Research Article

Forecasting Electrical Energy Consumption of Equipment Maintenance Using Neural Network and Particle Swarm Optimization

Xunlin Jiang, Haifeng Ling, Jun Yan, Bo Li, and Zhao Li

College of Field Engineering, PLA University of Science and Technology, Nanjing 210007, China

Correspondence should be addressed to Haifeng Ling; hf.ling@ymail.com

Received 1 June 2013; Revised 17 August 2013; Accepted 9 September 2013

1. Introduction

Equipment maintenance plays a key role in resuming and keeping regular operational capabilities of military equipment [1]. Today’s military equipment becomes more and more complicated and technology intensive, which has brought a heavy burden to equipment maintenance. As a result, the demand of electrical energy is increasing rapidly, and the trend is likely to continue in the future. On the one hand, the proportion of energy cost in the total maintenance cost increases rapidly, which was about 26% during 2006–2008 and 30% in 2009 and 2010, according to our historical data. Thus, an accurate forecasting of electrical energy consumption is very important for decision making in maintenance planning and implementation. On the other hand, the total electrical energy demand in China also has increased dramatically in recent years, which has caused many areas and industries to suffer (typically seasonal) electricity shortage [2–4]. In such cases, if we fail to anticipate the electrical energy consumption of maintenance adequately, prearranged maintenance plans will be interrupted or even abandoned, and a lot of important equipment cannot perform expected operations, which may cause significant economic losses or even catastrophic results. For example, in 2008 there are about 28% engineering machineries waiting for repair had their maintenance programs canceled due to unanticipated electricity shortage. In consequence, in some important disaster rescue operations in China in 2009 (such as the Ms6.0 Chuxiong Earthquake and the Ms6.4 Haixi earthquake), the rescue forces had to request a number of dozers, diggers, loaders, and other machineries from other areas, which reduced rescue efficiency. There are a number of methods and models for forecasting electric energy consumption [5]. Conventional forecasting approaches, such as principal component analysis, least error square (LES), and regression techniques [5–8], become difficult and impractical to provide an adequate forecasting of electrical energy consumption, mainly due to the ever increasing complexity and diversity of equipment maintenance demands. Moreover, those approaches have some inherent limitation and disadvantages such as requirement for large training datasets, sensitivity to noise, and low capability of handling missing data. In recent years, methods and models based on artificial intelligence (AI), including artificial neural networks (ANNs) [9], fuzzy logic and expert systems [10], have received much interest and gained popularity in forecasting electric energy consumption.
Inspired by biological nervous systems, an ANN consists of an interconnection of a number of neurons. It uses processing elements connected by links of variable weights to form a black box representation of systems [11]. One of the most widely used ANNs is the multilayer feed-forward neural network, which consists of an input layer, one or more hidden layers, and an output layer. Figure 1 presents an example of such a three-layer neural network.

One of the most useful and popular applications of ANNs is forecasting, the process of which can be divided into two steps: a learning step for constructing the network model based on training data and a forecasting step for predicting targets values for real data. The learning step is performed by iteratively processing a dataset of training tuples, comparing the network's output value for each tuple with the actual known target value, according to which adjusting the weights of the network so as to minimize the mean-squared error between the network's output value and the actual target value.

The most well-known neural network learning algorithm is backpropagation (BP) [12], which calculates the gradient of the error with respect to the weights for a given input by propagating error backwards through the network. However, BP involves long training times and its performance degrades rapidly when the problem complexity increases [13]. In the recent two decades, many intelligent heuristic algorithms, such as genetic algorithms (GAs) and particle swarm optimization (PSO) algorithms, have been applied to the problem of training different kinds of ANNs [14, 15], including the multilayer feed-forward neural network, and have demonstrated their efficiency and effectiveness in training ANNs for improving their forecasting accuracies [16, 17].

