Short Term Load Forecasting Based on IGSA- ELM Algorithm

Junwei Cao, Wanlu Zhang
Research Institute of Information Technology
Tsinghua University
Beijing, China
zhangwl15@mails.tsinghua.edu.cn

***Abstract*—This paper proposes a novel short-term load forecasting (STLF) method based on extreme learning machine (ELM) and improved gravitational search algorithm (IGSA). The IGSA is used to search the optimal set of input weights and hidden biases for ELM, adding the ability of exploitation in particle swarm optimization (PSO) to improve the basic gravitational search algorithm (GSA). Based on IGSA-ELM algorithm, a model to predict the maximal load data of the next day has been established, and some comparative experiments between IGSA-ELM and some conventional methods have been done. The results show that the proposed method performs equal to or better than the other algorithms in terms of speed and accuracy.**

Keywords—extreme learning machine; gravitational search algorithm; particle swarm optimization; load forecasting

I. Introduction

Energy Internet is considered as the basic platform for future energy, and the electric energy is the most important part of Energy Internet. With the development of electric equipment, the long-distance distribution of electricity has been achieved and the power grid already has a considerable scale. However, the electricity energy is not easy to store, and the high power loss during the transmission becomes an urgent problem to be solved. To deal with this problem, short-term load forecasting becomes an important research topic to predict the load data of power users and then choose better power-transmission solutions according to the forecast results.

Recently, lots of intelligent algorithms have been used for STLF, such as Back Propagation Neural Networks (BPNN), support vector machines (SVM) and so on [1-3]. However, these methods have some drawbacks. The convergence rate of traditional neural networks such as BPNN is very slow because of using gradient descent, and it is easy to fall into the local minima and then stop learning. While the SVM algorithm could not do well with huge data due to its high computational burden for the constrained optimization.

This paper presents a STLF model based on IGSA-ELM algorithm. ELM was first introduced by Guang-Bin Huang in [4,5]. Unlike those conventional feed-forward neural networks trained by slow gradient-based learning algorithms, the ELM algorithm randomly chooses input's coefficients and biases of hidden layer, raising the training speed and avoiding falling into the local minima. But the predict results changes a lot because of the randomly choosing of input weights and hidden bias. Thus IGSA, which improves GSA method with PSO algorithm, is used to optimize ELM and improve the prediction accuracy.

In this paper, Section II describes the theoretical basis of IGSA-ELM prediction neural network, and Section III introduces the detailed steps of the proposed method applied in STLF. The simulations are presented in Section IV, and Section V outlines conclusions. The experimental results indicate that this IGSA-ELM prediction model performs well on dealing with load forecasting problem.

II. IGSA-ELM Prediction Neural Network

A. Basic ELM Algorithm

The extreme learning machine is a newly developed learning algorithm for single-hidden layer feed-forward neural networks(SLFNs), which was first introduced by Guang-Bin Huang in [4,5]. Unlike those conventional feed-forward neural networks trained by slow gradient-based learning algorithms, the extreme learning machine randomly chooses input's coefficients and biases of hidden layer, and analytically determines the output weights of SLFNs, providing good generalization performance at extremely fast learning speed. Fig. 1 demonstrates the basic structure of the ELM.

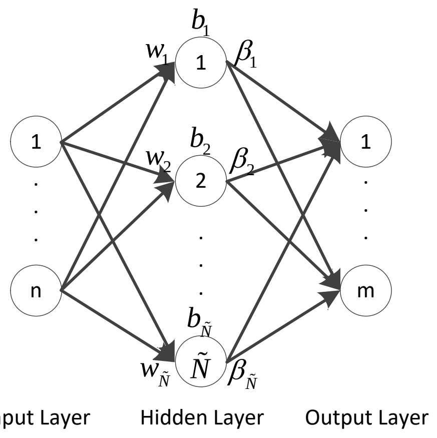

Fig. 1. The basic structure of ELM

Given a training set $\{(x_i, t_i)\}_{i=1}^{N}$ with N distinct samples, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in \mathbf{R}^n$ is the input vector and $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in \mathbf{R}^m$ is the output vector, then the standard SLFNs with $\tilde{N}$ hidden neurons and activation function $g(\mathrm{x})$ can approximate the N samples. Above is formulated as:

