Knowledge-Based Systems 37 (2013) 378–387

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

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

A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm

Hong-ze Li, Sen Guo *, Chun-jie Li, Jing-qi Sun

School of Economics and Management, North China Electric Power University, Beijing 102206, China

ARTICLE INFO

Article history:
Received 4 April 2012
Received in revised form 30 May 2012
Accepted 18 August 2012
Available online 30 August 2012

Keywords:
Annual power load forecasting
Generalized regression neural network
Fruit fly optimization algorithm
Optimization problem
Parameter selection

ABSTRACT

Accurate annual power load forecasting can provide reliable guidance for power grid operation and power construction planning, which is also important for the sustainable development of electric power industry. The annual power load forecasting is a non-linear problem because the load curve shows a non-linear characteristic. Generalized regression neural network (GRNN) has been proven to be effective in dealing with the non-linear problems, but it is very regretfully finds that the GRNN have rarely been applied to the annual power load forecasting. Therefore, the GRNN was used for annual power load forecasting in this paper. However, how to determine the appropriate spread parameter in using the GRNN for power load forecasting is a key point. In this paper, a hybrid annual power load forecasting model combining fruit fly optimization algorithm (FOA) and generalized regression neural network was proposed to solve this problem, where the FOA was used to automatically select the appropriate spread parameter value for the GRNN power load forecasting model. The effectiveness of this proposed hybrid model was proved by two experiment simulations, which both show that the proposed hybrid model outperforms the GRNN model with default parameter, GRNN model with particle swarm optimization (PSOGRNN), least squares support vector machine with simulated annealing algorithm (SALSSVM), and the ordinary least squares linear regression (OLS_LR) forecasting models in the annual power load forecasting.

1. Introduction

Power load forecasting is an important part of management modernization of electric power systems, which has attracted more and more attentions from the academic and the practice. Annual power load forecasting is of crucial importance to the economic operation of electric power systems and the reliability of electric networks. Accurate annual power load forecasting can relieve the conflict between electricity supply and demand. With the increasing energy shortage pressure, many countries concentrate to transform management philosophy and promote technological innovation. The smart grid, a digitally enabled electrical grid, is taken as one of the future power grid development goals. On May 21, 2009, the State Grid Corporation of China proposed the development planning of constructing 'Strong Smart Grid' in China. With the construction and development of smart grid, the generation capacity of renewable distributed energy will be improved, which will exert significant impacts on the secure and stable operation of electric power grid. Therefore, more accurate annual power load forecasting which is quite important for maintaining secure and stable operation of electric power grid is needed. However, because annual power load has complex and non-linear relationships with several factors such as political environment, economic policy, human activities, irregular behaviors and other non-linear factors, it is quite difficulty for forecasting power load accurately.

Many annual power load forecasting methods developed by the scholars and practitioners have been proposed to increase the forecasting accuracy in the past few years. The traditional forecasting methods, such as regression models [1,2] and time series technology [3–5], have the disadvantage of poor non-linear fitting capability. In recent years, owing to the development of intelligence techniques, many new intelligence forecasting methods were used for annual power load forecasting. Wang et al. [6] proposed a hybrid model combining support vector regression and differential evolution algorithm to forecast the annual load, and this method was proved to outperform the SVR model with default parameters, regression forecasting model and back propagation artificial neural network (BPNN). Pai and Hong [7] used support vector machines with simulated annealing algorithm (SVMSA) to forecast Taiwan's electricity load, and the empirical results revealed that the SVMSA model outperforms the general regression neural networks model and the autoregressive integrated moving average (ARIMA) model. Hong [8] proposed an electric load forecasting model which combined the seasonal recurrent support vector regression model with chaotic artificial bee colony algorithm (SRSVRCABC), which

* Corresponding author. Tel.: +86 15811424568; fax: +86 10 80796904.
E-mail address: guosen324@163.com (S. Guo).

0950-7051/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2012.08.015

yields more accurate forecasting results than TF-ε-SVR-SA and ARIMA models. Kandil et al. [9] implemented a knowledge-based expert system (ES) to support the choice of the most suitable load forecasting model, and the usefulness of this method was demonstrated by a practical application. Chen [10] proposed a collaborative fuzzy-neural approach which multiple experts construct their own fuzzy back propagation networks from various viewpoints for Taiwan's annual electricity load forecasting and the precision and accuracy were improved. Meng and Niu [11] applied the partial least squares method which could quantificational simulate the relationship between the electricity consumption and its factors to forecast electricity load, and it was proved to be effective. Abou El-Ela et al. [12] proposed the artificial neural network (ANN) technique for long-term peak load forecasting, and it was applied on the Egyptian electrical network dependent on its historical data. In order to predict the regional peak load of Taiwan, Hsu and Chen [13] formulated an artificial neural network model by collecting empirical data. Xia et al. [14] developed a medium and long term load forecasting model by using radial basis function neural networks (RBFNN), and the result indicated that the proposed model has a high accuracy and stability.

The generalized regression neural network (GRNN) which was developed by Specht [15] is a kind of probabilistic neural networks and also a powerful regression tool with a dynamic network structure. Because of the strong non-linear mapping capability, the simplicity of network structure and high fault tolerance and robustness, the GRNN can effectively solve the non-linear problems, and it has been widely applied to a variety of fields including pattern recognition [16], short-term load forecasting [17], the modeling and monitoring of batch processes [18], TWUSM drive system [19], medicinal chemistry [20], coal desulfurization [21], exchange rates forecasting [22], sales forecasting [23], wind speed forecasting [24], and so on. However, it is very regretfully finds that the GRNN have rarely been applied to the annual power load forecasting. This paper elucidates the feasibility of using the GRNN to forecast annual power load. However, the shortcoming of applying the GRNN model is that it is very difficult to select the spread parameter properly. Polat and Yıldırım [16] used genetic algorithm to optimize the spread parameter of the GRNN for pattern recognition, and this optimized GRNN can provide higher recognition ability compared with the unoptimized GRNN. However, most researchers selected the spread parameter by experience and a lot of experiments [17–22].

Fruit fly optimization algorithm (FOA) proposed by the scholar Pan [25] is a novel evolutionary computation and optimization technique. This new optimization algorithm has the advantages of being easy to understand and to be written into program code which is not too long compared with other algorithms. Therefore, this paper attempted to use the FOA to automatically select the spread parameter value of the GRNN for improving the GRNN's forecasting accuracy in the annual power load forecasting.

The rest of this paper is organized as follows: Section 2 introduces the GRNN and FOA methods, then a hybrid forecasting model combined GRNN and FOA for annual power load forecasting is discussed in detail. Section 3 introduces the process of the sample data used in this paper and further computations, comparisons and discussions of two examples are presented. Section 4 concludes this paper.

2. Generalized regression neural network with fruit fly optimization algorithm

2.1. Generalized regression neural network

The generalized regression neural network (GRNN) is a kind of radial basis function (RBF) networks which is based on a standard statistical technique called kernel regression. The GRNN has excellent performances on approximation ability and learning speed, and it is fast learning and convergence to the optimal regression surface as the number of sample data becomes very large. When the number of sample data is small, the GRNN still has a good forecasting result [26].