In this paper, we propose an approach of forecasting electrical energy consumption of equipment maintenance using ANN and PSO. We use a feed-forward neural network for modeling possible relationships between the input variables and the expected electrical energy consumption and develop a new adaptive PSO for training the network. Experimental results show that our approach is superior to some state-of-the-art methods.

The rest of the review paper is synthesized as follows: Section 2 reviews the related work of the application of ANNs in electrical energy consumption forecasting and the related problems, Section 3 presents our approach in detail, Section 4 presents the computational experiments and the real-world applications, and Section 5 concludes.

2. Related Work

ANNs have gained much popularity in forecasting electric loads and energy demands since 1990s. Kawashima [18] developed an ANN model for building energy prediction, which is trained by the BP algorithm. To improve performance, the algorithm is combined with a three-phase annealing method which gradually reduces the learning rate in order to improve accuracy in a relatively short time. In [19] Islam et al. proposed an ANN-based weather-load and weather-energy models, where a set of weather and other variables are identified for both models together with their correlations and contribution to the forecasted variables. The models were applied to the historical energy, load, and weather data available for the Muscat power system from 1986 to 1990, and the forecast results for 1991-1992 showed that monthly electric energy and load can be predicted within a maximum error of 6% and 10%. In [20] Al-Shehri employed an ANN for forecasting the residential electrical energy in the Eastern Province of Saudi Arabia, and the forecasting result was shown to be closer to the real data than that predicted by the polynomial fit model.

Hatziargyriou and Dialynas [21] proposed an adaptive ANN for midterm energy forecasting. Their model transforms the input variables to differences or relative differences, in order to predict energy values not included in the training set. The ANN parameters, such as the finally used input variables, the number of neurons, initial values, and time periods of momentum term and training rate, are simultaneously selected by an optimization process. Sözen et al. [22] also forecasted net energy consumption in Turkey based on an ANN approach, which employs a logistic sigmoid transfer function. They showed that the statistical coefficients of multiple determinations in the experiments are equal to 0.9999 and 1 for training and test data, respectively. In [23] Ermis et al. presented an ANN model which is trained based on world energy consumption data from 1965 to 2004 and applied for forecasting world green energy consumption to the year 2050. It is estimated that world green energy and natural gas consumption will continue increasing after 2050, while world oil and coal consumption are expected to remain relatively stable after 2025 and 2045, respectively.

The work of Azadeh et al. [24] presented an integrated algorithm for forecasting monthly electrical energy consumption based on ANN, computer simulation, and design of experiments using stochastic procedures. The integrated algorithm was applied for forecasting monthly electrical energy consumption in Iran from 1994 to 2005, the result of which demonstrated the effectiveness of the algorithm. Yokoyama et al. [25] proposed a global optimization method called "Modal Trimming Method" for determining ANN parameters, where the trend and periodic change are first removed from time series data on energy demand, and the converted data is used as the main input to a neural network.
Their approach was validated by the application to predict the cooling demand in a building used for a benchmark test. Geem and Roper [26] proposed an ANN model with feed-forward multilayer perceptron, error back-propagation algorithm, momentum process, and scaled data and applied the model to estimate the energy demand for the Republic of Korea. Nevertheless, to our best knowledge, research on the application of ANN models together with intelligent training algorithms to the field of energy consumption forecasting is still rare.

3. The Proposed Method

3.1. Network Structure. Our method adopts a three-layer forward neural network, since an ANN can solve a problem by using one hidden layer, provided it has the proper number of neurons [27]. From the original database of equipment maintenance, the input variables are preliminary selected by a team of experts in the domains of equipment maintenance and electrical power, and a dataset of twenty-three fields is filtered out. In general, the input variables can be divided into two categories. The first category is related to the equipment to be repaired. The equipment is classified into two categories. The first category is related to the heavy equipment, medium equipment, and light equipment. The equipment class (1-heavy, 2-medium, 3-light) and the second denotes the maintenance level (1-major, 2-medium, 3-minor). The first category also contains two variables related to rated power and three variables related to maintenance time of all equipment. The second category consists of nine variables related to maintenance resources and environments. Table 1 lists the input variables of our ANN for electrical energy consumption forecasting.