$$\mathbf{H}\beta = \mathbf{T} \tag{1}$$

Where,

$$\mathbf{H}(\mathrm{w}_1, \ldots, \mathrm{w}_{\tilde{N}}, \mathrm{b}_1, \ldots, \mathrm{b}_{\tilde{N}}, \mathrm{x}_1, \ldots, \mathrm{x}_N) = \begin{bmatrix} \mathrm{g}(\mathrm{w}_1 \cdot \mathrm{x}_1 + \mathrm{b}_1) \cdots \mathrm{g}(\mathrm{w}_{\tilde{N}} \cdot \mathrm{x}_1 + \mathrm{b}_{\tilde{N}}) \\ \vdots \quad \cdots \quad \vdots \\ \mathrm{g}(\mathrm{w}_1 \cdot \mathrm{x}_N + \mathrm{b}_1) \cdots \mathrm{g}(\mathrm{w}_{\tilde{N}} \cdot \mathrm{x}_N + \mathrm{b}_{\tilde{N}}) \end{bmatrix}_{\mathrm{N}\times\tilde{\mathrm{N}}},$$

$$\beta = \begin{bmatrix} \beta_1^{\mathrm{T}} \\ \vdots \\ \beta_{\tilde{\mathrm{N}}}^{\mathrm{T}} \end{bmatrix}_{\tilde{\mathrm{N}}\times\mathrm{m}} \quad \mathbf{T} = \begin{bmatrix} \mathrm{t}_1^{\mathrm{T}} \\ \vdots \\ \mathrm{t}_{\mathrm{N}}^{\mathrm{T}} \end{bmatrix}_{\mathrm{N}\times\mathrm{m}},$$

and $\mathbf{H}$ is called the hidden layer output matrix of the neural network.

Finally, as demonstrated in [4], the output weights matrix β can be calculated by (2):

$$\beta = \mathbf{H}^{\dagger}\mathbf{T} \tag{2}$$

Where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix $\mathbf{H}$.

Thus, the detailed steps of the ELM algorithm can be described as follows: Firstly, randomly generate the input weights matrix $\mathbf{w}$ and hidden biases vector $\mathbf{b}$, where $\mathbf{w} = [\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}]^T$ and $\mathbf{b} = [b_1, \ldots, b_{\tilde{N}}]^T$ .Then, calculate the hidden layer output matrix $\mathbf{H}$ according to $\mathbf{H} = \mathrm{g}(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$. Finally, the output weights matrix β can be calculated by (2), and then get the output of the ELM network.

B. *Improved Gravitational Search Algorithm (IGSA)*

The gravitational search algorithm is a novel heuristic optimization method based on the Newtonian laws of gravitation, which was proposed by Rashedi Esmat, Nezamabadi-pour Hossein and Saryazdi Saeid in [6]. In GSA, the searching agents are considered as objects which attract each other by gravity force, and their performance is measured by their masses. According to the Newtonian gravitational laws, all these agents would move towards the agents with heavier masses caused by the gravity force. The position of the mass corresponds to a solution of the problem, and the heavy masses correspond to optimal solutions.

As proved in [6], assumed there are N agents, the searching strategy on this concept can be described as follows:

$$v_i^d(\mathrm{t+1}) = \mathrm{rand}_i \times v_i^d(\mathrm{t}) + a_i^d(\mathrm{t}), i \in [1, \mathrm{N}] \tag{3}$$

$$x_i^d(\mathrm{t+1}) = x_i^d(\mathrm{t}) + v_i^d(\mathrm{t+1}), i \in [1, \mathrm{N}] \tag{4}$$

Where $x_i^d(\mathrm{t})$, $v_i^d(\mathrm{t})$ and $a_i^d(\mathrm{t})$ represent the position, velocity and acceleration of the *i*th agent in *d*th dimension at *t* iteration, respectively. And $a_i^d(\mathrm{t})$ is calculated by (5):

$$a_i^d(\mathrm{t}) = \frac{F_i^d(\mathrm{t})}{M_i(\mathrm{t})} \tag{5}$$

Where $F_i^d(\mathrm{t})$ is the randomly weighted sum of the forces exerted from other agents and $M_i(\mathrm{t})$ is the mass of the *i*th agent.