The main function of the GRNN is to estimate a non-linear or linear regression surface on independent variable (also called input vector) $X = [x_1, x_2, \ldots, x_n]^T$, given the dependent variable (also called output vector) $Y = [y_1, y_2, \ldots, y_k]^T$. The procedure of the GRNN model can be represented as

$$E[Y|X] = \frac{\int_{-\infty}^{\infty} Yf(Y,X)dX}{\int_{-\infty}^{\infty} f(Y,X)dX} \quad (1)$$

where X is a n-dimensional input vector, Y is the predicted value of the GRNN model, $E[Y|X]$ is the expected value of the output Y, given the input vector X,$f(Y,X)$ is the joint probability density function of X and Y.

The GRNN is organized using four layers: input layer, pattern layer, summation layer, and output layer, just as shown in Fig. 1.

The input layer receives information and stores an input vector X, which the number of neurons equals to the dimension of input vector. Then, the input neurons of input layer feed the data to the pattern layer. The pattern layer possesses a non-linear transformation from the input space to the pattern space. The neurons in the pattern layer (also called pattern neurons) can memorize the relationship between the input neuron and the proper response of pattern layer, and the number of neurons equals to the number of input variables. The pattern Gaussian function of p_i is expressed as

$$p_i = \exp\left[-\frac{(X - X_i)^T(X - X_i)}{2\sigma^2}\right] \quad (i = 1, 2, \ldots, n) \quad (2)$$

where σ denotes the smoothing parameter, X is the input variable of the network, X_i is a specific training vector of the neuron i in the pattern layer.

The summation layer has two summations, namely S_s and S_w. The simple summation S_s computes the arithmetic sum of the pattern layer outputs, and the interconnection weight equals to '1'. The weighted summation S_w computes the weighted sum of the pattern layer outputs, and the interconnection weight is w. The transfer functions can be represented as Eqs. (3) and (4), respectively:

$$S_s = \sum_{i=1} p_i \quad (3)$$

$$S_w = \sum_{i=1} w_i p_i \quad (4)$$

where w_i is the weight of pattern neuron i connected to the summation layer.

The number of neurons in the output layer equals to the dimension k of output vector Y. After the summations of neurons in the summation layer are fed into the output layer, the output Y of the GRNN model can be calculated as follows:

$$Y = S_s/S_w \quad (5)$$

Therefore, the GRNN model has only one parameter σ that needs to be determined, which is very important in using GRNN for forecasting. The parameter σ (also called 'spread' in Matlab program) determines the generalization capability of the GRNN. Many researchers selected the parameter by priori knowledge or individual experience, which may be un-efficient for forecasting. Therefore, we should develop an automatically efficiently method for selecting the appropriate spread parameter in the GRNN model.

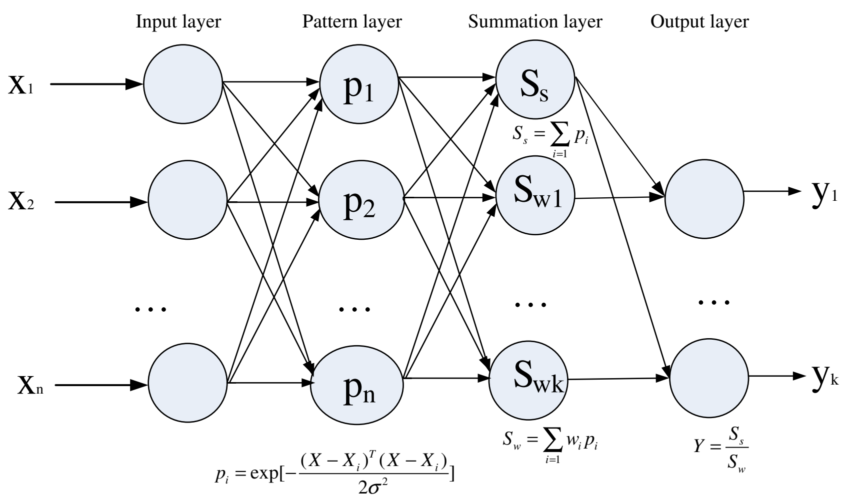

Fig. 1. Schematic diagram of the GRNN architecture.

In order to achieve this goal, this paper used the fruit fly optimization algorithm (FOA) to automatically determine the spread parameter value of the GRNN model.

2.2. Fruit fly optimization algorithm

Fruit fly optimization algorithm (FOA) is a new swarm intelligence method, which was proposed by Pan [25], and it belongs to a kind of interactive evolutionary computation. The FOA is a new method for finding global optimization based on the food finding behavior of the fruit fly. Fruit fly is a kind of insect, which lives in the temperate and tropical climate zones and eats rotten fruit (as illustrated in Fig. 2). The fruit fly is superior to other species in vision and osphresis. The food finding process of fruit fly is as follows: firstly, it smells the food source by osphresis organ, and flies towards that location; then, after it gets close to the food location, the sensitive vision is also used for finding food and other fruit flies' flocking location, and it flies towards that direction. Fig. 3 shows the food finding iterative process of fruit fly swarm.

According to the food finding characteristics of fruit fly swarm, the FOA can be divided into several steps, just as followings:

Step 1. *Parameters initialization.*
The main parameters of the FOA are the maximum iteration number *maxgen*, the population size*sizepop*, the initial fruit fly swarm location (X_axis, Y_axis), and the random flight distance range *FR*.

Step 2. *Population initialization.*
Give the random flight direction and the distance for food finding of an individual fruit fly by using osphresis.

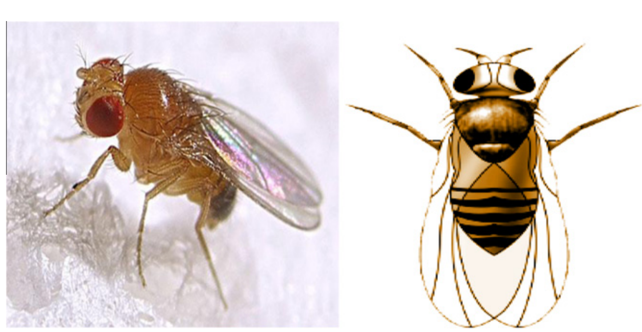

Fig. 2. Fruit fly and its body look illustration.