Preliminary normalization by the maximum and minimum values of each variable \( x \) is executed based on the following equation:

\[
x = c + \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} (b - c),
\]

where \( x_{\text{max}} \) and \( x_{\text{min}} \) are the upper and lower values of the original variable, respectively, and \( b \) and \( c \) are the corresponding values of the normalized variable.

Given a unit \( j \) in a hidden or output layer, the net input to unit \( j \), denoted by \( I_j \), is defined as

\[
I_j = \sum_i w_{ij}O_i + \sigma_j,
\]

where \( w_{ij} \) is the weight of the connection from unit \( i \) in the previous layer to unit \( j \), \( O_i \) is output of unit \( i \), and \( \sigma_j \) is acting as a threshold bias of the unit \( j \).

And we employ the widely used sigmoid transfer function to compute the output of unit \( j \) from its net input:

\[
O_j = \frac{1}{1 + \exp(-I_j)}.
\]

3.2. PSO and Control Parameters. PSO is a stochastic, population-based optimization technique that enables a number of individual solutions, called particles, to move through a hyperdimensional search space to search for the optimum or optima. Each particle has a position vector \( \mathbf{x} \) and a velocity vector \( \mathbf{v} \), which are adjusted at iterations by learning from a personal best solution \( \mathbf{pbest} \), found by the particle \( i \) itself and a global best solution \( \mathbf{gbest} \) found by the whole swarm so far at each dimension \( j \):

\[
\mathbf{v}_{ij}^{(t+1)} = \omega \mathbf{v}_{ij}^{(t)} + c_1 r_1 (\mathbf{pbest}_{ij}^{(t)} - \mathbf{x}_{ij}^{(t)}) + c_2 r_2 (\mathbf{gbest}^{(t)} - \mathbf{x}_{ij}^{(t)}),
\]

\[
\mathbf{x}_{ij}^{(t+1)} = \mathbf{x}_{ij}^{(t)} + \mathbf{v}_{ij}^{(t+1)},
\]

where \( \omega \) is the inertia weight for controlling the velocity, \( c_1 \) and \( c_2 \) are the acceleration coefficients reflecting the weighting of “cognitive” and “social” learning, respectively, and \( r_1 \) and \( r_2 \) are two random numbers ranging from 0 to 1. Empirical studies have shown that PSO has a high efficiency in convergence to desirable optima and performs better than GAs in many problems [28–34].

Most PSO algorithms use a linear decreasing inertia weight and constant acceleration coefficients in (5). In recent years, some work has been conducted on the nonlinear and nonmonotonic change of control parameters, which has shown the ability of increasing the adaptivity and diversity of the swarm. In the work of Yang et al. [35], the inertia weight is adapted based on two characteristic parameters in the search course of PSO, namely, speed factor and aggregation degree factor. Arumugam and Rao [36] used the ratio of the global best fitness and the average of particles’ personal best fitness to determine the inertia weight in each iteration. Panigrahi
(1) Randomly initialize a main swarm of \( n \) particles, each of which representing the number of the hidden nodes and the connection density of an ANN;

(2) Let \( t = 0 \); set the \( \text{pbest} \) of each particle to its original position;

(3) For \( t = 1 \) to \( t_{\text{max}} \) do

(4) For \( i = 1 \) to \( n \) do

(5) Use the sub PSO procedure to evolve the \( i \)th ANN on the training set;

(6) Move the \( i \)th solution in the main search space according to (4) and (5);

(7) Update the \( \text{pbest} \) of the \( i \)th solution;

(8) Update the \( \text{gbest} \) of the main-swarm;

(9) Update the control parameters of the outer loop according to (6)- (9);

(10) Use the sub PSO procedure to evolve the \( \text{gbest} \) of the main-swarm on the training set together with the validation set;

(11) Return the \( \text{gbest} \) of the main-swarm;

**Algorithm 1:** The pseudocode of the main PSO algorithm for evolving the ANN.

et al. [37] proposed another PSO method where different particles are assigned with different inertia weights based on the ranks of the particles. In [38] Nickabadi et al. proposed a new adaptive inertia weight approach that uses the success rate of the swarm as its feedback parameter to ascertain the particles’ situation in the search space.