While the particle swarm optimization (PSO) is a well-known population-based evolutionary computation technique which was inspired from social behavior of bird flocking, first introduced by Kennedy and Eberhart in [7]. Unlike GSA, the particles in PSO can remember the best solution found so far when they search their own best solutions. As proved by Kennedy and Eberhart in [7], assumed there are N agents, the searching strategy on this concept can be described as follows:

$$v_i^d(\mathrm{t+1}) = w \times v_i^d(\mathrm{t}) + c_1 \times \mathrm{rand}_{1i} \times (\mathrm{pbest}_i^d - \mathrm{x}_i^d(\mathrm{t})) + c_2 \times \mathrm{rand}_{2i} \times (\mathrm{gbest}^d - \mathrm{x}_i^d(\mathrm{t})), i \in [1, \mathrm{N}] \tag{6}$$

$$x_i^d(\mathrm{t+1}) = x_i^d(\mathrm{t}) + v_i^d(\mathrm{t+1}), i \in [1, \mathrm{N}] \tag{7}$$

Where w is the inertia weight, rand_{1i} and rand_{2i} are two random numbers in the range [0,1], c_1 and c_2 are weighting factors, pbest_i^d represent the best solution of the *i*th agent in *d*th dimension at *t* iteration, while gbest^d is the best solution found so far. $x_i^d(\mathrm{t})$ and $v_i^d(\mathrm{t})$ have the same meanings with them in GSA.

The improved gravitational search algorithm (IGSA) is a combination of PSO and GSA, which adds the idea of memory and social information of PSO to improve GSA. Some studies about PSO-GSA have been done, and there are some different ways to combine the two algorithms like [8] and [9]. The method in [8] has been used more often. Unlike other improved methods which one runs after another, PSO and GSA run in parallel in this method. Combining (4,5) and (6,7), the new searching strategy of IGSA is proposed as follows:

$$v_i^d(\mathrm{t+1}) = w \times v_i^d(\mathrm{t}) + c_1 \times \mathrm{rand}_{1i} \times a_i^d(\mathrm{t}) + c_2 \times \mathrm{rand}_{2i} \times (\mathrm{gbest}^d - \mathrm{x}_i^d(\mathrm{t})), i \in [1, \mathrm{N}] \tag{8}$$

$$x_i^d(\mathrm{t+1}) = x_i^d(\mathrm{t}) + v_i^d(\mathrm{t+1}), i \in [1, \mathrm{N}] \tag{9}$$

Where all the parameters have the same meanings with them in PSO or GSA. As shown in (8), the IGSA retains the local search capability of GSA and takes the global best solution (gbest) into consideration.

C. *IGSA-ELM Prediction Neural Network*

Based on the idea of IGSA algorithm and ELM learning algorithm, a novel single hidden feed-forward neural networks learning algorithm called IGSA-ELM is proposed. Some studies on improving ELM have been done, and there are two main different ways that improve ELM by optimizing the number of hidden layer neurons or optimizing the input weights matrix $\mathbf{w}$ and hidden biases vector $\mathbf{b}$, and in this paper the latter method as in [10] has been used. The procedure of the IGSA-ELM algorithm can be described as follows:

Step 1: Randomly initialize the N agents in the search space of proposed algorithm in the range of [-1,1] according to (10):

$$\mathbf{X}_i = [\mathrm{w}_{11i}, \mathrm{w}_{12i}, \dots, \mathrm{w}_{1\tilde{N}i}, \mathrm{w}_{21i}, \mathrm{w}_{22i}, \dots, \mathrm{w}_{2\tilde{N}i}, \dots, \mathrm{w}_{n1i}, \mathrm{w}_{n2i}, \dots, \mathrm{w}_{n\tilde{N}i}, \mathrm{b}_{1i}, \mathrm{b}_{2i}, \dots, \mathrm{b}_{\tilde{N}i}], i \in [1, \mathrm{N}] \quad (10)$$

Where n is the number of input layer neurons and $\tilde{N}$ is the number of hidden layer neurons, and each agent consists of a set of input weights and hidden biases.