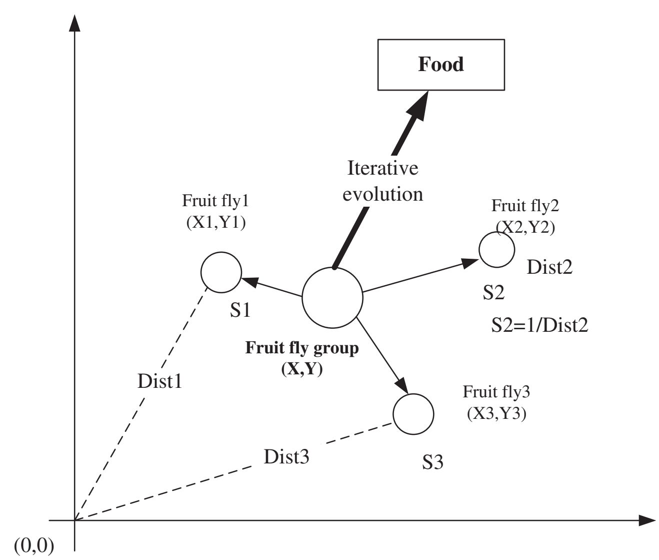

Fig. 3. Food finding iterative process of fruit fly swarm.

$$X_i = X\_axis + \text{Random Value} \quad (6)$$
$$Y_i = Y\_axis + \text{Random Value} \quad (7)$$

Step 3. *Population evaluation.*
Firstly, the distance of the food location to the origin (*Dist*) needs to be calculated. Secondly, the smell concentration judgment value (*S*) needs to be calculated, and the value of *S* is the reciprocal of the distance of the food location to the origin (*Dist*).

$$Dist_i = \left(X_i^2 + Y_i^2\right)^{1/2} \quad (8)$$
$$S_i = 1/Dist_i \quad (9)$$

Then, calculate the smell concentration ($Smell_i$) of the individual fruit fly location by substituting smell concentration judgment value (S_i) into the smell concentration judgment function (also called Fitness function). Finally, find out the individual fruit fly with the maximal smell concentration (the maximal value of $Smell_i$) among the fruit fly swarm.

$$Smell_i = \text{Function } (S_i) \tag{10}$$

$$[bestSmell\ bestIndex] = \max(Smell_i) \tag{11}$$

Step 4. *Selection operation.*
Keep the maximal smell concentration value and x, y coordinate. Then, the fruit fly swarm flies towards that location with the maximal smell concentration value by using vision. Enter iterative optimization to repeat the implementation of step 2–3. When the smell concentration is not superior to the previous iterative smell concentration any more or the iterative number reaches the maximal iterative number, the circulation stops.

$$Smellbest = bestSmell \tag{12}$$

$$X\_axis = X(bestIndex) \tag{13}$$

$$Y\_axis = Y(bestIndex) \tag{14}$$

2.3. Fruit fly optimization algorithm for the parameter selection of the GRNN model

Selecting appropriate expansion speed 'spread' value of radial basis function is extremely important. In this paper, the FOA was used for selecting the appropriate spread parameter value of the GRNN model in order to effectively improve the load forecasting accuracy.

The flowchart of the FOA for the parameter selection of the GRNN model (abbreviated as FOAGRNN) is shown in Fig. 4.

The details of the FOAGRNN model are shown as follows:

Step 1: *Initialization parameters.*
The maximum iteration number *maxgen*, the population size *sizepop*, the initial fruit fly swarm location (*X_axis*, *Y_axis*), and the random flight distance range *FR* should be determined at first. In this study, suppose *maxgen* = 100, *sizepop* = 10, $(X\_axis, Y\_axis) \subset [0,1]$, $FR \subset [-10,10]$.

Step 2: *Evolution starting.*
Set *gen* = 0, and give the random flight direction and the distance for food finding of an individual fruit fly.

Step 3: *Preliminary calculations.*
Calculate the flight distance $Dist_i$ of food finding of the fruit fly i, and then calculate the smell concentration judgment value S_i. Input S_i into the GRNN model for annual load forecasting. According to the load forecasting result, calculate the smell concentration $Smell_i$ (also called the fitness function value). The $Smell_i$ is employed the root-mean-square error (RMSE) which measures the deviation between the forecasting value and the actual value.

Step 4: *Offspring generation.*
The offspring generation is generated according to Eqs. (6)–(14). Then input the offspring into the GRNN model and calculate the smell concentration value again. Set *gen* = *gen* + 1.

Step 5: *Circulation stops.*
When *gen* reaches the max iterative number, the stop criterion satisfies, and the best parameter value of the GRNN model can be obtained. Otherwise, go back to Step 2.

3. Examples computation and comparison analysis

Two examples were selected to prove the effectiveness of the proposed FOAGRNN model in this paper, which are the annual electricity consumption of Beijing city and that of China. In the following sub-sessions, since the raw data processing method and the selected comparison models for these two examples are same, the procedure of applying the proposed FOAGRNN method to forecast the annual electricity consumption of Beijing city and the corresponding comparison analysis will be illustrated in detail (as shown from Sections 3.1, 3.2, 3.3), and the annual electricity consumption of China will be then forecasted using the same procedure (as shown in Section 3.4).

3.1. Forecast Beijing's annual electricity consumption using the proposed FOAGRNN model

The procedure for applying the proposed FOAGRNN method to forecast the Beijing's annual electricity consumption is described as follows:

Step 1. *The process of sample data.*
The sample data were selected from the annual electricity consumption of Beijing city (the capital of China) between 1978 and 2010, which includes 33 data points (as shown in Table 1). In this paper, the sample data were normalized to make data in the range from 0 to 1 using the following formula:

$$Z = \{z_i\} = \frac{x_i - x_{imin}}{x_{imax} - x_{imin}}, \quad i = 1, 2, 3 \tag{15}$$

where x_{imin} and x_{imax} are the minimal and maximal value of each input factor, respectively.

The sample data were divided into the training sample data and testing sample data. Different from the short load forecasting, the annual power load forecasting is not suitable for selecting the factors such as temperature, moderate. Therefore, this paper selected the last three load data (L_{n-3}, L_{n-2}, L_{n-1}) as the input variables of the FOAGRNN model, and the output variable is L_n. This method has been proven effective and feasible for annual power load forecasting [6]. Due to using the three load data as the input variables to forecast, the training sample data is started in 1981 and ended in 2005, and the testing sample data is from 2006 to 2010.

A roll-based forecasting process was used in the training stage. Firstly, the top three load data (from 1978 to 1980) of sample data were substituted into the FOAGRNN model, and the first load forecasting value of 1981 can be obtained. Secondly, the next roll-top three load data (from 1979 to 1981) were fed into the FOAGRNN model, and the forecasting value of 1982 can be gotten. In this step, the load value of 1981 which was fed into the proposed model should employ the real load value of 1981 in the load series. Similarly, the processes were cycling until all the load forecasting values (from 1981 to 2005) were obtained.

Step 2. *Train the GRNN model.*
In the FOAGRNN model, the spread parameter value of the GRNN model is dynamically tuned by the FOA. The initial spread parameter value of the GRNN model was set up in the range of [0.0001, 1]. Fig. 5 shows the fruit fly swarm flying route for optimization parameter, and the iterative RMSE trend of the FOAGRNN searching of optimization parameter is shown in Fig. 6. After 100 times of iterative evolution, the convergence can be seen in generation 23 with coordinate of (25.1286, 39.7878), and the RMSE value and the spread parameter value are 0.0189, 0.0191, respectively.

Step 3. *Forecast the new annual electricity consumption values.*
According to the result of the FOA tuning the spread parameter of the GRNN model, the spread parameter value of the GRNN model was chosen as 0.0191 to forecast the Beijing's annual electricity consumption from 2006 to 2010. The final forecasted electricity consumption values of Beijing city from 2006 to 2010 are shown in Table 2.