Zhan et al. [30] proposed an adaptive PSO that differentiates the evolutionary states as exploration, exploitation, convergence, and jumping out by evaluating the population distribution and particle fitness and uses the state information to automatically control inertia weight and acceleration coefficients at run time. Ratnaweera et al. [39] introduced a time-varying acceleration coefficient strategy where the cognitive coefficient is linearly decreased and the social coefficient is linearly increased. In [40] Zheng et al. introduced a new adaptive parameter strategy to the comprehensive learning particle swarm optimizer, which uses evolutionary information of individual particles to dynamically adapt the inertia weight and acceleration coefficient at each iteration.

### 3.3. An Adaptive PSO Algorithm for Evolving the ANN

Since the search process of PSO is highly complicated in the ANN training problems, here we use a self-adaptive mechanism for both the inertia weight and the acceleration coefficients. The mechanism is based on the concept of the success of a particle at an iteration, that is, whether or not a particle \( i \) improves its personal best at iteration \( t \):

\[
s_i^{(t)} = \begin{cases} 
1, & \text{if } f \left( \text{pbest}_i^{(t+1)} \right) < f \left( \text{pbest}_i^{(t)} \right) \\
0, & \text{else}
\end{cases}
\]  

(6)

Based on that we compute the rate of success of the whole swarm \( S \):

\[
\rho^{(t)} = \frac{\sum s_i^{(t)}}{|S|}.
\]

(7)

That is, \( \rho \in [0, 1] \) is the percentage of the particles which have had an improvement in the last iteration. A high value of \( \rho \) indicates a high probability that the particles have converged to a nonoptimum point or slowly moving toward the optimum, while a low value of \( \rho \) indicates that the particles are oscillating around the optimum without much improvement [38]. And we use the following equation for dynamically changing inertia weight of PSO:

\[
w(t) = w^{(0)} - 1.
\]

(8)

PSO is a population-based optimization method, which is preferred to encourage the exploration during the early stages and to enhance exploitation during the latter stages. Here we also employ a mechanism that linearly decreases the cognitive coefficient and increases the social coefficient as follows:

\[
\begin{align*}
\alpha_1^{(t)} &= \alpha_1^{\text{max}} - \frac{t}{t_{\text{max}}} \left( \alpha_1^{\text{max}} - \alpha_1^{\text{min}} \right), \\
\alpha_2^{(t)} &= \alpha_2^{\text{min}} + \frac{t}{t_{\text{max}}} \left( \alpha_2^{\text{max}} - \alpha_2^{\text{min}} \right).
\end{align*}
\]

(9)

The dataset of our forecasting problem is partitioned into three sets: a training set, a validation set, and a testing set. Our adaptive PSO contains a main procedure for evolving a swarm of ANNs with different structure (different number of the hidden nodes and the connection density), which is outlined in Algorithm 1, and a subprocedure for evolving a swarm of ANNs with the same structure but different connection weights, which is outlined in Algorithm 2. For each particle that represents an ANN, its fitness is defined as the root mean-squared error (RMSE) of the training dataset, which is calculated as follows:

\[
\text{RMSE} = \sqrt{\frac{\sum_{k=1}^{n} (y_k - y_k^*)^2}{N}}.
\]

(10)

where \( y_k \) is the network output of the \( k \)th training tuple and \( y_k^* \) is its desired output. The fitness of particles in the main-swarm is evaluated as the best ANN obtained by its subswarms.

