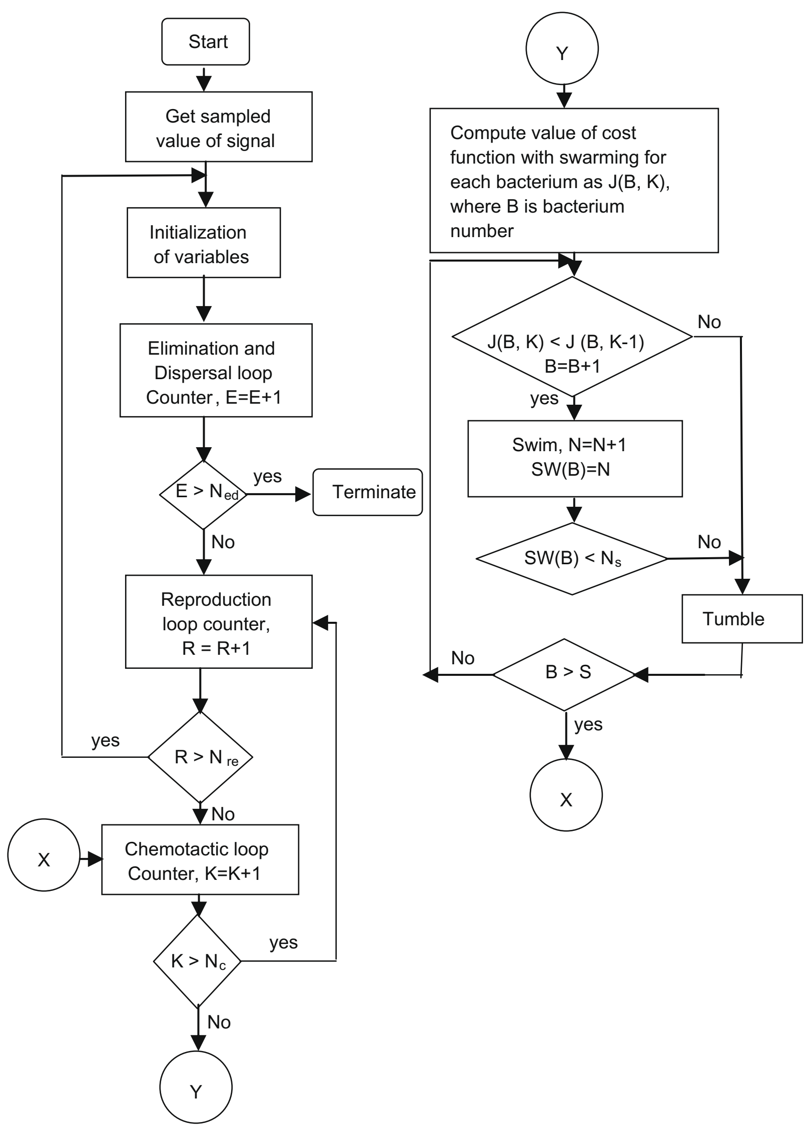

Fig. 1. The flowchart of BFO algorithm.

3. Application of BFO and ABFO to stock market prediction

The basic model for stock market prediction is assumed to be an adaptive linear combiner with parallel inputs as shown in Fig. 2. It is essentially an adaptive FIR filter having number of inputs equal to the number of features in the input pattern. The weights of the combiner are considered as the bacteria and initially their values are set to random numbers. A population of such bacteria is chosen to represent the initial solutions of the model. Each bacterium updates its values using the BFO principle so that the mean square error (MSE) of the model is minimized. The detail of optimization process is discussed in sequence.

3.1. Development of BFO based forecasting model

The development of the forecasting model using an adaptive linear combiner and BFO based training proceeds as follows

Step 1: Initialization of parameters
- (i) S = No. of bacteria to be used for searching the total region.
- (ii) N = Number of input patterns.
- (iii) p = Number of parameters to be optimized = no. of weights of the model.
- (iv) N_s = Swimming length after which tumbling of bacteria is undertaken in a chemotactic loop.
- (v) N_c = Number of iterations to be carried out in a chemotactic loop. Always $N_c > N_s$.
- (vi) N_{re} = Number of reproduction loops.
- (vii) N_{ed} = Maximum number of elimination and dispersal events to be imposed over the bacteria.
- (viii) p_{ed} = Probability with which the elimination and dispersal continues.
- (ix) θ^i = Location of ith ($i = 1,2,\ldots,S$) bacterium which is initially specified by random numbers between 0 and 1.
- (x) $C(i)$ = Step size of ith bacterium in the random direction.