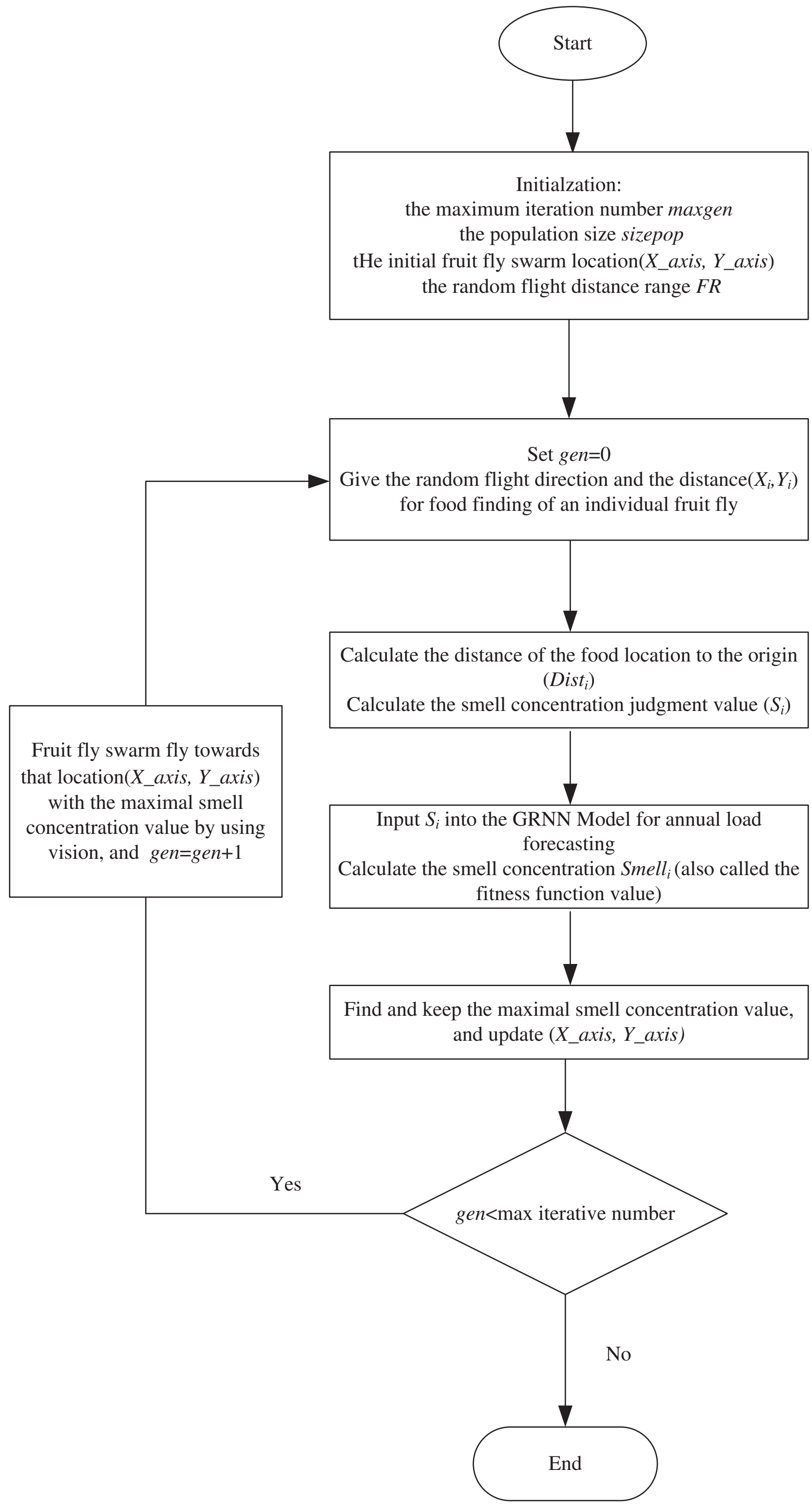

Fig. 4. The flowchart of the FOAGRNN model.

3.2. Select the comparison models

To compare the power load forecasting results, several other power load forecasting models should be selected. The other power load forecasting models for comparison with the FOAGRNN model also make power load forecasting based on the given sample data.

Just as shown in Fig. 7, it can be clearly seen that the annual load series shows an increasingly approximate linear trend. Therefore, the ordinary least squares linear regression (OLS_LR) forecasting model is employed. In the meantime, the GRNN forecasting model with default parameter, the GRNN model with particle swarm optimization (PSOGRNN), and the least squares support

Table 1
Annual electricity consumption of Beijing city and China between 1978 and 2010 (unit: 10^9 kW h).

Year	Beijing city	China
1978	7.350	246.53
1979	8.023	282.02
1980	8.546	300.63
1981	8.672	309.65
1982	9.257	327.92
1983	9.563	351.86
1984	10.294	377.89
1985	11.063	411.90
1986	11.812	451.03
1987	12.850	498.84
1988	13.786	547.23
1989	14.218	587.18
1990	15.048	623.59
1991	16.140	680.96
1992	17.596	759.27
1993	19.250	842.65
1994	20.545	926.04
1995	22.259	1002.34
1996	24.437	1076.43
1997	26.361	1128.44
1998	27.621	1159.84
1999	29.726	1230.52
2000	38.443	1347.24
2001	39.994	1463.35
2002	43.996	1633.15
2003	46.761	1903.16
2004	51.011	2197.14
2005	56.704	2494.03
2006	61.899	2858.80
2007	67.509	3271.18
2008	70.815	3454.14
2009	75.885	3703.22
2010	83.090	4199.90

Source: Energy Statistical Yearbook of China, 2011.

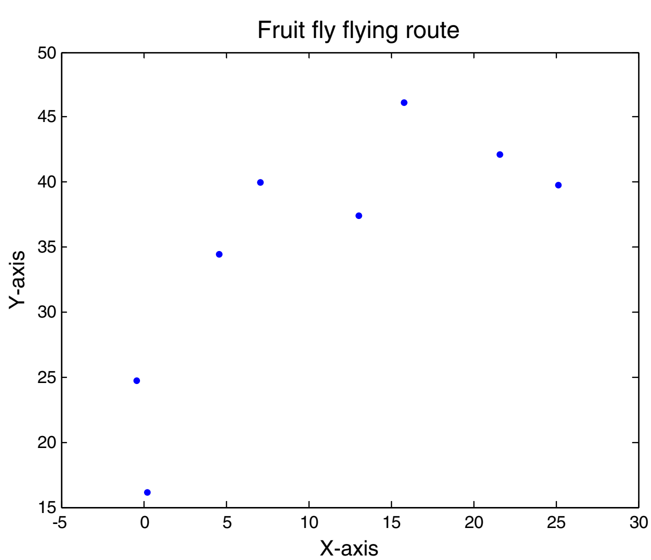

Fig. 5. The fruit fly swarm flying route for optimization parameter in the example of Beijing city.

vector machine with simulated annealing algorithm (SALSSVM) forecasting model are also employed for comparison. The structures of GRNN and PSOGRNN models are the same as that of the FOAGRNN model. Least squares support vector machine (LSSVM) is reformulation to support vector machine (SVM) which lead to solving linear KKT system [27,28]. LSSVM can approach the non-linear system with high precision, which is a powerful tool for

