Knowl Inf Syst (2009) 21:249–266
DOI 10.1007/s10115-009-0242-y

REGULAR PAPER

Parameter determination and feature selection for back-propagation network by particle swarm optimization

Shih-Wei Lin · Shih-Chieh Chen · Wen-Jie Wu · Chih-Hsien Chen

Received: 19 June 2008 / Revised: 12 May 2009 / Accepted: 27 June 2009 /
Published online: 13 August 2009

Abstract The back-propagation network (BPN) is a popular tool with applications in a variety of fields. Nevertheless, different problems may require different parameter settings for a given network architecture. A dataset may contain many features, but not all features are beneficial for classification by the BPN. Therefore, a particle-swarm-optimization-based approach, denoted as PSOBPN, is proposed to obtain the suitable parameter settings for BPN and to select the beneficial subset of features which result in a better classification accuracy rate. A set of 23 problems with a range of examples and features drawn from the UCI (University of California, Irvine) machine learning repository is adopted to test the performance of the proposed algorithm. The results are compared with several well-known published algorithms. The comparative study shows that the proposed approach improves the classification accuracy rate in most test problems. Furthermore, when the feature selection is taken into consideration, the classification accuracy rates of most datasets are increased. The proposed algorithm should thus be useful to both practitioners and researchers.

Keywords Back-propagation network · Particle swarm optimization · Feature selection · Parameter determination

S.-W. Lin (✉) · W.-J. Wu
Department of Information Management, Chang Gung University, 259 Wen-Hwa 1st Road,
Kwei-Shan, Tao-Yuan 333, Taiwan, ROC
e-mail: swlin@mail.cgu.edu.tw

S.-C. Chen
Department of Industrial Engineering, National Taiwan University of Science and Technology,
No. 43, Sec. 4, Keelung Rd., Taipei 106, Taiwan, ROC

C.-H. Chen
Department of Management, Fo-Guang University, No.160, Linwei Rd.,
Jiaosi Shiang, Yilan 26247, Taiwan, ROC

1 Introduction

Neural network is a well-known machine learning algorithm among data mining techniques, and can be used in a variety of applications, such as modeling of electrostatic fluidized bed coating processes [3], text classification [8], digital signal-type identification [11], industry automotive [16], condition monitoring of a pneumatic process valve actuator [24], database integration [34], defective classification [39], consumer loan evaluation [43], location determination of mobile devices [61], and diagnosis of heart disease [67].

Back-propagation network (BPN) is one of the most popular neural network, which is a viable, reliable, and attractive approach for data processing because (1) BPNs are capable of modeling non-linear processes; (2) the data-driven features of BPNs make them powerful in parallel computing and capable of handling large amounts of data, and (3) BPNs have good fault tolerance and adaptability [4].

Before applying a BPN to problem solving, the parameter settings of the BPN, including the hidden layer, learning rate, momentum term, number of hidden neurons, and learning cycle must be determined. The parameter settings for network architectures must be determined carefully in order to avoid constructing a suboptimal network model that may significantly increase computational costs and produce inferior results.

In most pattern classification problems, given a large set of potential features, it is usually necessary to find a small subset with which to classify. Data lacking any feature selection may be redundant or noisy and may reduce the efficiency of classification. The main benefits of feature selection are as follows: (1) reducing computational cost and storage requirements, (2) dealing with the degradation of classification efficiency due to the finite size of training sample sets, (3) reducing training and prediction time, and (4) facilitating data understanding and visualization [1].

In feature selection, because each feature is necessary to determine whether it is useful or not, the task of finding the optimal subset of features is inherently combinatory. Therefore, feature selection is an optimization problem. An optimal approach is needed to evaluate all possible subsets. This research proposes a particle swarm optimization (PSO) based approach, denoted as PSOBPN, to obtain the appropriate parameter settings for a BPN, and to select the beneficial subset of features which result in better classification accuracy rate.