Step 2: For each agent, calculate the output weights matrix β by (2), and then evaluate the fitness of each agent in concrete issue.

Step 3: Calculate the acceleration of each agent by (5) and find the best solution so far, and then update the position of each agent by (9) to generate a new swarm.

Step 4: Repeat the search process until reach the maximal iteration number.

Step 5: Output the best solution *gbest* as the optimal set of the input weights and hidden biases and use the optimal ELM to train the model.

III. Proposed Forecasting Model

According to the theoretical knowledge introduced in Section II, the structure of the proposed STLF method is shown in Fig.2, and the detailed steps of the method are as follows:

Step 1: Divide the data into training dataset, validation dataset and testing dataset.

Step 2: Select the input variables and choose the set of input variables which gives the best prediction performance. And the details of the input variables selection are presented in Section 4.2.

Step 3: Normalize the data into [0,1] to lift the limits of the dimension and improve the operation speed.

Step 4: Determine the parameters of the ELM algorithm. As the input weights matrix $\mathbf{w}$ and hidden biases vector $\mathbf{b}$ will be optimized by IGSA, only the number of hidden neurons and the activation function should be chosen.

Step 5: Get the optimal input weights matrix $\mathbf{w}$ and hidden biases vector $\mathbf{b}$ by IGSA and then pass them to ELM.

Step 6: Obtain the best prediction model which gives the best performance on the validation dataset.

Step 7: Deploy the model on the testing dataset to forecast future load and then compare the results.

To evaluate the forecasting performance, an error metric called mean absolute percentage error (MAPE) is used, and MAPE can be calculated by (11):

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{A_i - F_i}{A_i}\right| \times 100\% \quad (11)$$

Where N is the number of data points, while A_i and F_i represent the actual value and the forecast value, respectively.

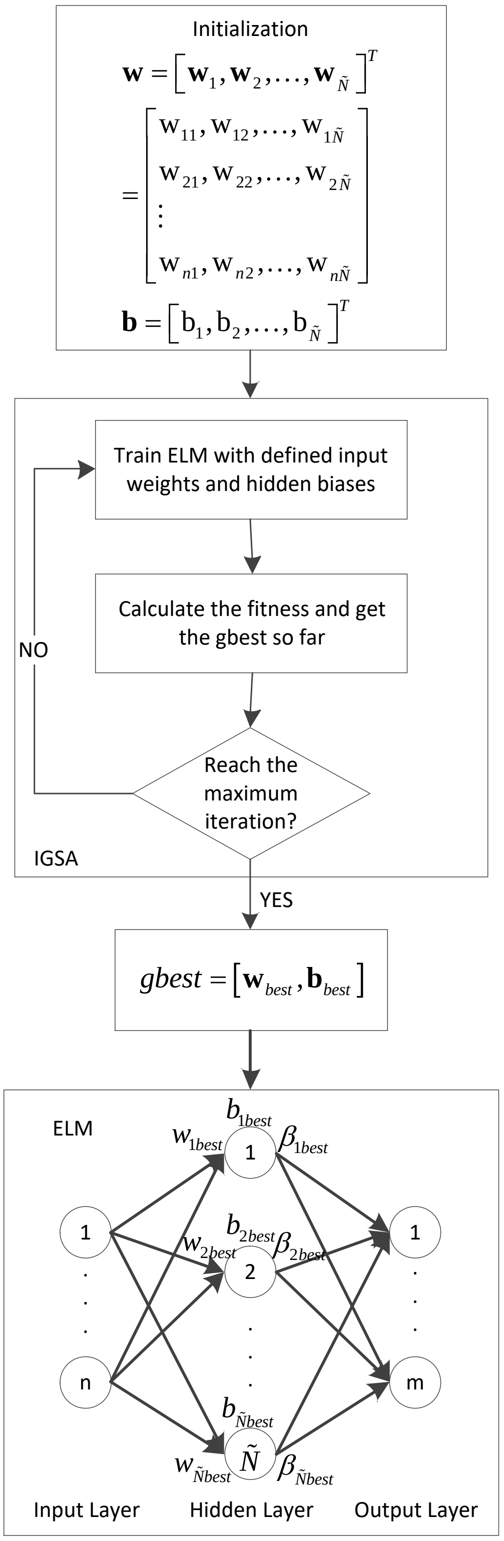
Initialization
$\mathbf{w} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{\tilde{N}}]^T = \begin{bmatrix} \mathrm{w}_{11}, \mathrm{w}_{12}, \dots, \mathrm{w}_{1\tilde{N}} \\ \mathrm{w}_{21}, \mathrm{w}_{22}, \dots, \mathrm{w}_{2\tilde{N}} \\ \vdots \\ \mathrm{w}_{n1}, \mathrm{w}_{n2}, \dots, \mathrm{w}_{n\tilde{N}} \end{bmatrix}$
$\mathbf{b} = [\mathrm{b}_1, \mathrm{b}_2, \dots, \mathrm{b}_{\tilde{N}}]^T$
Train ELM with defined input weights and hidden biases
Calculate the fitness and get the gbest so far
Reach the maximum iteration?
NO
YES
IGSA
$gbest = [\mathbf{w}_{best}, \mathbf{b}_{best}]$
ELM
b_{1best} w_{1best} β_{1best}
b_{2best} w_{2best} β_{2best}
$b_{\tilde{N}best}$ $w_{\tilde{N}best}$ $\beta_{\tilde{N}best}$
1 n 1 2 $\tilde{N}$ 1 m
Input Layer
Hidden Layer
Output Layer