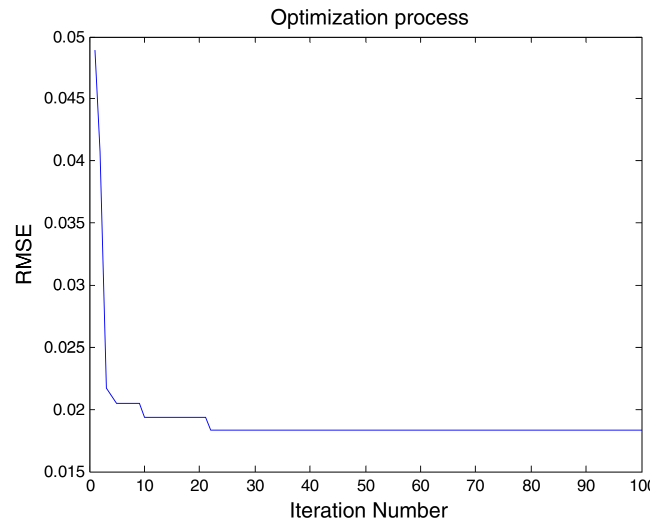

Fig. 6. The iterative RMSE trend of the FOAGRNN searching of optimization parameter in the example of Beijing city.

the modeling and forecasting of the non-linear system [29]. LSSVM model with Gaussian RBF kernel includes two parameters: γ is the regularization parameter which determines the trade-off between the training error minimization and smoothness; σ^2 is the squared bandwidth of the Gaussian RBF kernel. In order to improve the forecasting accuracy, the parameters of the LSSVM model should be tuned. Pai and Hong [7] applied the simulated annealing algorithm to optimize the parameters of LSSVM model for annual load forecasting, and obtained good results. Therefore, this study also selects SALSSVM forecasting model for comparison.

3.3. Comparisons of the FOAGRNN, GRNN, PSOGRNN, SALSSVM and OLS_LR models

The experiment's environment includes Matlab 2010a, GRNN toolbox, LSSVMlabv1.8 toolbox, self-written MATLAB program and the computer with the Intel(R) Core(TM)2 T2450 2 GHz CPU, 1.5 GB RAM and Windows 7 professional system.

In the GRNN model without using FOA, the parameter was chosen as 0.3. In the PSOGRNN model, the particle swarm optimization was used to automatically select the spread parameter value of the GRNN model, and the initialization parameters were set as: $maxgen = 100$; $sizepop = 10$; $Vmax = 1$; $Vmin = -1$; $popmax = 5$; $popmin = -5$. By simulation, the convergence can be seen in generation 66, and the *bestfitness* value and the spread parameter value are 0.0794, 0.0258, respectively. In the SALSSVM model, radial basis function was chosen as the kernel function. According to the result of SA optimizing the parameters of the LSSVM model, the convergence was obtained in generation 13, and the parameter γ value and the parameter σ value are respectively [2485.8411, 38.034].

With the FOAGRNN model, GRNN model, PSOGRNN model, SALSSVM model and OLS_LR model, the training times of the sample data are 51 s, 23 s, 113 s, 76 s, 11 s, respectively. The training times of five models on disposing the training data are different. The FOAGRNN, PSOGRNN and SALSSVM models use longer time than the GRNN and OLS_LR models because they need determine the parameters in the each generation. However, the PSOGRNN uses 62 s longer than the FOAGRNN computation, and the SALSSVM uses 25 s longer than FOAGRNN computation.

Table 2 and Fig. 8 give the annual power load forecasting results with the FOAGRNN, GRNN, PSOGRNN, SALSSVM and OLS_LR models. Fig. 9 describes the error analysis of the five forecasting models. Table 2 also lists the relative errors of the five forecasting

Table 2
Forecasting results of Beijing's annual electricity consumption using FOAGRNN, GRNN, PSOGRNN, SALSSVM and OLS_LR models (unit: 10^9 kW h).

Year	Actual value	FOAGRNN		GRNN		PSOGRNN		SALSSVM		OLS_LR	
		Result	Error (%)	Result	Error (%)	Result	Error (%)	Result	Error (%)	Result	Error (%)
2006	61.899	61.981	0.132	63.685	2.885	62.066	0.270	62.018	0.192	62.140	0.389
2007	67.509	67.144	−0.541	66.380	−1.672	67.213	−0.438	67.426	−0.123	68.129	0.918
2008	70.815	72.914	2.964	72.295	2.090	75.033	5.956	73.266	3.461	74.611	5.360
2009	75.885	77.484	2.107	77.967	2.744	77.486	2.110	78.059	2.865	79.434	4.677
2010	83.089	83.090	−0.001	80.955	−2.570	82.666	−0.510	82.929	−0.194	85.065	2.377

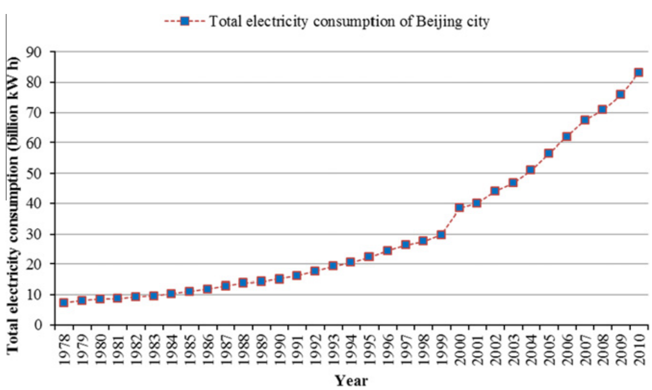

Fig. 7. Total electricity consumption of Beijing city between 1978 and 2010.

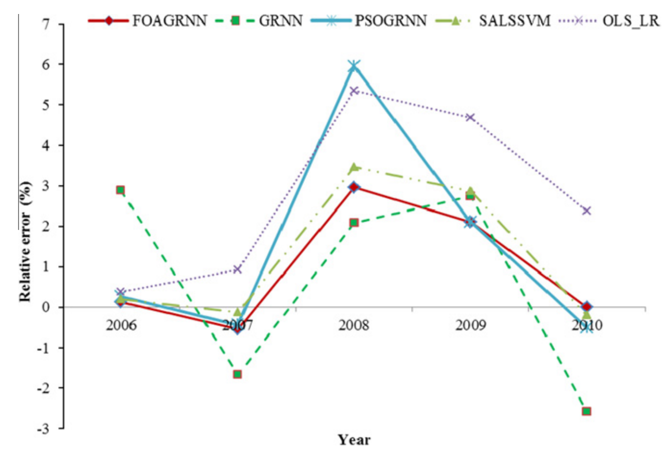

Fig. 9. Error analysis.

