Time Series Forecasting Using a Hybrid Adaptive Particle Swarm Optimization and Neural Network Model

Yi XIAO

School of Information Management, Central China Normal University, Wuhan 430079, China
E-mail: yxiao@mail.ccnu.edu.cn

John J. LIU

Center for Transport Trade and Financial Studies, City University of Hong Kong, Hong Kong, China
E-mail: johnliu@cityu.edu.hk

Yi HU

School of Management, University of Chinese Academy of Sciences, Beijing 100190, China
E-mail: huji@amss.ac.cn

Yingfeng WANG

Center for Transport Trade and Financial Studies, City University of Hong Kong, Hong Kong, China
E-mail: gilbertw@cityu.edu.hk

Abstract For time series forecasting, the problem that we often encounter is how to increase the prediction accuracy as much as possible with the irregular and noise data. This study proposes a novel multilayer feedforward neural network based on the improved particle swarm optimization with adaptive genetic operator (IPSO-MLFN). In the proposed IPSO, inertia weight is dynamically adjusted according to the feedback from particles' best memories, and acceleration coefficients are controlled by a declining arccosine and an increasing arccosine function. Further, a crossover rate which only depends on generation and does not associate with the individual fitness is designed. Finally, the parameters of MLFN are optimized by IPSO. The empirical results on the container throughput forecast of Shenzhen Port show that forecasts with IPSO-MLFN model are more conservative and credible.

Keywords time series forecasting; particle swarm optimization; genetic operator; multilayer feedforward neural network

1 Introduction

With the rapid growth of economic globalization, economic forecast has always been an attention-holding issue. In general, there are two methods for economic forecasting: qualitative and quantitative. Qualitative methods, e.g., Delphi method and expert meetings, forecast the future development of the object mainly depending on the experts' experience, knowledge and analytical skills. Quantitative methods usually establish mathematical forecasting models

Received November 19, 2013, accepted February 25, 2014

Supported by the National Social Science Foundation of China (Grant No. 14BGL175) and the Selfdetermined Research Funds of CCNU from the Colleges' Basic Research and Operation of MOE (Grant No. CCNU13F030)

based on historical statistical data. Since the latter are more objective and precise, they have gotten more and more attentions. According to the difference of the throughput quantitative forecast methods, they can be divided into three categories: time series, causal analysis and combination forecasting. Time series methods, establishing a mathematical model only by historical data, include autoregressive integrated moving average model (ARIMA), exponential smoothing, gray system method, seasonal adjustment method, etc.^[1]. Causal analysis methods examine the correlation among a series of economic indicators, and build a forecasting model according to the relevant economic indicators. At present, such methods include regression analysis, the elasticity coefficient method, system dynamics, etc.^[2]. Combination forecasting methods get the final forecast result by integrating the results of some individual models, such as TEI@I methodology integrating qualitative and quantitative analysis proposed by Wang et al.^[3].

In monitoring the changes in seasonal patterns and business cycles, short-term forecasts often yield better results than long-term forecasts^[4]. However, it is not easy to forecast short-term economy due to their typical irregularity and noise. The difficulty in economic forecasting is usually attributed to the limitation of many traditional forecasting models, which has encouraged academic researchers and business practitioners to develop more effective forecasting models. In this case, the artificial intelligence models such as artificial neural network (ANN) have been recognized as more useful than traditional statistical forecasting models^[5–15]. For example, Lam et al.^[16] proposed and developed the neural network models for forecasting 37 types of freight movements of Hong Kong port cargo throughput, and it is shown that the forecasting results are more accurate compared with that of regression analysis.

As we all know, neural networks are a kind of unstable learning methods^[17–19]. Even for some simple problems, different structures of neural networks (e.g., different number of hidden layers, different hidden nodes and different initial conditions) result in different patterns of network generalization^[20, 21]. For the neural network design, the key is how to determine the various parameters which can solve an actual problem based on a performance evaluation criteria. However, it is difficult to design a neural network when the problem is particularly complex because there is few rigorous design criteria. Thus, it requires efficient automatic design methods for ANN with the development of macro-scale and complexity.