Fig. 2. The structure of IGSA-ELM algorithm

The detailed steps of load forecasting by IGSA-ELM method is described in Fig.3.

IV. Performance

A. Experimental data

The experimental data are from EUNITE competition load forecasting dataset, which is provided by East-Slovakia Power Distribution Company, including load data every 30 min, maximum daily load value, daily mean temperature and holiday information. And the target of the competition is to predict the maximum daily load value of the next month.

B. Input Variables Selection

As one of the most important procedures in machine learning, input variables selection has a great impact on the forecasting results. While different data in different problems are specific, hence analyzing the correlation between different input variables with professional knowledge before input variables selection is very necessary.

In this experiment, the load data in every 30 minute and the daily mean temperature in 1997 and 1998 are given, and the target is to predict the daily maximum load value in January 1999. Based on the dataset, the maximum and the minimum daily load value in 1997 and 1998 can be calculated, as shown in the Fig.4.

Fig.4 shows that the load data are strongly correlated with the seasons. As the target of this experiment is to forecast the load data of January 1999, the training dataset can be divided into winter and summer these two seasons and then choose the winter dataset to train the model.

In order to find more information, the load data in 100 days has been cut out, shown in the Fig.5.

Fig.5 shows that the load data change on a weekly basis. According to the real date of the load data, the day of week information tags are added to the input variables from 1 to 7. For instance, 1 is used for Monday and Sunday is marked by 7.

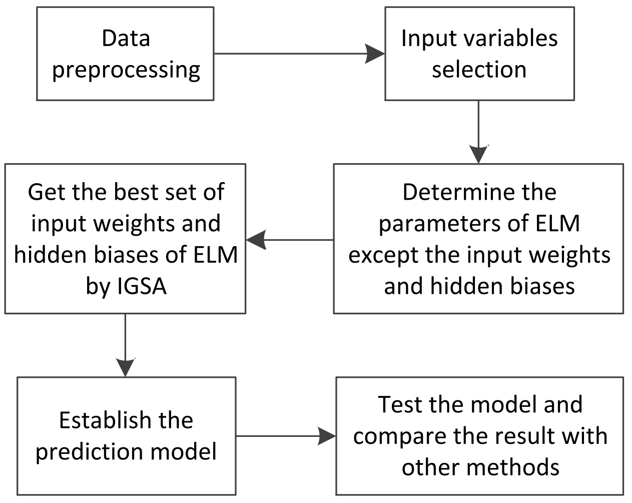
Data preprocessing
Input variables selection
Determine the parameters of ELM except the input weights and hidden biases
Get the best set of input weights and hidden biases of ELM by IGSA
Establish the prediction model
Test the model and compare the result with other methods