models. According to Fig. 8, it can be clearly seen that all of the five forecasting models capture the increasing trends, but the performance of four intelligence load forecasting models of FOAGRNN, GRNN, PSOGRNN, SALSSVM are better than the OLS_LR model. From Table 2 and Fig. 9, the deviation between the forecasting results of the five forecasting models and the actual value can be captured. The error range [−3%,+3%] is always considered as a standard to measure the performance of the forecasting model [30]. Therefore, this paper uses this range to compare the five load forecasting models. Firstly, the relative errors of the power load forecasting points of the FOAGRNN model are all in this error range, and the maximum and minimum relative errors are 2.964% and −0.001% in 2008 and 2010, respectively. In addition, out of 5 points 3 that means 60% of the forecasting points are in the scope of [−1%,+1%]. Secondly, the relative errors of the power load forecasting points of GRNN model are also all in the error range, and the maximum and minimum relative errors are 2.885% and −1.672% in 2006 and 2007, respectively. However, all

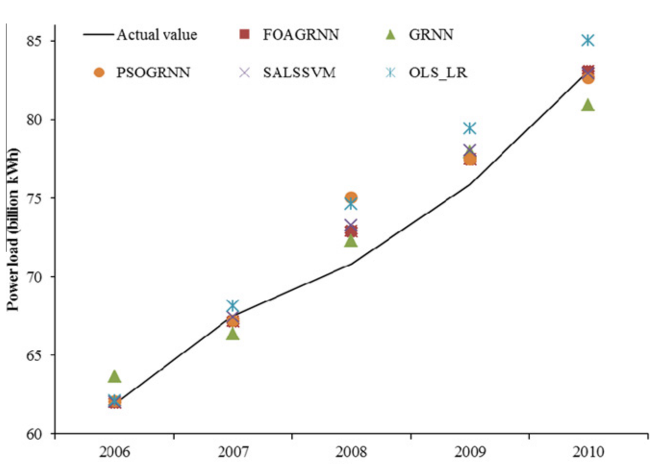

Fig. 8. Forecasting results of Beijing's annual electricity consumption from 2006 to 2010.

the forecasting points exceed the scope of [−1%,+1%]. Thirdly, the PSOGRNN model has one forecasting result point exceed the error range, which is 5.956% in 2008, and the maximum and minimum relative errors are 5.956% and 0.270% in 2008 and 2006, respectively. 60% of the forecasting points of the PSOGRNN model are in the scope of [−1%,+1%]. Fourthly, the SALSSVM model has one forecasting result point exceed the error range, which is 3.461% in 2008, and the maximum and minimum relative errors are 3.461% and −0.123% in 2008 and 2007, respectively. 60% of the forecasting points of the SALSSVM model are in the scope of [−1%,+1%]. Finally, the OLS_LR model has two forecasting result points exceed the error range, which is 5.360% and 4.677% in 2008 and 2009, and the maximum relative errors reaches 5.360%. 40% of the forecasting points of the OLS_LR model are in the scope of [−1%,+1%].

The mean absolute percentage error (*MAPE*) and mean square error (*MSE*) are always used to measure the performance of the forecasting models in the power load forecasting. The *MAPE* and *MSE* values can be calculated by

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A(i)-F(i)}{A(i)}\right| \times 100\% \quad (16)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(A(i)-F(i))^2 \quad (17)$$

where $A(i)$ is the actual power load value at time i, $F(i)$ is the forecasting power load value at time i, *MAPE* is the mean absolute percentage error of the forecasting power load, and *MSE* is the mean square error of the forecasting power load.

The *MAPE* and *MSE* values of the five forecasting models are listed in Table 3. It can be seen that the *MAPE* and *MSE* values of the proposed FOAGRNN model are 1.149% and 1.421, respectively, which are both the smallest among the five forecasting models. This means that the proposed FOAGRNN model has the best performance in the annual load forecasting results. Second is the SALS-

SVM model, the *MAPE* and *MSE* values are 1.367% and 2.156, respectively, which are both larger than that of FOAGRNN model and smaller than the remaining three models. Although the forecasting points of the GRNN model with default parameter are all in the error range, the *MAPE* and *MSE* values are 2.392% and 3.110, respectively, much larger than that of FOAGRNN model. The *MAPE* value of PSOGRNN model is 1.857%, larger than that of FOAGRNN model, and the *MSE* value is 4.130, dramatically larger than that obtained by FOAGRNN model. This shows that the performance of the GRNN model optimized by FOA is much better than the GRNN model optimized by PSO. The *MAPE* and *MSE* values of the OLS_LR model are 2.744% and 6.270, respectively, the largest among the five forecasting models, which shows the poorest performance.

The electricity consumption of Beijing city in 2011 and 2012 are also forecasted by the five models, and the corresponding results are shown in Table 4.

3.4. Further simulation

The annual electricity consumption values of China were also predicted using the five models. The annual electricity consumption of China between 1978 and 2010 is shown in Table 1.

In the FOAGRNN model, the fruit fly swarm flying route for optimization parameter is shown in Fig. 10, and the iterative RMSE trend of the FOAGRNN searching of optimization parameter is shown in Fig. 11. The convergence can be seen in generation 29 with coordinate of (36.6992, 33.5863), and the RMSE value and the spread parameter value are 0.0175, 0.0324, respectively. In the PSOGRNN model, the convergence can be seen in generation 10, and the *bestfitness* value and the spread parameter value are 0.0919, 0.0725, respectively. In the SALSSVM model, the convergence was obtained in generation 11, and the parameter γ value and the parameter σ value are respectively [60082.7431, 3.32012].

With the FOAGRNN model, GRNN model, PSOGRNN model, SALSSVM model and OLS_LR model, the training times of the sample data were 50 s, 23 s, 111 s, 77 s, 13 s, respectively. This again reveals that the proposed FOAGRNN model uses much shorter training time than the PSOGRNN and SALSSVM models.

Table 5 lists the China's annual power load forecasting results and the relative errors with the FOAGRNN, GRNN, PSOGRNN, SALSSVM and OLS_LR models. The relative errors of power load forecasting points using FOAGRNN model are all in the error range [−3%, +3%], which the maximum is −2.64% in 2007, and two forecasting points are in the scope of [−1%, +1%]. The GRNN model has three forecasting result points exceed the error range, which the maximum is −4.67% in 2007, and one point (−0.37% in 2008) is in the scope of [−1%, +1%]. The PSOGRNN model has one forecasting result point exceed the error range, which is 5.16% in 2009, and 40% of the forecasting points are in the scope of [−1%, +1%]. The SALSSVM model has one forecasting result point exceed the error range, which is 5.98% in 2008, and only one forecasting point is in the scope of [−1%, +1%]. The OLS_LR model has three forecasting result points exceed the error range, and the maximum relative errors reaches 7.35% in 2008.

The *MAPE* and*MSE* values for the five models in forecasting China's annual electricity consumption are listed in Table 6. It can be seen that the proposed FOAGRNN model still has the smallest *MAPE* and *MSE* values, which are 1.252% and 2839.47, respectively. This again reveals that the proposed FOAGRNN model has the best performance in the annual power load forecasting results. Moreover, the*MSE* values of the five forecasting models demonstrate the significant differences, which the value of the FOAGRNN model is dramatically smaller than that of other four models. The comparisons also show that the *MAPE* value of the PSOGRNN is smaller than that of the GRNN model, but the*MSE*value is larger, and this

Table 3
MAPE and *MSE* values of the FOAGRNN, GRNN, PSOGRNN, SALSSVM and OLS_LR models in the example of Beijing city.