In different ANN models, the multilayer feedforward neural network (MLFN) is the most widely used one. Evolutionary computation algorithms are demonstrated to be suitable for the optimization in MLFN^[22]. As a popular evolutionary computation paradigm, the particle swarm optimization (PSO) utilizes a “population” of candidate solutions to evolve toward an optimal or near optimal solution of an actual problem. Because of its features of simplicity, easy implementation, and quick convergence, PSO has attracted more and more researchers and been applied extensively to various fields^[23].

Despite of its success and popularity, Grimaldi et al. have indicated that, although PSO may find solutions of reasonable quality much faster than other evolutionary computation algorithms, it can not improve the quality of the solutions as the number of iterations increases. Hence, a premature phenomenon may occur for the standard PSO, especially in optimizing complex multi-objective functions^[24]. Therefore, many improved PSO algorithms have been

proposed. For example, Alfi et al.^[25] have presented a methodology for finding optimal system parameters and optimal control parameters by a novel adaptive particle swarm optimization (APSO) algorithm. Wang et al.^[26] have proposed a poly-hybrid PSO optimization method with intelligent parameter adjustment.

Because the future irregular events as causal factors are not reflected in models based on the historical data, contextual knowledge usually must be introduced to correct the forecasts from the models.

In light of the previous studies, this paper develops a multilayer feedforward neural network (MLFN) based on improved swarm particle optimization algorithm (IPSO-MLFN). The main creative contributions of this algorithm are as follows: (a) All parameters in the MLFN model are adaptively adjusted by the improved particle swarm optimization (IPSO) algorithm. To ameliorate the performance of standard PSO, IPSO employs adaptive nonlinear inertia weight updating with fitness values, which will help to balance the exploring and exploiting capabilities at different stages during its search process. At the same time, acceleration parameters are controlled by a declining arccosine function and an increasing arccosine function. The two strategies attempt to promote particles to be placed in an unexplored area so that they can contribute to the process of finding better solutions in the early-stage of IPSO. Further, this method is more conducive to the algorithm to get rid of the interference of local minimum, avoid premature convergence and improve the convergence speed and accuracy in the latter stage of IPSO. In a word, the improved design in IPSO enhances the prophase global search capability and late continuous optimizing ability. (b) In PSO, the crossover operation is introduced to improve the performance of the candidate particles. To achieve this goal, it is a key to determine the probability curve of crossover operation. In this study, we adopt two-point crossover and design a crossover rate only depending on generation. Finally, the optimal structure and parameters of MLFN are adjusted by IPSO in the training process. With IPSO the deadly drawbacks of the MLFN, e.g., difficult to select the parameters and frequent confinement to local minima, have been significantly improved. Empirical results in this study show that the IPSO-MLFN model achieves the better forecasting accuracy than MLFN model and PSO-MLFN model.

The remainder of this study is organized as follows: it describes the process of the multilayer feedforward neural network forecasting model based on improved swarm particle optimization algorithm (IPSO-MLFN) in detail in Section 2; proceeds the empirical study on container throughput forecasting of Shenzhen Port with monthly time series in Sections 3; finally, some concluding remarks are drawn in Section 4.

2 The improved GA-PSO based multilayer feedforward neural network

2.1 Artificial neural network

Artificial neural networks (ANNs) are a set of systems derived through neuropsychology models. The basic idea of ANNs is to emulate the biological system of the human brain to learn and identify patterns. Different ANNs have been proposed, in which multilayer feedforward neural network (MLFN) is the most widely used one. The MLFN is widely used for time series forecasting because its nonlinear modeling capability can capture the nonlinear characteristics

of time series well. When applying MLFN to time series forecasting, the final output can be represented as

$$y_t = \varphi(y_{t-1}, y_{t-2}, \dots, y_{t-p}, v) + \xi_t \quad (1)$$

where v is the parameter vector and φ is a function determined by the network structure and connection-weights. Thus, in some senses, the MLFN model is equivalent to a nonlinear autoregressive model.