Step 2: Iterative algorithm for optimizationThis section models the bacterial population, chemotaxis, reproduction, elimination and dispersal processes. The swarming operation of bacteria is not considered in this paper to keep the algorithm computationally simple and without much sacrificing the accuracy of prediction.Initially $j = k = l = 0$
- (i) Elimination-dispersal loop $l = l + 1$.
- (ii) Reproduction loop $k = k + 1$.
- (iii) Chemotaxis loop $j = j + 1$.

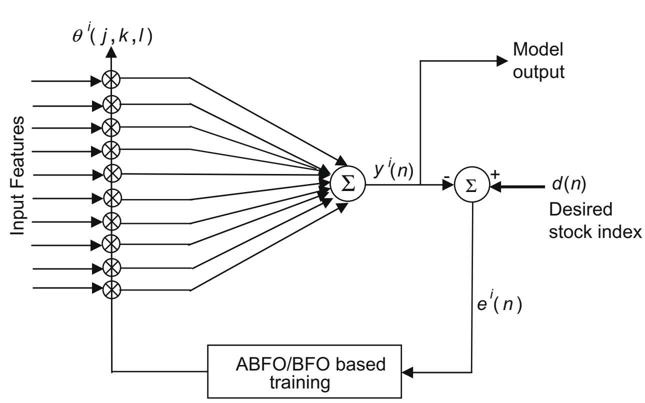

Fig. 2. Linear combiner-based forecasting model.

(a) For $i = 1,2,\ldots,S$
- (1) N training samples are passed through the model (10 different technical indicators computed from the stock market data are used at a time as input) and the weighted sum is computed to give the output.
- (2) The output is then compared with the corresponding desired signal to calculate the error.
- (3) The cost function, $J(i,j,k,l)$ for each ith bacterium which in the present case is the mean squared error is calculated using (4)

$$J(i,j,k,l) = \frac{1}{N}\sum_{n=1}^{N_i} e_i^2(n) = \frac{1}{N}\sum_{n=1}^{N_i}[d(n) - y^i(n)]^2 \tag{4}$$

The output of the linear combiner due to ith bacterium when nth pattern applied is given by

$$y^i(n) = \sum \underline{X}^T(n)\underline{\theta}^i \tag{5}$$

$d(n)$ = nth desired pattern (normalized closing stock index value)

$e^i(n)$ = error value for nth pattern due to ith bacterium

- (4) End of For loop.

(b) For $i = 1,2,\ldots,S$, the tumbling/swimming decision is taken.

Tumble: A random vector $\Delta(i)$ with each element $\Delta_m(i)$, $m = 1,2,\ldots,p$, a random number in the range $[-1,1]$ is generated.

Move: The location is updated as

$$\text{Let } \theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i) \times \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{6}$$

The second part of (6) is an adaptable step size in the direction of tumble for bacterium i.

The new cost function $J(i, j + 1, k, l)$ is computed corresponding to new location of bacteria..

Swim: (i) Let c = 0 (counter for swim length)

(ii) While $c < N_s$ (the bacteria have not climbed down too long)

Let $c = c + 1$

If $J^i(j + 1,k,l) < J_{last}$ (if doing better), let $J_{last} = J^i(j + 1,k,l)$. $\theta^i(j + 1,k,l)$ is again computed by (6) and then used to compute new $J^i(j + 1,k,l)$

ELSE let $c < N_s$. This is end of the WHILE statement.

(c) Go to next bacterium $(i + 1)$ if $i \neq S$ to process the next bacterium.

(d) If min(J) {minimum value of J among all the bacteria} is less than the tolerance limit then break all the loops.

Step 3: If $j < N_c$, go to (iii) i.e. continue chemotaxis loop since the life of the bacteria is not over.

Step 4: Reproduction.
- (a) For the given k and l, and for each ($i = 1,2,\ldots,S$) let J^i be the health of ith bacterium. Sort bacteria in ascending order of cost J (higher cost means lower health).
- (b) The $S_r = S/2$ bacteria with highest J value die and other S_r bacteria with the best value split and the copies that are made are placed at the same location as their parents.