Index	FOAGRNN	GRNN	PSOGRNN	SALSSVM	OLS_LR
MAPE (%)	1.149	2.392	1.857	1.367	2.744
MSE	1.421	3.110	4.130	2.156	6.270

Table 4
Forecasting results of Beijing's electricity consumption in 2011 and 2012 (unit: 10^9 kW h).

Year	FOAGRNN	GRNN	PSOGRNN	SALSSVM	OLS_LR
2011	943.0293	925.0478	938.2215	948.7083	921.3994
2012	1096.546	1138.145	1076.158	1195.892	986.1772

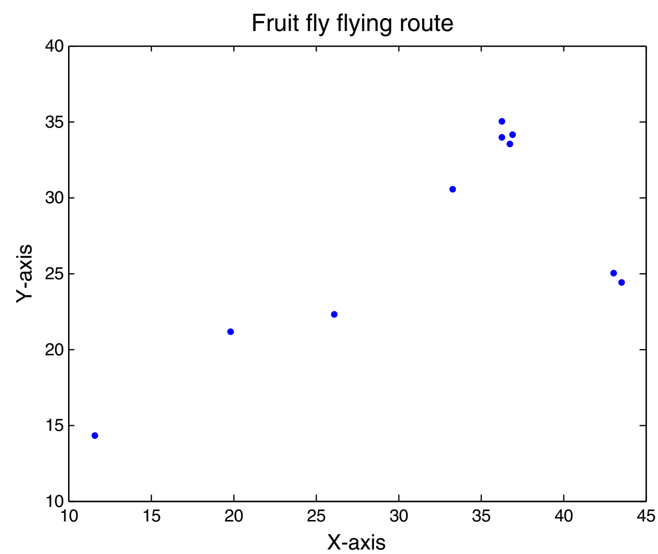

Fig. 10. The fruit fly swarm flying route for optimization parameter in the example of China.

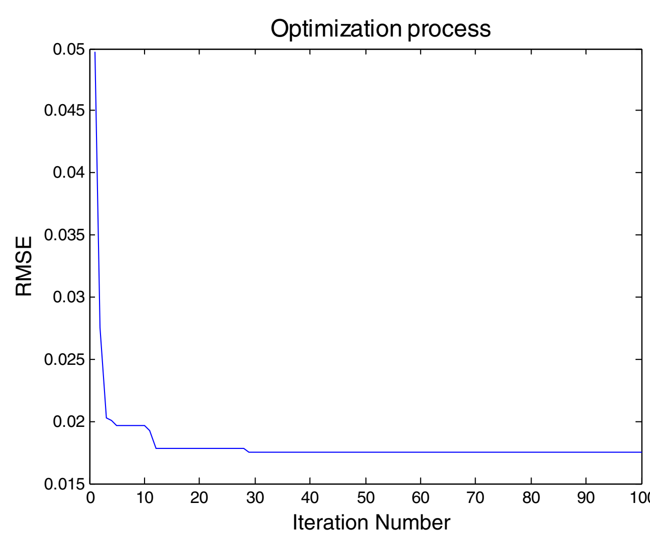

Fig. 11. The iterative RMSE trend of the FOAGRNN searching of optimization parameter in the example of China.

Table 5
Forecasting results of China's annual electricity consumption using FOAGRNN, GRNN, PSOGRNN, SALSSVM and OLS_LR models (unit: 10^9 kW h).

Year	Actual value	FOAGRNN		GRNN		PSOGRNN		SALSSVM		OLS_LR	
		Result	Error (%)	Result	Error (%)	Result	Error (%)	Result	Error (%)	Result	Error (%)
2006	2858.80	2867.99	0.32	2960.08	3.54	2872.11	0.47	2859.92	0.04	2794.15	−2.26
2007	3271.18	3184.81	−2.64	3118.42	−4.67	3183.05	−2.69	3318.76	1.45	3257.77	−0.41
2008	3454.14	3384.02	−2.03	3441.51	−0.37	3395.49	−1.70	3660.80	5.98	3708.16	7.35
2009	3703.22	3744.35	1.11	3779.99	2.07	3894.27	5.16	3788.81	2.31	3591.50	−3.02
2010	4199.90	4193.19	−0.16	4070.52	−3.08	4187.83	−0.29	4320.73	2.88	4068.92	−3.12

Table 6
Comparisons of the forecasting accuracies of the five models in the example of China.

Models	FOAGRNN	GRNN	PSOGRNN	SALSSVM	OLS_LR
MAPE (%)	1.252	2.746	2.533	3.232	2.061
MSE	2839.47	11277.36	13379.51	19704.40	9606.46

Table 7
Forecasting results of China's electricity consumption in 2011 and 2012 (unit: 10^9 kW h).

Year	FOAGRNN	GRNN	PSOGRNN	SALSSVM	OLS_LR
2011	4681.42	4885.27	4703.92	4744.07	4853.09
2012	5262.17	5513.12	5480.90	5490.97	5582.85

result agrees with the one presented in Section 3.3. However, the *MAPE* and *MSE* values of the PSOGRNN are both smaller than that of the SALSSVM model, and this finding does not agree with the one presented in Section 3.3.

The electricity consumption of China in 2011 and 2012 are also forecasted by the five models, just as shown in Table 7.

In summary, the proposed FOAGRNN model can narrow the deviation between the forecasting result and the actual value, and it outperforms the GRNN, PSOGRNN, SALSSVM and OLS_LR models in the annual power load forecasting. Compared with the GRNN model, the FOAGRNN model which uses the FOA to select the spread parameter value of the GRNN model can improve the annual load forecasting accuracy effectively. Compared with the PSOGRNN and SALSSVM models, the FOAGRNN model also has much more excellent performance on the training time and forecasting accuracy. But it is still unclear when the PSOGRNN model performs better than the SALSSVM model.

4. Conclusions

The generalized regression neural network has been widely applied to a variety of fields, but it is very regretfully finds that the GRNN have rarely been applied to the annual power load forecasting. In this paper, a hybrid model based on the generalized regression neural network and fruit fly optimization algorithm was proposed for the annual power load forecasting. This proposed model (FOAGRNN) uses the FOA to automatically select the appropriate spread parameter value of the GRNN model in order to improve the power load forecasting accuracy. For comparison, another four models (GRNN, PSOGRNN, SALSSVM, and OLS_LR) were selected. Taking the annual electricity consumption of Beijing city and that of China from 1978 to 2010 as examples, the power load forecasting results of the FOAGRNN, GRNN, PSOGRNN, SALSSVM and OLS_LR models were obtained. The two experiment results show: Firstly, because of the good non-linear fitting capacity of intelligence forecasting model, the intelligence annual power load forecasting models have better performance than the regression model in this study. Secondly, the FOA can select the appropriate spread parameter value of the GRNN model, which could effectually improve the annual power load forecasting accuracy. Thirdly, compared with the PSOGRNN and SALSSVM models, the training time of the FOAGRNN model is shorter, and the *MAPE* and *MSE*values are both much smaller. Especially, the *MSE* values demonstrate the significant differences among the five forecasting models, and the*MSE* value of the FOAGRNN model is dramatically smaller than that of other four models. This means that the FOAGRNN model has more excellence performance than the PSOGRNN and SALSSVM models on the annual load forecasting. Conclusively, the proposed FOAGRNN model which uses the FOA to automatically select the appropriate spread parameter value of the GRNN model can effectively improve the annual power load forecasting accuracy.