The flowchart of the algorithm is shown in Figure 2.

### 4. Computational Experiment

The proposed method was applied to the actual historical data collected from seven equipment maintenance organizations.
(1) Randomly initialize a sub-swarm of \( n \) particles, each of which representing the weights of the \( i \)th ANN;
(2) Let \( t' = 0 \); set the \( \text{pbest} \) of each particle to its original position;
(3) For \( t' = 1 \) to \( t'_{\text{max}} \) do
(4) For each particle in the \( i \)th sub-swarm do
(5) Move the particle in the sub search space according to (4) and (5);
(6) Update the \( \text{pbest} \) of the particle;
(7) Update the \( \text{gbest} \) of the \( i \)th sub-swarm;
(8) Update the control parameters of the inner loop according to (6)–(9);
(9) return the \( \text{gbest} \) of the \( i \)th sub-swarm as the \( i \)th ANN;

**Algorithm 2: The pseudocode of the sub-PSO procedure.**

> Initialize a main-swarm, each particle representing an ANN structure
> Moving particles in the main search space based on the motion equations
> Update the \( \text{gbest} \) and the \( \text{pbest} \) of the particles
> Update the control parameters of the main-swarm
> For each particle in the main-swarm
> Initialize a sub-swarm, each particle representing the weights of the ANN
> Moving particles in the sub search space based on the motion equations
> Update the \( \text{gbest} \) and the \( \text{pbest} \) of the particles
> Update the control parameters of the sub-swarm
> Terminate?

**Figure 2: The flowchart of the proposed approach.**

from 1999 to 2011. We used the data from 1999 to 2006 and the data of 2008 as the training dataset, the data of 2007 and 2010 as the validation dataset, and the data of 2009 and 2011 as the testing dataset. Table 2 presents a summarization of the datasets.

We, respectively, implement three other approaches: the ANN trained by BP (denoted by ANN), the ANN trained by basic PSO (denoted by ANN-PSO), and the nonlinear regression model optimized by PSO (denoted by R-PSO) [41], in order to comparatively evaluate the performance of our method (denoted by ANN-APSO).

The experiments are conducted on a computer with Intel i5 3450 (4 \( \times \) 3.1 G) multicore processor and DDR3 8 GB memory. The basic parameters of our approach are set as \( c_1_{\text{min}} = c_2_{\text{min}} = 0.5, c_1_{\text{max}} = c_2_{\text{max}} = 2.0, t_{\text{max}} = 1000, t'_{\text{max}} = 80, n = 40, \) and \( n' = 100 \). The maximum running time of the other approaches is set as they consume nearly the same CPU time as our method.

Tables 3 and 4, respectively, present the experimental results of the four approaches on the testing datasets of 2009 and 2011. As we can see from the results, on every test data tuple, our ANN trained by adaptive PSO achieves the
Table 2: The summarization of the historical datasets on electrical energy consumption of equipment maintenance, where \( N_E \) is the number of equipment repaired and \( E_C \) is the total electrical energy consumption (in thousand kWh).