Step 5: If $k < N_{re}$ go to 2.

Step 6: Elimination-dispersal.

For ($i = 1,2,\ldots,S$) with probability p_{ed}, eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To achieve this a bacterium is eliminated by simply dispersing one to a random location on the optimization domain.

Table 1
Selected technical indicators and their formula.

Technical indicators used	Formula
Exponential moving average (EMA) (3 numbers)	$(P \times A)$ + (Previous EMA $\times (1 - A)$); $A = 2/(N+1)$, P – current price, A – smoothing factor, N – time period (EMA10, EMA20 and EMA30 are calculated using the given formula)
Accumulation/distribution oscillator (ADO)	$\frac{(\text{CP}-\text{LP})-(\text{HP}-\text{CP})}{(\text{HP}-\text{LP})\times(\text{Period's volume})}$ CP – closing price, HP – highest price, LP – lowest price
Stochastic indicator (STI)	$\%\,K = \frac{(\text{Today's close}-\text{Lowest low in } K \text{ period})}{(\text{Highest high in } K \text{ period}-\text{Lowest low in } K \text{ period})} \times 100$ $\%\,D$ = SMA of $\%\,K$ for the period
Relative strength index (RSI) (2 numbers)	$\text{RSI} = 100 - \frac{100}{1+(U/D)}$ U – total gain/n, D – total losses/n, n – number of RSI period (RSI9 and RSI14 are calculated using the given formula)
Price rate of change (PROC)	$\frac{(\text{Today's close}-\text{Close } X\text{-period ago})}{(\text{Close}X\text{-period ago})} \times 100$ (PROC27 is calculated using the formula)
Closing price acceleration (CPACC)	$\frac{(\text{Close price}-\text{Close price } N\text{-period ago})}{(\text{Close price } N\text{-period ago})} \times 100$
High price acceleration (HPACC)	$\frac{(\text{High price}-\text{High price } N\text{-period ago})}{(\text{High price } N\text{-period ago})} \times 100$

ABFO algorithm for training of the forecasting model

Step 1: Same as that of the BFO based training.
Step 2: Same as that of the BFO. But the only difference is that while updating the location in Eq. (6) the adaptive runlength unit, $C(i)$, defined in Eq. (3) is used instead of fixed runlength unit.
*Steps 3–6:*Same as BFO based training.

4. Simulation study

4.1. Experimental data

The data for the stock market prediction experiments have been collected for Standard's and Poor's 500 (S&P 500), USA and Dow Jones Industrial Average (DJIA), USA. The experimental data used here consists of technical indicators and daily closing price of the indices. The total number of samples for the stock indices is 3228 trading days, from 3rd January 1994 to 23rd October 2006. Each sample consists of the closing price, opening price, lowest price, highest price and the total volume of stocks traded for the day. Ten technical indicators are selected as feature subsets by the review of domain experts and prior research. These are computed from the raw data as indicated in the Table 1.

The data are splitted into two sets – training and testing sets. The training set consists of 2510 patterns and the rest is set aside for testing. All the inputs are normalized to values between −1 and +1. The normalization is carried out using Eq. (7) by expressing the data in terms of the maximum and minimum value of the dataset.

$$y = \frac{(2 * x - (\text{max.} + \text{min.}))}{(\text{max.} + \text{min.})} \tag{7}$$

where y and x represent the normalized and the actual value, respectively.

4.2. Training and testing of the forecasting model

Training of the forecasting models are carried out using both BFO and ADBFO algorithms as described in Section 3 and the optimized weight values of the model are obtained through simulation. Then using the developed forecasting models the long term and short term prediction is carried and the performance of the models is assessed by comparing with known test data. To compare the performance of the proposed models, other two (GA and PSO) evolutionary computing based models are also simulated and the learning characteristics and test results of those are also presented.

The experiments carried out in this study to test the performance of the model for predicting the closing price of the index 1 day, 3, 5, 7, 15, 30 and 60 days in advance.