The remainder of this paper is organized as follows. Section 2 reviews the studies on the back-propagation network, feature selection, and particle swarm optimization. Section 3 elaborates on the proposed PSOBPN approach to determine the appropriate parameter settings for the BPN, and determines the beneficial subset of features. Section 4 describes the experiment results. Conclusions and future research are offered in the last section.

2 Literature review

2.1 BPN

The BPN is a common neural network model whose architecture consists of multilayer perceptrons (MLP). The BPN uses the idea of "the gradient steepest descent method" to minimize the errors between actual and predictive output functions. An increased number of hidden layers and the transformation function of the smoothing differential can allow the network to apply the gradient steepest descent method to correct the network weights formula. Therefore, if there are enough hidden layers or hidden neurons, the linear threshold curve can approach any function [17].

The learning procedures of a BPN include initialization, forward, and reverse processes. The net input and output of each neuron in the hidden and output layers are computed. Initially, the training data are fed to the input layer of the network. To compute the net input of the neuron, each input connected to the neuron is multiplied by its corresponding weight, and then summed. The error is propagated backwards by updating the weights and biases to reflect the error of the network's prediction. The weights and biases are updated to reflect the propagated errors.

Since the parameter settings for a BPN are often designed quite differently due to the unique characteristics of the data, trial-and-error seems to be the most common way to identify the optimum value of learning rate, momentum, hidden neurons, and learning cycle. The following are problems generally faced when using a BPN:

(1) Learning rate and momentum term: Too high a learning rate will make it hard for it to converge; too low a learning rate will cause slow convergence and may fall into local optimization. Too small momentum term cannot increase the classification accuracy rate; too big momentum term cause extreme modification.

(2) Number of hidden neurons: When there are too few hidden neurons, a larger error is likely to occur. Increasing the number of hidden neurons can alleviate this situation but will simultaneously affect the speeds of convergence and computing cost.

(3) Learning cycle: In the study of Schittenkopf et al. [50] back-propagation on the illustrated data set caused over-fitting which started after a certain number of training cycles. Too much the learning cycle may cause the over-fitting. On the contrary, too low a learning cycle can lead to under-fitting.

Therefore, rule of thumb or "trial and error" methods are used to determine the parameter settings for network architectures. However, it is difficult to obtain the optimal parameter settings for network architectures. If parameter values are not set appropriately, that may lead to over-fitting or under-fitting problem [13].

Lin and Ting [35] and Wang et al. [65] have tested a few different network architectures to find the best network architecture. However, there is still a need for a systematic method of determining the appropriate network architecture for building the model. Common issues with the BPN-based modeling approach include the under-training problem, convergence problem, over-fitting problem, and topology optimization problem [9]. While the first two can be mitigated by careful selection of the stopping criteria, the latter two are of interest here through an optimization approach, which optimizes the network topology as well as reduces the risk of over-fitting.

Researchers have used three types of approaches to handle the problems discussed above.

(1) Search for the appropriate parameter settings of the BPN:
This method seeks to combine the other search approaches with the gradient steepest descent method in order to rapidly determine excellent parameter settings for the network architectures of the BPN. In most cases, one hidden layer is sufficient to compute the arbitrary decision boundaries for the outputs [22]. The number of hidden neurons required depends on the case. Khaw et al. [26] proposed using the Taguchi method to adjust the number of hidden layers, number of hidden neurons, and the learning rate. They attempted to find optimal network architectures faster while maintaining good classification accuracy rates at the same time. Castillo et al. [6] used the genetic algorithm (GA) to adjust the number of hidden neurons and initial weight. They constructed network architectures through fixed learning cycles and adequate learning rates. In addition to the utilization of GA to adjust the number of hidden neurons and initial weight,

Castillo et al. [5] further calculated the suitable learning rate. Wang and Huang [64] used the GA-based approach to tune BPN parameters and applied them to chaotic time series problems.