Fig. 3. The detailed steps of load forecasting by IGSA-ELM

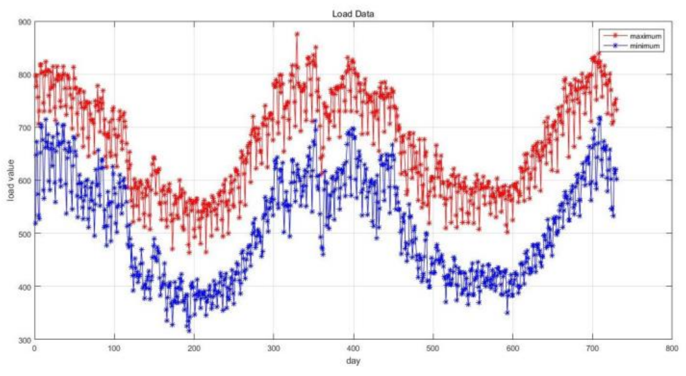

Fig. 4. Maximum and minimum of load data

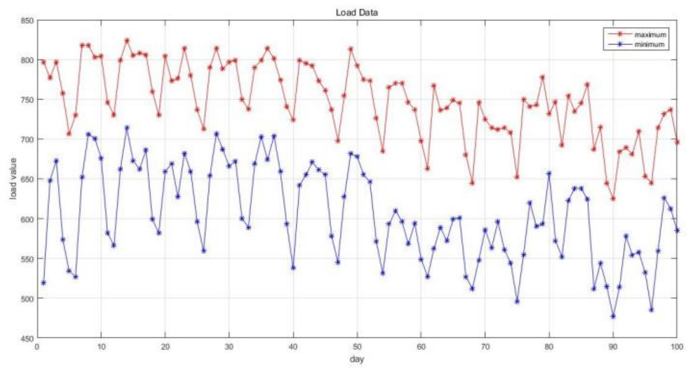

Fig. 5. Maximum and minimum of load data in 100 days

And also, the load data in weekday and weekend are different, so the number 0 and 1 are used to distinguish weekdays and weekends (0 for a weekday and 1 for a weekend).

In addition, the maximum and the minimum load data and the average daily temperature in a week before the day need to be predicted are also chosen as the input variables. But whether to add holiday tags is still to be considered because there are not too many holidays in a year to find the correlation between holidays and load value.

The comparison between the results of experiments with different input variables is presented in Table I.

TABLE I. MAPE Results (%) with Different Input Variables

Select as input variables or not	**Variables**				
	Winter	***Week***	***Weekend***	***Tempera-ture***	***Holiday***
yes	1.75	1.98	2.01	1.96	1.98
no	2.68	3.02	3.03	2.35	1.96

From Table I, it is found that using load data in winter performs better than using all of the data, and adding the week information tags and temperature information into the input variables is a wise choice. But there is no significant difference in whether or not to add holiday tags, so just remove them from input variables to simplify the algorithm and save time.

C. *Comparison of Results*

In these experiments, the parameters of IGSA-ELM are set as follows: the population size $N = 80$, the maximum number of iterations is 100 or 1000, and the inertia weight and weighting factors in (8) are chosen as: w is random number in [0,1], $c_1 = 0.5$ and $c_2 = 1.5$.

In order to reduce the error caused by the randomness of experimental results, each experiment was repeated 20 times and the average test accuracies are presented.

As the comparative experiments of IGSA-ELM method, the prediction models based on BP Neural Networks, GA-BP method, basic ELM algorithm and GSA-ELM method are used. And the iteration number of GA, basic GSA and IGSA are chosen to be 100 or 1000, indicated in the parenthesis. The results are presented in Table II, Table III and Fig.6.