The mean absolute percentage error (MAPE) is used to gauge the performance of the prediction model when the test data are used. The MAPE is defined as

$$\text{MAPE} = \frac{1}{N}\sum_{j=1}^{N}\frac{|y(j) - \hat{y}(j)|}{y(j)} \times 100 \tag{8}$$

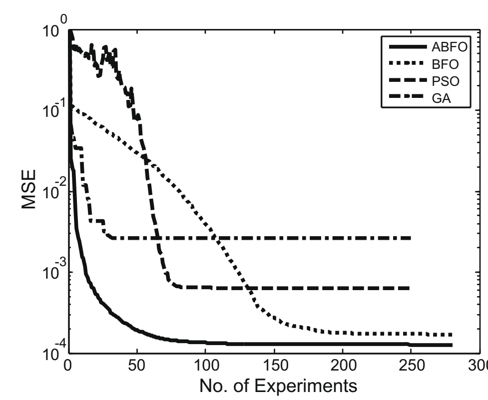

Fig. 3a. Comparison of learning characteristics of S&P 500 for 1 day advance.

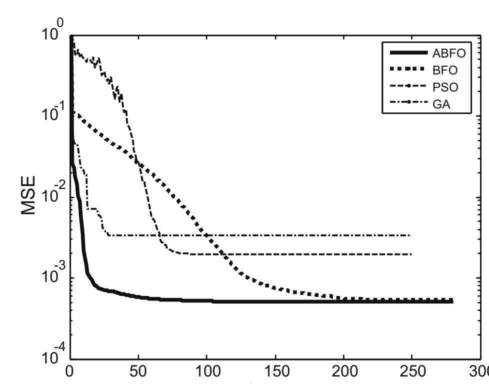

Fig. 3b. Comparison of learning characteristics of S&P 500 for 7 days advance.

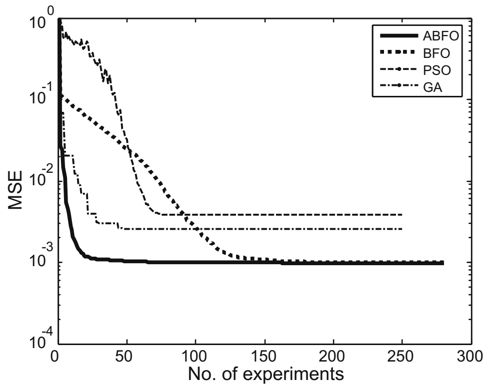

Fig. 3c. Comparison of learning characteristics of S&P 500 for 15 days advance.

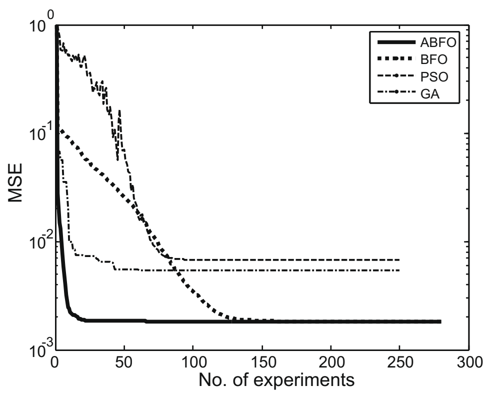

Fig. 3d. Comparison of learning characteristics of S&P 500 for 30 days advance.

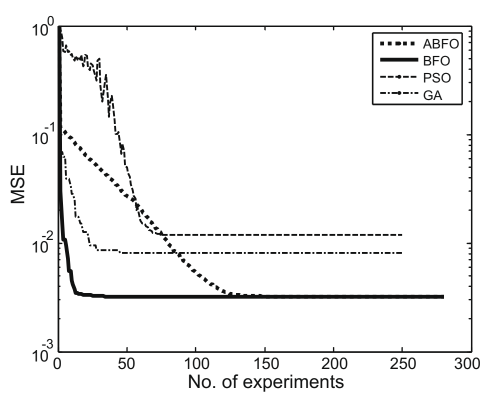

Fig. 3e. Comparison of learning characteristics of S&P 500 for 60 days advance.