(2) Search for the optimal weights after training:
This approach fixes the network architecture of the BPN, such as the number of hidden neurons and the number of hidden layers, to find optimal weights. Unlike the back-propagation learning process of a BPN, this approach directly adjusts the weights and searches for a training-completed model. The purpose of this approach is to use prediction methods that are able to search for local optimization to find these weights because the concept of the gradient steepest descent method employed by the BPN falls relatively easily into local optimization. Sexton et al. [53] used the Tabu search method, and Gupta and Sexton [15] employed genetic algorithm (GA) to find the optimal weights. Ghosh and Verma [14] used an evolutionary least square based learning method to find the weights and a special mechanism to adjust the number of hidden neurons to achieve optimization. Huang et al. [22] used genetic algorithm to optimize BPN weight matrices and evaluated its performance in six test functions.

(3) Neural network pruning:
This approach assumes that there are too many connections in the MLP architecture. A new, smaller neural network architecture results from pruning some connections. The pruned neural network architecture undergoes renewed training, but the classification accuracy rate remains the same (or even better). Sexton et al. [54] proposed an SA-based approach to determine the network architecture of the BPN. Yeung and Zeng [69] used sensitivity measurement, the main idea of which is to constantly train the network according to a certain performance criterion and then remove the neurons with the lowest relevant values. Yamasaki et al. [66] applied an SA-based approach to decide which connections should be eliminated. Heo and Oh [19] used genetic algorithm to node pruning of the BPN input and hidden layers (neural networks node pruning with genetic algorithm, NNPGA) and experiment with various databases from UCI machine learning repository. Their study showed that the NNPGA has a better performance.

The genetic algorithm has been extensively used in artificial neural network optimization and is known to achieve optimal solutions fair successfully. Li et al. [33] developed a GA-based combined BPN for estimation. The study shows that the BPN model combined with GA is more effective in finding the parameters of BPN than trial-and-error method. There are several genetic algorithm (GA) approaches used to enhance the BPN performance. They had been used in a variety of applications [7,21,23,28,31,42,48,52,60,70,71]. However, they only consider their problems; furthermore comparisons are hard to make.

2.2 Feature selection

The BPN requires a dataset to construct a model. A dataset may consist of many features, but not all features are helpful for classification. If the dataset has tremendous noise and complex dimensionality, the BPN may face limitations in learning the classification patterns. Feature selection may be considered part of the data cleaning and/or pre-processing step where the actual extraction and learning of knowledge or patterns is done after a suitable set of features is extracted. It is a process that aims to refine the list of features used, thereby removing potential sources of noise and ambiguity.

Approaches for feature selection may be categorized into two models, filter models and wrapper models [40]. In a filter model, statistical approaches, such as factor analysis (FA), independent component analysis (ICA), principal component analysis (PCA), and discriminant analysis (DA) have been devoted to the investigation of indirect performance measures, mostly based on distance and information measures, in feature selecting. Even though this model is faster, the resulting feature subset may not be optimal [40].

The wrapper model uses a variety of selection methods to choose feature subsets and then evaluates the result after the classification algorithm calculates the classification accuracy rate. If the relevant features can be selected or the noise removed, the classification accuracy rate classifier can be improved.

The wrapper model is widely used in BPN feature selection. Yang and Honavar [68] presented a DISTAL approach using feature selection to improve classification accuracy rate. Kim and Han [27] proposed a GA approach to perform feature selection in neural networks for the prediction of a stock price index. Lezoray and Cardot [32] employed floating search methods [29,45] to do the feature selection finding that the classification accuracy rate apparently improves after the feature selection. Verikas and Bacauskiene [62] presented a neural network-based feature selection technique. A network was trained with an augmented cross-entropy error function. Zhang et al. [72] applied a GA approach to feature selection in neural networks for fault defection in manufacturing industry. Sexton et al. [55] first calculated the network weight of architecture, then pruned the neural network architecture branches under a fixed number of hidden neurons, and finally used GA to conduct the feature selection. Sivagaminathan and Ramakrishnan [58] used ant colony optimization to optimize the feature subset that is suitable for feeding the neural network. Wang et al. [63] proposed a hybrid intelligent system called R-FC-DENN. Their experiments are carried out based on the UCI dataset. However, the above researches did not consider the parameter settings for network architectures of BPN at the same time.