A major advantage of MLFN is its ability to provide flexible mapping between inputs and outputs. Furthermore, Hornik et al.^[27] have theoretically proved that a three-layer feedforward neural network (MLFN) can approximate any continuous function to any desire accuracy. Therefore, a three-layer MLFN model is used as a basic learning paradigm in this study.

2.2 Particle swarm optimization algorithm

Particle swarm optimization (PSO) is a population-based stochastic optimization algorithm which has been proposed by Eberhart and Kennedy in 1995. The concept is mainly from the natural flocking and swarming behavior of birds and insects. It is considered to be able to optimize the performance of MLFN by improving its disadvantages such as difficult to select the parameters and easy to get stuck in a local minimum, because it does not require gradient and differentiable information.

Suppose that the search space is h dimensional, the particles of the swarm can be represented by an n dimensional vector $X_i = (x_{i1}, x_{i2}, \dots, x_{ih})^T$. The fitness of each particle can be evaluated according to the objective function of the actual optimization problem. The velocity of each particle can be represented by n dimensional vector $V_i = (v_{i1}, v_{i2}, \dots, v_{ih})^T$. Let $P_b = (p_{b1}, p_{b2}, \dots, p_{bh})^T$ be the last best position of the i -th particle, which is noted as its individual best position. Further, $G_b = (g_{b1}, g_{b2}, \dots, g_{bh})^T$ is the global best position. The new velocity of particle will be assigned according to the following equations:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_1[p_{bj} - x_{ij}(t)] + c_2r_2[g_{bj} - x_{ij}(t)] \quad (2)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), j = 1, 2, \dots, h \quad (3)$$

where c_1 and c_2 represent the acceleration parameters, w represents the inertia weight, and r_1 and r_2 are random numbers ranging from 0 to 1. The velocities of the particles on each dimension are clamped to a maximum velocity: $v_{\max}$. The new position of each particle is calculated by Eq. (3).

2.3 The improved particle swarm optimization algorithm

Although the traditional PSO can usually find good solutions rapidly, it may be trapped in local minimum and fail to converge to the best position. So, in recent years, some researches have been done to deal with this problem. In order to reduce the opportunity of trapping in a local optimum, expand the search scope of the algorithm and enhance the algorithm's climbing ability, it is certainly critical to always maintain the diversity of particles. The existing algorithms such as chaos mechanism optimization, hybrid simplex search PSO, comprehensive learning PSO, dynamic random search technique are difficult to solve the two problems (global optimization and premature convergence) simultaneously. Therefore, we design an improved particle swarm optimization (IPSO) with adaptive nonlinear inertia weight and dynamic arccosine function

acceleration parameters. At the same time, the crossover operation of GA is introduced in IPSO in order to improve the performance of the candidate particles.

1) Improved acceleration coefficients: In the particle swarm optimization algorithm, the acceleration coefficients c_1 and c_2 control the “cognitive” part and “social” part of the particle velocity respectively. In general, population-based optimization methods always hope the individuals to search entire solution space in the initial stages of optimization, which can increase the diversity of the particles and avoid trapping into a local value prematurely. At the end of the optimization, it is very important to find the global optimum effectively for improving the algorithm convergence speed and accuracy^[28]. Thus, a large “cognitive” c_1 and a small “social” c_2 are required in the initial stages, and a small “cognitive” c_1 and a large “social” c_2 are required in the end stages of optimization. Based on this idea, many methods have been proposed such as linear adjustment strategy, fuzzy control strategy, and random change strategy. However, these methods are unstable. Therefore, we propose a dynamic acceleration parameters adjustment strategy based on arccosine function. c_1 is controlled by a declining arccosine function and c_2 is controlled by an increasing arccosine function. This strategy attempts to promote particles to be placed in an unexplored area so that they can contribute to the process of finding better solutions in the early stages of optimization. The method is more conducive to getting rid of the interference of local minimum, obtaining the global optimal solution to avoid premature convergence, and improving the convergence speed and accuracy in the latter stages of optimization. The improved design in PSO enhances the prophase global search capability and anaphase continuous optimizing ability. The strategy can be represented as