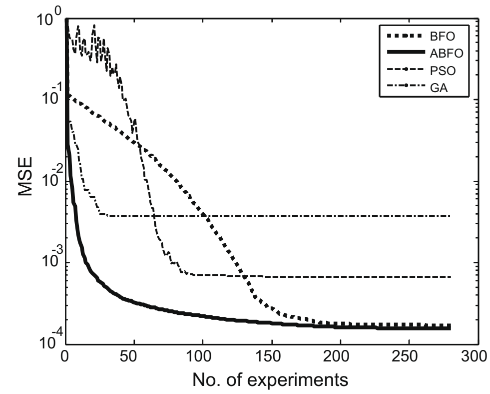

Fig. 4a. Comparison of learning characteristics of DJIA for 1 day advance.

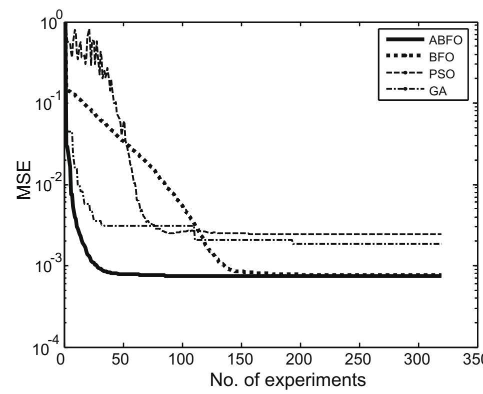

Fig. 4b. Comparison of learning characteristics of DJIA for 7 days advance.

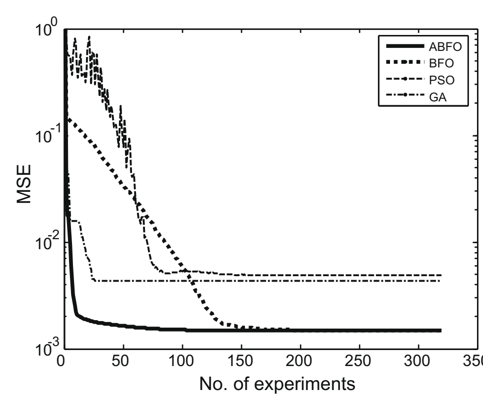

Fig. 4c. Comparison of learning characteristics of DJIA for 15 days advance.

where

$y(j)$ actual closing stock index on jth day
$\hat{y}(j)$ predicted closing stock index on jth day
N number of test data

The 10 technical indicators used for this simulation are defined in Table 1. Various parameters employed in the simulation study for BFO and ABFO are:

$S_b = 16$, $N_{is} = 2510$, $p = 10$, $N_s = 3$, $N_c = 5$, $N_{re} = 140$–160, $N_{ed} = 2$, $P_{ed} = 0.25$, $C(i) = 0.0075$ and $\alpha = 0.2$. To compare the performance of the proposed model the PSO and GA based models are also simulated using same technical indicators.

5. Results and discussion

The convergence characteristics of ABFO, BFO, GA and PSO models for 1 day, 7, 30 and 60 days ahead prediction of S&P 500 and DJIA stock indices are shown in Figs. 3a–3e, 4b–4e, respectively. These plots clearly indicate that the ABFO based model offers fastest convergence during training followed by GA, PSO and then BFO based models. Further, according to residual MSE achieved the models are graded as ABFO/BFO, GA and PSO. In both counts the proposed ABFO model provides best performance compared to other three models. Since residual MSEs of BFO and ABFO models are same, they provide identical accuracy in prediction. However

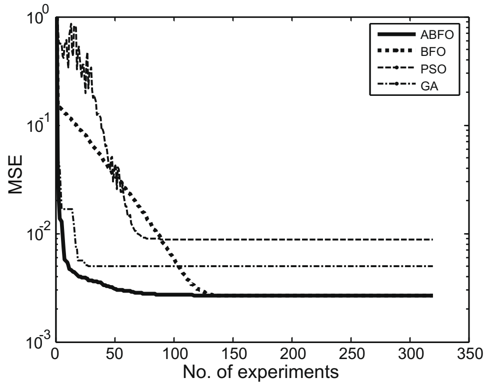

Fig. 4d. Comparison of learning characteristics of DJIA for 30 days advance.