Several researchers have proposed methods to obtain the optimal parameter settings for the network architectures of the BPN [5,14,15,26,53,54,66,69]. Nevertheless, the above research did not consider the feature selection to find the optimal feature subset in order to increase the performance simultaneously. Lin et al. [36] developed a simulated-annealing-based back-propagation network (SABPN) to determine parameter settings and feature selection simultaneously, but since the number of learning cycles is set to 500, the learning ability of the system may be limited.

2.3 Particle swarm optimization

Particle swarm optimization [25] is an emerging population-based meta-heuristic that simulates social behavior such as birds flocking to a promising position to achieve precise objectives in a multidimensional space. It has been applied successfully to a wide variety of highly complicated optimization problems [37] as well as various real-world problems [18,30,41,46,47]. Like evolutionary algorithms, PSO performs searches using a population (called a swarm) of individuals (called particles) that are updated from iteration to iteration. The size of population is denoted as p_{size}. To discover the optimal solution, each particle changes its search direction according to two factors, its own best previous experience (pbest) and the best experience of all other members (gbest). Shi and Eberhart [56] termed pbest the cognitive part, and gbest the social part.

Each particle represents a candidate position (i.e., solution). A particle is considered as a point in a D-dimensional space, and its status is characterized according to its position and velocity. The D-dimensional position for the particle i at iteration t can be represented as $x_i^t = \{x_{i1}^t, x_{i2}^t, \ldots, x_{iD}^t\}$. Likewise, the velocity (i.e., distance change) for particle i at iteration t, which is also a D-dimensional vector, can be described as $v_i^t = \{v_{i1}^t, v_{i2}^t, \ldots, v_{iD}^t\}$.

In the simple version of PSO, there was no actual control over the previous velocity of the particles. In the later versions of PSO, this shortcoming was addressed by incorporating a new parameter, called inertia weight introduced by Shi and Eberhart [57]. Let $p_i^t = \{p_{i1}^t, p_{i2}^t, \ldots, p_{iD}^t\}$ represent the best solution that particle i has obtained until iteration t, and $p_g^t = \{p_{g1}^t, p_{g2}^t, \ldots, p_{gD}^t\}$ denote the best solution obtained from p_i^t in the population at iteration t. To search for the optimal solution, each particle changes its velocity based on the cognitive and social parts as using Eq. (1).

$$V_{id}^t = w * V_{id}^{t-1} + c_1 r_1 (P_{id}^t - x_{id}^t) + c_2 r_2 (P_{gd}^t - x_{id}^t), \quad d = 1, 2, \ldots, D \tag{1}$$

where c_1 indicates the cognitive learning factor; c_2 indicates the social learning factor, inertial weight (w) is used to slowly reduce the velocity of the particles to keep the swarm under control, and r_1 and r_2 are random numbers uniformly distributed in U(0,1). It is possible to clamp the velocity vectors by specifying upper and lower bounds on $v_{\max}$ to avoid too rapid movement of particles in the search space. That is, the velocities of all the particles are limited within the range of $[-v_{\max}, v_{\max}]$ [51].

Each particle then moves to a new potential solution based on the following equation:

$$X_{id}^{t+1} = X_{id}^t + V_{id}^t, \quad d = 1, 2, \ldots, D \tag{2}$$

The basic process of the PSO algorithm is given as follows:

Step 1: (Initialization) Randomly generate initial particles.
Step 2: (Fitness) Measure the fitness of each particle in the population.
Step 3: (Update) Compute the velocity of each particle with Eq. (1).
Step 4: (Construction) For each particle, move to the next position according to Eq. (2).
Step 5: (Termination) Stop the algorithm if termination criterion is satisfied; return to Step 2 otherwise.

The process of PSO is finished if the termination condition is satisfied.

3 The proposed PSOBPN approach

The BPN uses the idea of "gradient steepest decent method" to minimize the errors between actual and predictive output functions. However, worse parameter values used may obtain the local optimal. Therefore, this study developed a particle swarm optimization (PSO) approach, termed PSOBPN, for parameter determination and feature selection in the BPN. The PSO approach is a multi-point search algorithm, which may be provide more probability to determine appropriate parameter values to escape the local optimal. In some complex problems the BPN may need more learning cycles to learn the pattern. Therefore, compared with the SA approach proposed by Lin et al. [36], a variable is added for determining the number of learning cycles of BPN.