<table>
<thead>
<tr>
<th>Year</th>
<th>Tuples</th>
<th>Org1</th>
<th>Org2</th>
<th>Org3</th>
<th>Org4</th>
<th>Org5</th>
<th>Org6</th>
<th>Org7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>12</td>
<td>48</td>
<td>39.2</td>
<td>65</td>
<td>50.5</td>
<td>52</td>
<td>39.9</td>
<td>37</td>
</tr>
<tr>
<td>2000</td>
<td>15</td>
<td>50</td>
<td>43.6</td>
<td>60</td>
<td>49.3</td>
<td>55</td>
<td>48.7</td>
<td>58</td>
</tr>
<tr>
<td>2001</td>
<td>15</td>
<td>54</td>
<td>58.2</td>
<td>64</td>
<td>60.3</td>
<td>56</td>
<td>57.1</td>
<td>56</td>
</tr>
<tr>
<td>2002</td>
<td>17</td>
<td>57</td>
<td>66.7</td>
<td>63</td>
<td>68.2</td>
<td>55</td>
<td>62.2</td>
<td>58</td>
</tr>
<tr>
<td>2003</td>
<td>18</td>
<td>61</td>
<td>72.8</td>
<td>66</td>
<td>81.6</td>
<td>59</td>
<td>68.9</td>
<td>60</td>
</tr>
<tr>
<td>2004</td>
<td>21</td>
<td>66</td>
<td>91.0</td>
<td>68</td>
<td>88.3</td>
<td>58</td>
<td>70.3</td>
<td>61</td>
</tr>
<tr>
<td>2005</td>
<td>20</td>
<td>72</td>
<td>113</td>
<td>68</td>
<td>95.8</td>
<td>62</td>
<td>81.7</td>
<td>65</td>
</tr>
<tr>
<td>2006</td>
<td>21</td>
<td>74</td>
<td>118</td>
<td>72</td>
<td>106</td>
<td>69</td>
<td>105</td>
<td>68</td>
</tr>
<tr>
<td>2007</td>
<td>21</td>
<td>75</td>
<td>123</td>
<td>78</td>
<td>137</td>
<td>65</td>
<td>116</td>
<td>73</td>
</tr>
<tr>
<td>2008</td>
<td>18</td>
<td>89</td>
<td>161</td>
<td>78</td>
<td>145</td>
<td>73</td>
<td>145</td>
<td>75</td>
</tr>
<tr>
<td>2009</td>
<td>21</td>
<td>72</td>
<td>132</td>
<td>79</td>
<td>151</td>
<td>72</td>
<td>138</td>
<td>78</td>
</tr>
<tr>
<td>2010</td>
<td>21</td>
<td>80</td>
<td>158</td>
<td>82</td>
<td>159</td>
<td>70</td>
<td>139</td>
<td>76</td>
</tr>
<tr>
<td>2011</td>
<td>21</td>
<td>83</td>
<td>169</td>
<td>80</td>
<td>160</td>
<td>72</td>
<td>145</td>
<td>75</td>
</tr>
</tbody>
</table>

Table 3: The MAPE (%) associated with the four approaches on the testing dataset of 2009.

<table>
<thead>
<tr>
<th>Organization</th>
<th>ANN</th>
<th>ANN-PSO</th>
<th>R-PSO</th>
<th>ANN-APSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>1.816(^*)</td>
<td>1.220(^*)</td>
<td>1.093(^*)</td>
<td>0.835</td>
</tr>
<tr>
<td>#2</td>
<td>1.493(^*)</td>
<td>0.000</td>
<td>0.355(^*)</td>
<td>0.000</td>
</tr>
<tr>
<td>#3</td>
<td>5.267(^*)</td>
<td>1.801(^*)</td>
<td>1.103</td>
<td>1.044</td>
</tr>
<tr>
<td>#4</td>
<td>2.205(^*)</td>
<td>0.934(^*)</td>
<td>0.783(^*)</td>
<td>0.297</td>
</tr>
<tr>
<td>#5</td>
<td>1.315(^*)</td>
<td>0.303(^*)</td>
<td>0.526(^*)</td>
<td>0.067</td>
</tr>
<tr>
<td>#6</td>
<td>3.655(^*)</td>
<td>1.500(^*)</td>
<td>1.218(^*)</td>
<td>0.722</td>
</tr>
<tr>
<td>#7</td>
<td>0.452(^*)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

In column 2–4, \(^*\) means that the result of ANN-APSO has statistically significant improvement over that of the comparative algorithm (using paired one-tailed \(t\)-test at a 0.05 level of significance).