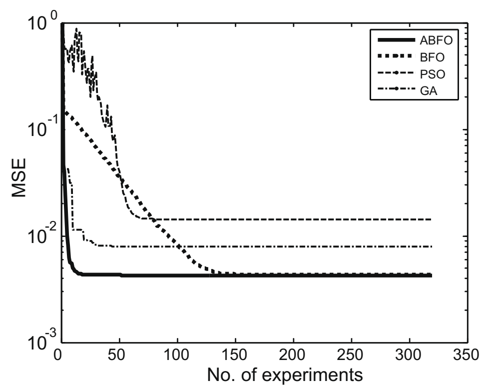

Fig. 4e. Comparison of learning characteristics of DJIA for 60 days advance.

Table 2
Comparison of MAPE for S&P 500 stock index obtained from different models.

Days ahead	MAPE			
	ABFO	BFO	PSO	GA
1	0.6747	0.8108	0.8070	1.8465
3	0.9751	1.0105	1.0770	1.9800
5	1.2082	1.2399	1.2546	1.3441
7	1.3865	1.3912	1.3984	1.5278
15	1.8365	1.8447	1.8530	1.8690
30	2.4086	2.5310	2.3854	2.6171
60	3.1591	3.2489	3.2719	3.2724

Table 3
Comparison of MAPE for DJIA stock index obtained from different models.

Days ahead	MAPE			
	ABFO	BFO	PSO	GA
1	0.6098	0.8110	0.8634	1.5985
3	0.9436	0.9614	0.9846	1.4489
5	1.1847	1.1888	1.1939	1.7093
7	1.3884	1.3831	1.3953	1.5547
15	1.8879	1.8662	1.8707	2.2634
30	2.3048	2.3442	2.3843	2.6931
60	2.7825	2.7832	2.8084	3.2684

GA and PSO based models offer inferior prediction compared to other two models. The above finding is true for both stock indices and for all ranges of prediction studied. The MAPE defined in Eq. (8) denotes the accuracy of prediction. The MAPE values for S&P 500

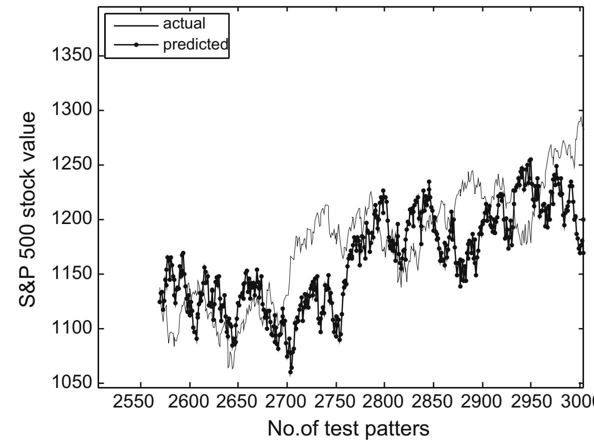

Fig. 5. Comparison of actual and 60 days ahead predicted value during testing for S&P 500 using ABFO.

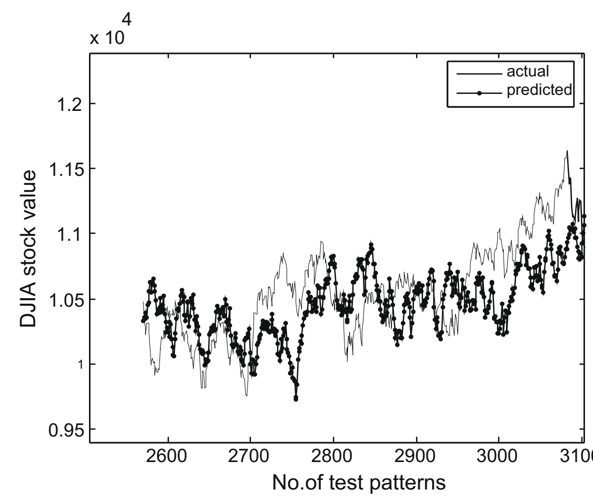

Fig. 6. Comparison of actual and 60 days ahead predicted value during testing for DJIA using ABFO.

for different range of prediction computed using four different models are listed in Table 2. The same for the DJIA stock index is displayed in Table 3. Comparison of the result in both cases, as expected, reveals that the ABFO model makes best prediction among the four models and then comes the BFO one. Figs. 5 and 6 show the actual vrs predicted S&P 500 and DJIS indices for 60 days ahead using ABFO model when test data are used as input. Comparison reveals very good agreement between the two for both indices. It is in general observed that the proposed models predict S&P 500 and DJIA stock indices with less than 1% error for one and three days ahead, less than 2% error up to 15 days ahead and about 3% error for 60days ahead prediction for both the indices.