In order to find the best parameter settings for the BPN, the classification accuracy rate of testing data is adopted as the fitness value of PSO-based approach. Therefore, for the condition without feature selection, four decision variables, designated learning iteration, learning rate, momentum term, and number of hidden neurons are required. Therefore, for the feature selection, if n features are required to decide which features are chosen, and then $4 + n$ decision variables must be adopted. The value of these n variables ranges between 0 and 1. If the value of a variable is less than or equal to 0.5, then its corresponding feature is not chosen. Conversely, if the value of a variable is greater than 0.5, then its corresponding feature is chosen. For example, if the data set has six attributes and the BPN requires four

2400	0.375	0.579	5	0.93	0.37	0.15	0.72	0.81	0.44

Fig. 1 An example of solution representation for the proposed approach

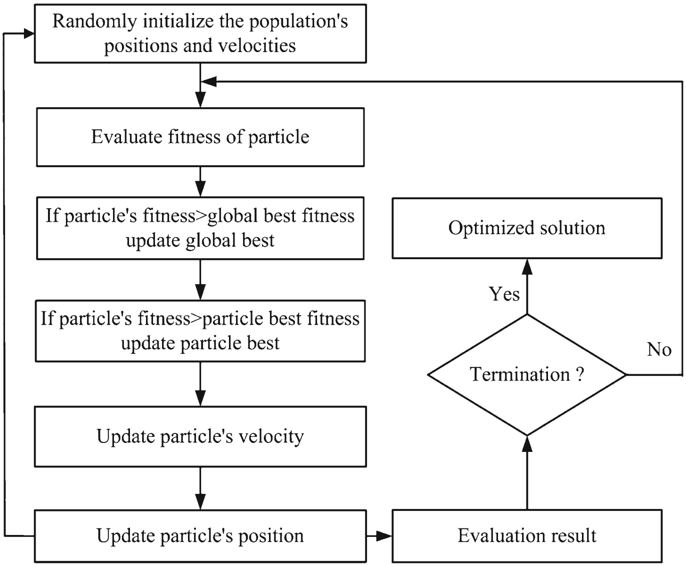
Randomly initialize the population's positions and velocities
Evaluate fitness of particle
If particle's fitness>global best fitness update global best
If particle's fitness>particle best fitness update particle best
Update particle's velocity
Update particle's position
Evaluation result
Termination ?
Yes
No
Optimized solution

Fig. 2 The flowchart of PSO algorithm

parameters, there are ten variables used as shown in Fig. 1. This solution can be decoded as follows. The learning cycle is 2,400, the learning rate is 0.375, the momentum is 0.579, the number of hidden neurons is 5, and the selected features are 1, 4, and 5. The parameter settings and selected features are then adopted to build a BPN classifier.

In order to discover the optimal solution of the PSOBPN, each particle represents a candidate solution and changes its search direction according to two factors, its own best previous experience (pbest) and the best experience of all other members (gbest). Figure 2 shows the flowchart for the PSOBPN. First, the population of particles is initialized, each particle having a random position within the D-dimensional space and a random velocity for each dimension. Second, each particle's fitness for the BPN is evaluated. Each particle's fitness in this study is the classification accuracy rate. If the fitness is better than the particle's best fitness, then the position vector is saved for the particle. If the particle's fitness is better than the global best fitness, then the position vector is saved for the global best. Finally the particle's velocity and position are updated until the termination condition is satisfied. The termination in this study is the pre-determined maximum number of solutions evaluated.

In this study, the classification accuracy rates for the datasets were measured by comparing the predict class and actual class. For example, in the classification problem with two-class positive and negative, a single prediction has the four different possible outcomes as shown in Table 1.

The true positive (TP) and true negative (TN) are correct classifications. A false positive (FP) occurs when the outcome is incorrectly predicted as positive when it is actually negative.