Table 4: The MAPE (%) associated with the four approaches on the testing dataset of 2011.

<table>
<thead>
<tr>
<th>Organization</th>
<th>ANN</th>
<th>ANN-PSO</th>
<th>R-PSO</th>
<th>ANN-APSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>3.357(^*)</td>
<td>1.472(^*)</td>
<td>1.164(^*)</td>
<td>0.767</td>
</tr>
<tr>
<td>#2</td>
<td>6.060(^*)</td>
<td>1.833(^*)</td>
<td>2.261(^*)</td>
<td>0.935</td>
</tr>
<tr>
<td>#3</td>
<td>8.248(^*)</td>
<td>2.707(^*)</td>
<td>2.095(^*)</td>
<td>1.363</td>
</tr>
<tr>
<td>#4</td>
<td>5.131(^*)</td>
<td>0.896(^*)</td>
<td>0.803(^*)</td>
<td>0.520</td>
</tr>
<tr>
<td>#5</td>
<td>3.690(^*)</td>
<td>0.422(^*)</td>
<td>0.693(^*)</td>
<td>0.193</td>
</tr>
<tr>
<td>#6</td>
<td>2.851(^*)</td>
<td>1.658(^*)</td>
<td>1.827(^*)</td>
<td>0.079</td>
</tr>
<tr>
<td>#7</td>
<td>0.941(^*)</td>
<td>0.081(^*)</td>
<td>0.235(^*)</td>
<td>0.000</td>
</tr>
</tbody>
</table>

In column 2–4, \(^*\) means that the result of ANN-APSO has statistically significant improvement over that of the comparative algorithm (using paired one-tailed \(t\)-test at a 0.05 level of significance).

minimum mean absolute percentage error (MAPE) among the four approaches. On the testing dataset of 2009, nearly all of the MAPEs of ANN-APSO are less than one percent. Roughly speaking, the testing dataset of 2011 is much more difficult than that of 2009, and thus the MAPEs of the other three approaches generally increase to a certain extent (except for the ANN-PSO on organization #4), but the MAPEs of ANN-APSO decrease on organizations #1 and #6, remain zero on #7, and increase very slightly on the other four cases. Statistical tests also show the following.

(i) On the dataset of 2009, our ANN-APSO has significant performance improvement over ANN in all cases and over ANN-PSO and R-PSO in most cases.

(ii) On the dataset of 2011, the ANN-APSO always has significant performance improvement over the other three methods.

In summary, ANN-APSO provides the best forecasting accuracy on the testing datasets, and it is much more reliable and robust than the other three approaches.
5. Conclusion

The paper presents an approach for forecasting electrical energy consumption of equipment maintenance based on ANN trained by PSO. The ANN model is used for modeling possible relationships between the input variables and the expected electrical energy consumption, and a new PSO algorithm with adaptive control parameters is developed for evolving the ANN. Experimental results demonstrate that our approach performs better than some state-of-the-art methods on the historical data of equipment maintenance in China. And we believe that the proposed method can be applied to a wider range of forecasting problems in engineering.

In practice, we find that it is often difficult to get exact values of a certain number of input variables, and currently we just use their expected or estimated values. To further improve the adaptivity and flexibility of our approach, we are testing the use of fuzzy numbers and fuzzy-neural networks. For testing the use of fuzzy numbers and fuzzy-neural networks to improve the adaptivity and flexibility of our approach, we are currently just using their expected or estimated values. To further improve the adaptivity and flexibility of our approach, we are testing the use of fuzzy numbers and fuzzy-neural networks [42–46] in forecasting and adapting our ANN-APSO algorithm for training and evolving the fuzzy-neural networks. Our future work also includes improving the algorithm by information sharing between different subswarms.

Acknowledgments

This work was supported by the Natural Science Foundation of China under Grant nos. 61020106009 and 61105073. The authors are grateful to the reviewers’ valuable comments that help to improve the paper greatly.

References