6. Conclusion

In this paper two new forecasting models based on BFO and ABFO are developed to predict different stock indices using technical indicators derived from the past stock indices. The structure of these models are basically an adaptive liner combiner, the weights of which are trained using the ABFO and BFO algorithms. To demonstrate the performance of the proposed models simulation study is carried out using known stock indices and their prediction performance is compared with standard GA and PSO based forecasting models. The comparison indicates that the proposed models offer lesser complexity, better prediction accuracy and faster training compared to those obtained from the GA and PSO based models. Out of the two new models proposed the ABFO model provides best performance in all counts followed by the simple PSO based model.

References

Acharya, D. P., Panda, G., Mishra, S., & Lakhshmi, Y. V. S. (2007). Bacteria foraging based independent component analysis. In *Proceedings of the international conference on computational intelligence and multimedia applications, 13–15 December* (Vol. 2, pp. 527–531).

Ayeni Babatunde, J., & Pilat, Richard (1992). Crude oil reserve estimation: An application of the autoregressive integrated moving average (ARIMA) model. *Journal of Petroleum Science and Engineering, 8*(1), 13–28.

Chen, Y., Dong, X., & Zhao, Y. (2005). Stock index modeling using EDA based local linear wavelet neural network. In *Proceedings of the international conference on neural networks and brain, 13–15 October* (Vol. 3, pp. 1646–1650).

Hann, J., & Kamber, N. (2001). *Data mining: Concepts and techniques*. San Francisco: Morgan Kanfmann Publishers.

Kim, Kyoung-Jae (2006). Artificial neural networks with evolutionary instance selection for financial forecasting. *Expert System with Applications, 30*, 519–526.

Masters, T. (1993). *Practical neural network recipes in C++*. New York: Academic Press.

Mishra, S. (2005). A hybrid least square fuzzy bacterial foraging strategy for harmonic estimation. *IEEE Transactions on Evolutionary Computation, 9*(1), 61–73.

Mishra, S., & Bhende, C. N. (2007). Bacterial Foraging technique based optimized active power filter for load compensation. *IEEE Transactions on Power Delivery, 22*(1), 457–465.

Oh, K. J., & Kim, K.-j. (2002). Analyzing stock market tick data using piecewise non linear model. *Expert System with Applications, 22*(3), 249–255.

Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control System Magazine*, 52–66.

Peters, E. (1996). *Chaos and order in capital markets: A new view of cycles prices market volatility*. John Wiley and Sons Inc.

Saad, E. W., Prokhorov, D. V., & Wunsch, D. C. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions of Neural Network, 9*(6), 1456–1470.

Tan, T. Z., Quek, C., & Ng, G. S. (2005). Brain inspired genetic complimentary learning for stock market prediction. In *Proceedings of the IEEE congress on evolutionary computation, 2–5 September* (Vol. 3, pp. 2653–2660).

Taylor, S. (1986). *Modeling financial time series*. John Wiley and Sons.

Tripathy, M., & Mishra, S. (2007). Bacteria foraging based to optimize both real power loss and voltage stability limit. *IEEE Transactions on Power Systems, 22*(1), 240–248.

Tripathy, M., Mishra, S., Lai, L. L., & Zhang, Q. P. (2006). Transmission loss reduction based on FACTS and Bacteria Foraging algorithm. *PPSN*, 222–231.

Ulagammai, L., Venkatesh, P., Kannan, S. P., & Padhy, N. P. (2007). Application of bacteria foraging technique trained and artificial and wavelet neural networks in load forecasting. *Neurocomputing*, 2659–2667.

Wang, Y. (2002). Predicting stock price using fuzzy grey prediction system. *Expert System with Applications, 22*, 33–39.

Wang, Y. (2003). Mining stock prices using fuzzy rough set system. *Expert System with Applications, 24*(1), 13–23.

Yamashita, T., Hirasawa, K., & Hu, J. (2005). Application of multi-branch neural networks to stock market prediction. In *Proceedings of the international joint conference on neural networks, Montreal, Canada, 31 July–4 August* (Vol. 4, pp. 2544–2548).