TABLE II. MAPE RESULTS (%) OF DIFFERENT METHODS

Different Methods	**On Validation Dataset**		**On Testing Dataset**	
	Average MAPE	***Best MAPE***	***Average MAPE***	***Best MAPE***
BP	3.56	3.32	3.92[a]	3.63
GA-BP(1000)	2.56	1.87	2.88[a]	2.03
ELM	2.33	1.76	2.38	1.89
GSA-ELM(100)	1.91	1.80	2.20	1.88
GSA-ELM(1000)	1.59	1.48	1.86	1.77
IGSA-ELM(100)	1.52	1.38	2.06	1.52
IGSA-ELM(1000)	1.31	1.14	1.75	1.66

a. *BP and GA-BP usually fall into the local minima and stop learning*

TABLE III. AVERAGE RUNNING TIME (S) OF DIFFERENT METHODS

Different Methods	**Maximum Number of Iterations**			
	0	***100***	***500***	***1000***
BP	41	--[b]	--	--
GA-BP	--[c]	6	27	61
ELM	0.1	--	--	--
GSA-ELM	--	43	220	441
IGSA-ELM	--	42	209	423

b. *The number of iterations in this method must be positive*

c. There is no iteration in this method

IN Table II, the average and the best performance of each method working on both validation dataset and testing dataset are all presented. During the experiments, it is found that not only BP but also GA-BP can easily fall into the local minima and then stop learning, leading to bad prediction results. Overall considerate the results presented in Table II and Table III, it is found that the basic ELM algorithm is the fastest one in all these methods, but due to the randomly generation of the parameters, the performance of basic ELM is very unstable, and the worst value of MAPE has reached 4.56% which is much higher than other methods. In order to improve the accuracy of ELM algorithm, the basic GSA and the IGSA are used. The GSA-ELM and IGSA-ELM have similar running time, which is determined by the maximum number of iterations. However, from the Fig.7, it is found that the convergence speed of IGSA is higher than GSA, which means that IGSA can get the similar performance by smaller number of iterations, so that IGSA can save more time and get better performance when doing the same work.