$$c_1 = c_{1_{start}} + (c_{1_{end}} - c_{1_{start}}) \times \left[1 - \arccos \left(\frac{-2Iter}{Iter_{max}} + 1 \right) / \pi \right] \quad (4)$$

$$c_2 = c_{1_{start}} - c_1 \quad (5)$$

where c_{start} represents the iteration initial value of acceleration parameters, c_{end} represents the iteration final value of acceleration parameters, $Iter$ is the current iteration number, $Iter_{max}$ is the maximum iteration number.

2) Improved inertia weight: The inertia weight w represents the contribution of past velocity values to the current velocity of the particle. A large inertia weight biases the search towards global exploration, while a smaller inertia weight directs towards fine-tuning the current solutions. Suitable selection of the inertia weight and acceleration coefficients can provide a balance between the global and the local search. Based on this idea, many methods have been proposed such as linear decreasing inertia weight strategy, random inertia weight strategy, inertia weight strategy based on concave function and convex function, and fuzzy control strategy. However, these methods are not adaptive. In this study, we employ an adaptive nonlinear adjustment inertia weight strategy depending on particle's fitness value, which will help balance the exploring and exploiting capabilities at different stages during its search process. The strategy can be represented as

$$w = w_{min} + \frac{(w_{max} - w_{min})(fitness - fitness_{min})}{fitness_{avg} - fitness_{min}}, \quad fitness \leq fitness_{avg} \quad (6)$$

$$w = w_{max}, \quad fitness > fitness_{avg} \quad (7)$$

where $w_{\min}$, $w_{\max}$ represent the range of inertia weight, fitness represents the current fitness value of some particle, fitnessmin and fitnessavg represent the minimum fitness value and the average fitness value of all particles respectively. It can be seen from Eq. (6~7) that, the inertia weight will increase when the fitness values of particles are consistent (become local optimum) and will decrease when the fitness values of all particles are scattered. Therefore, the inertia weights of the superior particles whose fitness values are larger than the average fitness value are smaller to protect their properties. In contrast, the inertia weights of the poor particles whose fitness values are smaller than the average fitness value are larger so that they can search better space.

3) Genetic operators: In PSO algorithm, when the individual optimum solution P_{best} has not been updated for a long time in the latter part of the training, the particles will be close to the global optimum solution G_{best} . At this point the particle update velocity mainly depends on wv_{ij} of the first part of Eq. (2) because the inertia weight $w < 1$, the particle velocity will become increasingly smaller. The particle swarm will “fly” toward a direction, which will lead to its falling into local minimum position. In this study, the adaptive genetic operators is introduced in order to improve the performance of the candidate particles. The particles can execute crossover operation according to a certain probability.

Crossover is the main search operator in GAs, creating offsprings by randomly mixing sections of the parental genome. The number of sections exchanged varies widely with the GA implementation. The most common crossover algorithms include one-point crossover, two-point, k -point crossover and uniform crossover^[4]. In this study, we use two-point crossover and design a crossover rate only depending on iteration number and not associating with the individual fitness. The crossover rate can be represented as

$$p_{ct} = p_{c,\max} \times 8^{(-t/T)} \quad (8)$$

$$p_c(t) = p_{ct}, p_{ct} > p_{c,\min} \quad (9)$$

$$p_c(t) = p_{c,\min}, p_{ct} \leq p_{c,\min} \quad (10)$$

where p_{ct} is calculated variable, T is maximum iteration number, t is current iteration number, $p_{c,\min}$ is minimum crossover probability, $p_{c,\max}$ is maximum crossover probability, $p_c(t)$ is crossover probability of t -th iteration.