Table 1 Different outcomes of a two-class prediction

	Predicted class	
	Positive	Negative
Actual class		
Positive	TP	FN
Negative	FP	TN

A false negative (FN) occurs when the outcome is incorrectly predicted as negative when it is actually positive. The overall classification accuracy rate is the number of correct classifications divided by the total number of classifications, which is computed as $\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$.

In a multi-class prediction, the classification result is often displayed as a two-dimensional confusion matrix with a row and column for each class. Each matrix element shows the number of test cases for which the actual class is the row and the predicted class is the column.

To evaluate the classification accuracy rate, the k-fold approach [49] is used. This study set k as 10; that is, the data are divided into ten slices, and each slice of the data shares the same proportion of each class of data. Nine data slices were used as training data, while the tenth is used as the testing data. Since the number of data in each class was not a multiple of ten, the dataset could not be partitioned equitably. However, the ratio of the number of data in the training set to the number of data in the testing set was maintained as closely as possible to 9:1.The proposed PSOBPN approach was run ten times to allow each slice of data to take a turn as the testing data. The classification accuracy rate in of this experiment was calculated by summing the individual accuracy rate for each run of testing, and then dividing the total by ten.

4 Experimental results

The proposed approach was implemented using the C language and the Windows XP operating system and run on a personal computer with Pentium IV-3.0 GHz CPU and 512 MB of RAM. In order to verify the proposed PSOBPN approach, 23 datasets in the UCI Machine Learning Repository [20] were used for evaluation. The number of features, instances, and classes for each UCI dataset used in this research are shown in Table 2. Normalization is particularly useful for classification algorithms involving a neural network [17]. Thus, a feature is normalized by scaling its value so that it falls within scaled to $[-1, 1]$. If instance has missing values in some of its features, the instance is then removed. The predicted data of the Boston housing dataset was transformed from continuous into a binary class [12].

Parameter selection may influence the quality of the computational results. To avoid too rapid movement of particles in the search space, the lower and upper bounds on $v_{\max}$ is set to -1 and 1, respectively. In order to obtain better parameter values used in PSOBPN, the initial experiment is done as follows. Four datasets, Bupa, German, Pima, and Sonar, are used to test various combinations of parameters. At the beginning, the maximum number of solution evaluated is set to 50,000 (a large value), while w, c_1, c_2, and p_{size} is set to 1.0, 1.0, and 20, respectively. That is, the number of iterations is 2,500 (50,000/20 = 2,500). After several runs of execution, we found that the classification accuracy rate is stable when the maximum number of solution evaluated equal to 300 and 500 for the proposed approach

Table 2 Datasets from the UCI repository

Dataset	Number of features	Number of instances	Number of classes
Australian	14	690	2
Balance scale	4	625	3
Breast cancer (old)	9	699	2
Breast cancer (new)	32	569	2
Bupa liver	6	345	2
Car evaluation	9	1,728	4
CMC	9	1,473	3
German	30	1,000	2
Glass	9	214	6
Heart disease	13	270	2
Housing	13	506	2
Ionosphere structure	34	351	2
Iris	4	150	3
New thyroid	5	215	3
Pima Indians diabetes	8	768	2
Segmentation	19	2,310	7
Sonar	60	208	2
Vehicle	18	846	4
Vowel	10	990	11
Waveform with noise	40	5,000	3
Wine	13	178	3
Yeast	8	1,484	10
Zoo	17	101	7

without and with feature selection, respectively. After determining the maximum number of solution evaluated, the following combinations of parameters were tested:

$$\begin{aligned} \mathrm{c}_1 &= 0.5, 0.8, 1.0, 1.2, 1.5, 2.0; \\ \mathrm{c}_2 &= 0.5, 0.8, 1.0, 1.2, 1.5, 2.0; \\ p_{\text{size}} &= 5, 10, 15, 20, 25, 30; \\ w &= 0.7, 0.8, 0.9, 1.0; \end{aligned}$$