Acknowledgments

This study is supported by the National Natural Science Foundation of China (70971038) and the Beijing Philosophy and Social Science Planning Project (11JGB070). The authors are grateful to the editor and anonymous reviewers for their suggestions in improving the quality of the paper.

References

[1] Antti Sorjamaa, Jin Hao, Nima Reyhani, Yongnan Ji, Amaury Lendasse, Methodology for long-term prediction of time series, Neurocomputer 70 (16–18) (2007) 861–869.
[2] H.M. Ai-Hamadi, S.A. Soliman, Long-term/mid-term electric load forecasting based on short-term correlation and annual growth, Electric Power Systems Research 74 (3) (2005) 353–361.
[3] Ruijun Dong, Witold Pedrycz, A granular time series approach to long-term forecasting and trend forecasting, Physica A: Statistical Mechanics and its Applications 387 (13) (2008) 3253–3270.
[4] S.S. Pappas, L. Ekonomou, D.C. Karamousantas, G.E. Chatzarakis, S.K. Katsikas, P. Liatsis, Electricity demand loads modeling using auto regressive moving average (ARMA) models, Energy 33 (2008) 1353–1360.
[5] S.Sp. Pappas, L. Ekonomou, P. Karampelas, D.C. Karamousantas, S.K. Katsikas, G.E. Chatzarakis, P.D. Skafidas, Electricity demand load forecasting of the Hellenic power system using an ARMA model, Electric Power Systems Research 80 (3) (2010) 256–264.
[6] Jianjun Wang, Li Li, Dongxiao Niu, Zhongfu Tan, An annual load forecasting model based on support vector regression with differential evolution algorithm, Applied Energy 94 (2012) 65–70.
[7] Ping-Feng Pai, Wei-Chiang Hong, Support vector machines with simulated annealing algorithms in electricity load forecasting, Energy Conversion and Management 46 (17) (2005) 2669–2688.
[8] Wei-Chiang Hong, Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm, Energy 36 (9) (2011) 5568–5578.
[9] M.S. Kandil, S.M. El-Debeiky, N.E. Hasanien, The implementation of long-term forecasting strategies using a knowledge-based expert system: Part-II, Electric Power Systems Research 58 (1) (2001) 19–25.
[10] T. Chen, A collaborative fuzzy-neural approach for long-term load forecasting in Taiwan, Computers & Industrial Engineering (2011), http://dx.doi.org/10.1016/j.cie.2011.06.003.
[11] Ming Meng, Dongxiao Niu, Annual electricity consumption analysis and forecasting of China based on few observations methods, Energy Conversion and Management 52 (2) (2011) 953–957.
[12] A.A. Abou El-Ela, A.A. El-Zeftawy, S.M. Allam, G.M. Atta, Long-term load forecasting and economical operation of wind farms for Egyptian electrical network, Electric Power Systems Research 79 (7) (2009) 1032–1037.
[13] Che-Chiang Hsu, Chia-Yon Chen, Regional load forecasting in Taiwan – applications of artificial neural networks, Energy Conversion and Management 44 (12) (2003) 1941–1949.

[14] Changhao Xia, Jian Wang, Karen McMenemy, Short, medium and long term load forecasting model and virtual load forecaster based on radial basis function neural networks, International Journal of Electrical Power & Energy Systems 32 (7) (2010) 743–750.
[15] D.F. Specht, A general regression neural network, IEEE Transactions on Neural Networks 2 (11) (1991) 568–576.
[16] Övünç Polat, Tülay Yıldırım, Genetic optimization of GRNN for pattern recognition without feature extraction, Expert Systems with Applications 34 (4) (2008) 2444–2448.
[17] Changhao Xia, Bangjun Lei, Hongping Wang, Jiangnan Li, GRNN short-term load forecasting model and virtual instrument design, Energy Procedia 13 (2011) 9150–9158.
[18] Savita G. Kulkarni, Amit Kumar Chaudhary, Somnath Nandi, Sanjeev S. Tambe, Bhaskar D. Kulkarni, Modeling and monitoring of batch processes using principal component analysis (PCA) assisted generalized regression neural networks (GRNN), Biochemical Engineering Journal 18 (3) (2004) 193–210.
[19] Tien-Chi Chen, Chih-Hsien Yu, Motion control with deadzone estimation and compensation using GRNN for TWUSM drive system, Expert Systems with Applications 36 (8) (2009) 10931–10941.
[20] Mohsen Shahlaei, Razieh Sabet, Maryam Bahman Ziari, Behzad Moeinifard, Afshin Fassihi, Reza Karbakhsh, QSAR study of anthranilic acid sulfonamides as inhibitors of methionine aminopeptidase-2 using LS-SVM and GRNN based on principal components, European Journal of Medicinal Chemistry 45 (10) (2010) 4499–4508.
[21] S. Chehreh Chelgani, E. Jorjani, Microwave irradiation pretreatment and peroxyacetic acid desulfurization of coal and application of GRNN simultaneous predictor, Fuel 90 (11) (2011) 3156–3163.
[22] Mark T. Leung, An-Sing Chen, Hazem Daouk, Forecasting exchange rates using general regression neural networks, Computers & Operations Research 27 (11–12) (2000) 1093–1110.
[23] Pei-Chann Chang, Chen-Hao Liu, Chin-Yuan Fana, Data clustering and fuzzy neural network for sales forecasting: a case study in printed circuit board industry, Knowledge-Based Systems 22 (5) (2009) 344–355.
[24] Zhen-hai Guo, Jie Wu, Hai-yan Lu, Jian-zhou Wang, A case study on a hybrid wind speed forecasting method using BP neural network, Knowledge-Based Systems 24 (7) (2011) 1048–1056.
[25] Wen-Tsao Pan, A new fruit fly optimization algorithm: taking the financial distress model as an example, Knowledge-Based Systems 26 (2) (2012) 69–74.
[26] Mahmood Amiri, Hamed Davande, Alireza Sadeghian, Sylvain Chartier, Feedback associative memory based on a new hybrid model of generalized regression and self-feedback neural networks, Neural Networks 23 (7) (2010) 69–74.
[27] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.
[28] T. Van Gestel, J.A.K. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, J. Vandewalle, Benchmarking least squares support vector machine classifiers, Machine Learning 54 (1) (2001) 5–32.
[29] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Processing Letters 9 (3) (1999) 293–300.
[30] Dongxiao Niu, Yongli Wang, Desheng Dash Wu, Power load forecasting using support vector machine and ant colony optimization, Expert Systems with Applications 37 (3) (2010) 2531–2539.