In order to improve search efficiency, take advantages of PSO's training speed and GA's global search, the genetic operator control function in this study is defined as

$$GP_k = 1 - \frac{1}{1 + \ln k}, \quad k = 1, 2, \dots \quad (11)$$

where k represents current iteration number. In the process of each iteration, a random number will be created ranging from 0 to 1. If the number is less than GP_k , the current particle will execute genetic operator. As can be seen from Eq. (11), in the early iterations $GP_k \ll 1$, genetic operator will be executed at a small probability, in the later iterations GP_k , it will be close to be 1, so the particle will execute genetic operator at greater probability. Genetic operator expands population search space shrinking in the process of iteration, so that particles can escape from the optimal value searched previously to a larger search space. The particles

maintain the diversity of the population, thus it increases the possibility of finding better solutions.

2.4 MLFN optimized by the improved PSO

The deadly drawbacks of the MLFN (frequent confinement to local minima and parameters selection) are expected to be improved with IPSO. The basic idea is to optimize weights and bias of MLFN by particle swarm optimization. A particle in PSO real-coded represents a set of MLFN weight vector and bias weight vector.

Let the number of input layer nodes be R , the number of hidden layer nodes Q , the output layer nodes S , the weight and bias vector of MLFN can be represented as

$$X = [w_{11}^1, \dots, w_{1Q}^1, w_{21}^1, \dots, w_{2Q}^1, \dots, w_{R1}^1, \dots, w_{RQ}^1, b_1^1, \dots, b_Q^1, w_{11}^2, \dots, w_{1S}^2, w_{21}^2, \dots, w_{2S}^2, \dots, w_{Q1}^2, \dots, w_{QS}^2, b_1^2, \dots, b_S^2] \quad (12)$$

$$h = R(Q + 1) + Q(S + 1) \quad (13)$$

where $w_{ij}^1 (i = 1, 2, \dots, R; j = 1, 2, \dots, Q)$ represents the weight vector from the input layer to the hidden layer, $w_{ij}^2 (i = 1, 2, \dots, Q; j = 1, 2, \dots, S)$ represents the weight vector from the hidden layer to the output layer, $b_i^1 (i = 1, 2, \dots, Q)$ represents the hidden layer bias vector, $b_i^2 (i = 1, 2, \dots, S)$ represents the output layer bias vector, and h is the dimension of vector X .

3 Empirical study

3.1 Data

The monthly container throughput time series of Shenzhen Port from the ICEC database depicted in Fig. 1 have been used in our experiments in the period from January 2001 to June 2013 (150 observations). The monthly data from January 2001 to June 2011 (126 observations) are used as the training set and the remaining data from July 2011 to June 2013 (24 observations) as the test set.

Figure 1 Monthly container throughput of Shenzhen port

3.2 Set parameters

The initial parameters of IPSO-MLFN model are defined as: the population size of particle swarm is 45, the maximum iteration number is 180, the training times of MLFN are 500; in IPSO the initial particle positions are random numbers ranging from -12 to 12 and the initial particle velocity randomly varies between -12 and 12 , $c_{1_{start}} = 3.65$, $c_{1_{end}} = 1.05$, $w_{min} = 0.25$, $w_{max} = 0.75$; in genetic algorithm $P_{c,min} = 0.46$, $P_{c,max} = 0.86$.

3.3 Forecasting

The monthly container throughput data of Shenzhen port are standardized to the range from 0 to 1 and the inputs to IPSO-MLFN are irregular data. Fig. 2 illustrates the forecast results in the training set and test set. Meanwhile, the performance of IPSO-MLFN is also compared with some commonly used methods including ARIMA, MLFN and PSO-MLFN. In order to evaluate the performance of IPSO-MLFN forecasting model, three performance measures, MAE, MAPE and RMSE, are used. Table 1 shows the in-sample fit and the out-of-sample forecasting results with the average performance of the ARIMA, MLFN, PSO-MLFN and IPSO-MLFN when retrained 10 times.