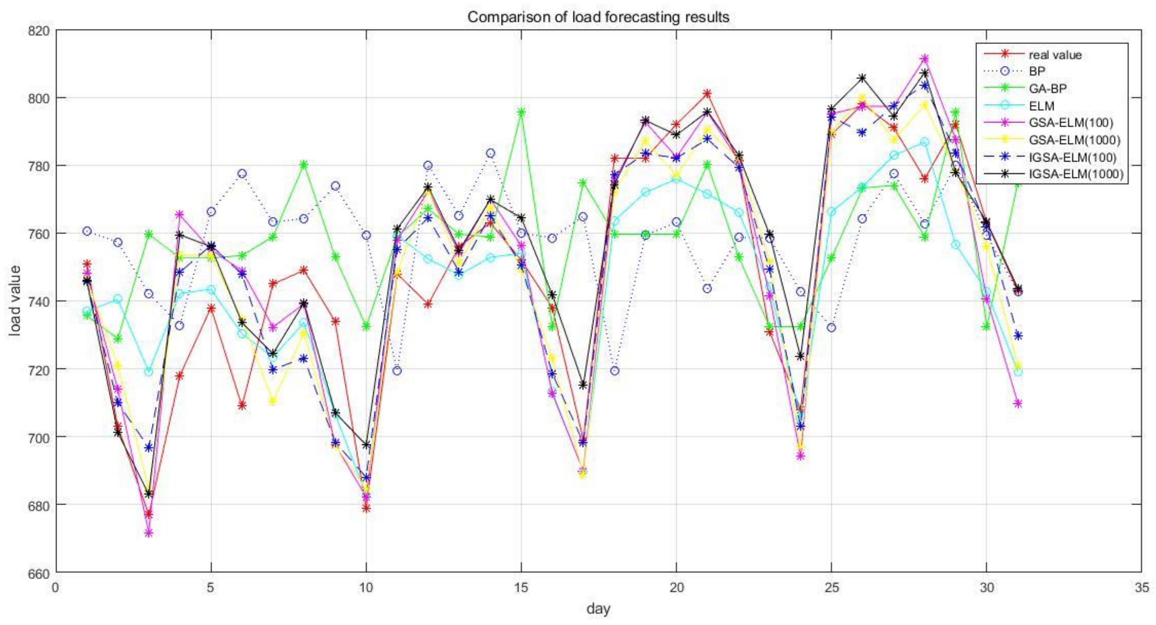

Fig. 6. Comparison of load forecasting results between different methods

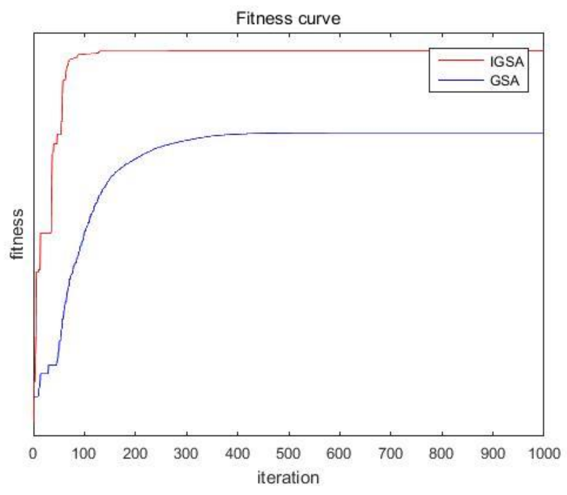

Fig. 7. The fitness curve of GSA-ELM and IGSA-ELM

V. CONCLUSIONS

This paper proposes a novel load forecasting model based on the IGSA-ELM algorithm. The optimal set of input weights and hidden biases of ELM is chosen by IGSA, which is an improved GSA method adding the idea of memory and social information of PSO. Comparative experiments between IGSA-ELM and some conventional methods have been done, and according to the results, the proposed method has an equal or better performance in terms of speed and accuracy.

During the experiments, it is also found that the input variables selection has a great impact on the results. Adding a useful input variable can even reduce the error by half, sometimes more than the error reduction caused by algorithm optimization. Hence, besides the algorithm optimization, the work about input variables selection during establishing a forecasting model also need to be taken seriously, and the future work could be carried out based on this.

REFERENCES

[1] D.M. Zhou, X.H. Guan, J. Sun, Y. Huang. A short-term load forecasting system based on BP artificial neural network[J]. Power System Technology, 2002, 26(2):10-13.

[2] Y.X. Yang. Short term load forecasting using a multilayer neural=network with BP-GA mixed algorithms[J]. Information & Control, 2002.

[3] J.F. Yang, Cheng H Z. Application of SVM to power system short-term load forecast[J]. Electric Power Automation Equipment, 2004.

[4] G.B. Huang, Q.Y. Zhu, C.K. Siew. Extreme learning machine: a new learning scheme of feedforward neural networks[J]. Proc.int.joint Conf.neural Netw, 2004, 2:985-990 vol.2.

[5] G.B. Huang, Q.Y. Zhu, C.K. Siew. Extreme learning machine: Theory and applications[J]. Neurocomputing, 2006, 70(1–3):489-501.

[6] Rashedi Esmat, Nezamabadi-pour Hossein, Saryazdi Saeid (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248

[7] J. Kennedy, R. Eberhart. Particle swarm optimization[C]// IEEE International Conference on Neural Networks, 1995. Proceedings. IEEE, 1995:1942-1948 vol.4.

[8] S. Mirjalili, S.Z.M. Hashim. A new hybrid PSOGSA algorithm for function optimization[C]// International Conference on Computer and Information Application. 2010:374-377.

[9] C. Li, J. Zhou. Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm[J]. Energy Conversion & Management, 2011, 52(1):374-381.

[10] B. Li, Y. Li, X. Rong. A hybrid optimization algorithm for extreme learning machine[J]. Lecture Notes in Electrical Engineering, 2015, 336:297-306.