Figure 2 Comparison of forecast results of different forecasting models

Table 1 Comparison of average performance of different models over 10 runs

Model	In-sample			Out-of-sample		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
MLFN	20.9856	0.12196	27.1232	21.6671	0.12591	28.0037
ARIMA	19.4472	0.10328	25.9267	20.0696	0.1067	26.7638
PSO-MLFN	19.5941	0.10547	25.3687	19.6169	0.11146	26.1876
IPSO-MLFN	15.6317	0.08749	23.4863	16.1424	0.0904	24.2449

It can be seen from Table 1 that: (a) according to all indices, MLFN is the worst model, the forecasting performance of PSO-MLFN is better than that of ARIMA and MLFN, which indicates the PSO algorithm can improve the performance of the traditional MLFN; (b) the

prediction performance of IPSO-MLFN is better than that of PSO-MLFN, which confirms the improvements to the PSO algorithm can effectively reduce errors and provide better performance than the standard PSO; (c) the in-sample fit performance is superior to the out-of-sample forecasting performance.

As can be seen in Figure 2, in the period from 2009 to 2010, because of the global financial crisis origination from United States, the container throughput growth of Shenzhen port slowed down. But with the rapid recovery of Chinese economy from the financial crisis, from 2010 the container throughput of Shenzhen Port began to grow quickly. The financial crisis as a causal factor is not reflected in models based on the historical data, therefore, the forecasting error of three models in this year is relatively large. However, the prediction performance of IPSO-MLFN is better than the other two models.

4 Conclusions

In this study we hope to design a model that can effectively improve the performance of container throughput forecasting. To overcome the drawbacks of the traditional MLFN and PSO, we design an improved particle swarm optimization (IPSO) with adaptive nonlinear inertia weight and dynamic arccosine function acceleration parameters. At the same time, the crossover operation and mutation operation of GA are introduced to IPSO in order to improve the performance ability of the candidate particle. The proposed model incorporates the difference between particles into IPSO, so that it can simulate a more precise biological model rather than a roughly animal behavior and truly reflect the actual search process in terms of feedback taken from particles' best memories. By applying monthly data to these models and comparing the forecast results based on mean absolute error, mean absolute percent error and root mean squared error, we find that the MLFN based on improved PSO (IPSO-MLFN) offers generally superior forecasting performance than PSO-MLFN and standard MLFN. This experiment suggests that it is possible to extract information hidden in the container throughput because of its ability of data processing and knowledge discovering. The future work is to apply IPSO-MLFN model to dynamic financial market with high volatility and irregularity such as exchange rates, stock, etc.

Acknowledgements We thank the referees for their time and comments. This paper was completed during the first author's visit at Center for Transport Trade and Financial Studies, City University of Hong Kong. He is grateful to the center and the university for financial support for his visit.

References

- [1] Chen S H, Chen J N. Forecasting container throughputs at ports using genetic programming. *Expert Systems with Applications*, 2010, 37(3): 2054–2058.
- [2] Peng W Y, Chu C W. A comparison of univariate methods for forecasting container throughput volumes. *Mathematical and Computer Modelling*, 2009, 50(7–8): 1045–1057.
- [3] Wang S Y, Yu L, Lai K K. Crude oil price forecasting with TEI@I methodology. *Journal of Systems Science and Complexity*, 2005, 18(2): 145–166.
- [4] Franses P H, Van Dijk D. The forecasting performance of various models for seasonality and nonlinearity for quarterly industrial production. *International Journal of Forecasting*, 2005, 21(1): 87–102.

- [5] Xiao Y, Xiao J, Lu F B, et al. Ensemble ANNs-PSO-GA approach for day-ahead stock e-exchange prices forecasting. *International Journal of Computational Intelligence Systems*, 2013, 6(1): 96–114.
- [6] Yu L, Wang S Y, Lai K K. A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates. *Computers & Operations Research*, 2005, 32(10): 2523–2541.
- [7] Xiao Y, Xiao J, Wang S Y. A hybrid model for time series forecasting. *Human Systems Management*, 2012, 31(2): 133–143.
- [8] Yu L, Lai K K, Wang S Y. Currency crisis forecasting with general regression neural networks. *International Journal of Information Technology & Decision Making*, 2006, 5(3): 437–454.
- [9] Huang W, Wang S Y, Zhang H, et al. Selection of the appropriate lag structure of foreign exchange rates forecasting based on autocorrelation coefficient. *Advances in Neural Networks*, 2006, 3973: 512–517.
- [10] Yu L, Wang S Y, Lai K K. An integrated data preparation scheme for neural network data analysis. *IEEE Transactions on Knowledge and Data Engineering*, 2006, 18(2): 217–230.
- [11] Xiao Y, Xiao J, Wang S Y. A hybrid forecasting model for non-stationary time series: An application to container throughput prediction. *International Journal of Knowledge and Systems Sciences*, 2012, 3(2): 67–81.
- [12] Yu L, Wang S Y, Lai K K, et al. Developing and assessing an intelligent forex rolling forecasting and trading decision support system for online e-service. *International Journal of Intelligent Systems*, 2007, 22(5): 475–499.
- [13] Xiao Y, Xiao J, Liu J, et al. A Multiscale modeling approach incorporating ARIMA and ANNs for financial market volatility forecasting. *Journal of Systems Science and Complexity*, 2014, 27(1): 225–236.
- [14] Yu L, Wang S Y, Lai K K. *Foreign-exchange-rate forecasting with artificial neural networks*. Springer, New York, 2007.
- [15] Xiao Y, Liu J J, Hu Y, et al. A neuro-fuzzy combination model based on singular spectrum analysis for air transport demand forecasting. *Journal of Air Transport Management*, 2014, 39: 1–11.
- [16] Lam W H K, Ng P L P, Seabrooke W, et al. Forecasts and reliability analysis of port cargo throughput in Hong Kong. *Journal of Urban Planning and Development-asce*, 2004, 130(3): 133–144.
- [17] Yu L, Wang S Y, Lai K K. Credit risk assessment with a multistage neural network ensemble learning approach. *Expert Systems with Applications*, 2008, 34(2): 1434–1444.
- [18] Yu L, Chen H H, Wang S Y, et al. Evolving least squares support vector machines for stock market trend mining. *IEEE Transactions Evolutionary Computation*, 2009, 13(1): 87–102.
- [19] Yu L, Wang S Y, Lai K K. A neural-network-based nonlinear metamodeling approach to financial time series forecasting. *Applied Soft Computing*, 2009, 9(2): 563–574.
- [20] Yu L, Wang S Y, Lai K K, et al. A multiscale neural network learning paradigm for financial crisis forecasting. *Neurocomputing*, 2010, 73(4–6): 716–725.
- [21] Wang Y Q, Wang S Y, Lai K K. Measuring financial risk with generalized asymmetric least squares regression. *Applied Soft Computing*, 2011, 11(8): 5793–5800.
- [22] Liu Q S, Wang J. Finite-time convergent recurrent neural network with a hard-limiting activation function for constrained optimization with piecewise-linear objective functions. *IEEE Transactions on Neural Networks*, 2011, 22(4): 601–613.
- [23] Kuo R J, Han Y S. A hybrid of genetic algorithm and particle swarm optimization for solving bi-level linear programming problem — A case study on supply chain model. *Applied Mathematical Modelling*, 2011, 35(8): 3905–3917.
- [24] Wei H L, Billings S A, Zhao Y F, et al. Lattice dynamical wavelet neural networks implemented using particle swarm optimization for spatio-temporal system identification. *IEEE Transactions on Neural Networks*, 2009, 20(1): 181–185.
- [25] Alfi A, Modares H. System identification and control using adaptive particle swarm optimization. *Applied Mathematical Modelling*, 2011, 35(3): 1210–1221.
- [26] Wang P C, Shoup T E. A poly-hybrid PSO optimization method with intelligent parameter adjustment. *Advances in Engineering Software*, 2011, 42(8): 555–565.
- [27] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 1989, 2(5): 359–366.
- [28] Ratnawecra A, Halgamuge S. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 240–255.