Setting $c_1 = 0.8$, $c_2 = 1.5$, $w = 0.9$, and $p_{\text{size}} = 10$ seemed to give better results; therefore they were used for further computational study. Because the proposed PSOBPN is a non-deterministic approach, the solution obtained may not be equal for the same data. Thus, the proposed PSOBPN approach is executed ten times for each fold in the dataset. When not considering feature selection, due to the maximum number of solutions evaluated was 300 and the number of particles set to be 10, the number of iterations equals 30 (300/10). With feature selection, the number of features selected for use can be obtained by the PSOBPN approach. Since the PSOBPN approach has a larger solution space, in terms of number of features, the number of solutions evaluated is also larger. Because the maximum number of solutions evaluated was raised to 500, the number of iterations was 50 (500/10). The search

Table 3 The classification accuracy rate of training and testing data set (%)

	Without feature selection		With feature selection	
	Training data	Testing data	Training data	Testing data
Australian	93.74	93.85	91.88	92.80
Balance scale	99.56	99.36	99.47	99.36
Breast cancer (old)	99.19	98.83	98.49	98.98
Breast cancer (new)	99.68	98.02	99.26	99.28
Bupa liver	82.45	81.39	74.01	81.79
Car evaluation	99.94	99.93	100.00	99.93
CMC	65.67	60.82	61.30	62.52
German	92.81	80.58	86.62	81.88
Glass	89.63	86.14	90.05	86.66
Heart disease	96.58	91.20	91.59	94.24
Housing	99.24	99.71	98.76	99.51
Ionosphere structure	99.84	97.72	99.31	99.19
Iris	90.87	95.06	93.83	97.56
New thyroid	99.90	99.05	99.74	99.52
Pima Indians diabetes	82.60	82.16	78.96	83.33
Segmentation	99.10	98.32	99.51	98.49
Sonar	99.07	89.93	100.00	97.61
Vehicle	96.72	90.41	96.36	90.65
Vowel	98.57	98.18	98.31	98.55
Waveform with noise	93.45	87.26	94.53	87.18
Wine	98.37	100.00	100.00	100.00
Yeast	65.75	64.42	65.08	64.42
Zoo	100.00	97.00	99.45	100.00

ranges for parameter values of BPN are set as follows. The learning cycle ranged from 500 to 5000, the learning rate ranged from 0 to 0.45, while the momentum term ranged from 0.4 to 0.9. The BPN used one hidden layer and the sigmoid transfer function. The number of hidden neurons ranged from one to the number of features in two-class datasets, while the number of hidden neurons ranged from the number of classes to two multiplied by the sums of the number of input features and the number of output neurons.

The classification accuracy rates obtained by the proposed PSOBPN approach for each data set (both the training data and testing data) are shown in Table 3. It is noted that because the difference in classification accuracy rates between the training data and testing data is not excessive, the proposed approach can avoid the over-fitting problem and achieve better classification accuracy rate.

In this study, results obtained by PSOBPN without feature selection are then compared with those of SABPN [36], SASVM [37], PSOSVM [38], MONNA [32], GA DISTAL [68], G-Prop [6], and NNPGA [19] as shown in Table 4. Only the classification accuracy rates of Breast Cancer (old), (98.3% in PSOBPN and 99.0% in G-Prop), German (80.58% in PSOBPN and 80.92% in SABPN), Housing (99.71% in PSOBPN and 99.90% in SASVM), Iris (95.16% in PSOBPN and 97.33% in NNPGA), Segmentation (98.32% in PSOBPN and

Table 4 The comparison results of approaches without feature selection (%)

Dataset	SABPN	SASVM	PSOSVM	MONNA	GA DISTAL	G-Prop	NNPGA	PSOBPN	Time (s)
Australian	90.85	88.34	88.09	–	–	–	–	93.85[a]	450
Balance scale	–	–	–	–	–	–	92.74	99.36[a]	541
Breast cancer (old)	98.31	97.95	97.95	97.8	97.8	99.0a	–	98.83	321
Bupa liver	80.10	80.81	80.81	–	–	–	–	81.39[a]	193
German	80.92[a]	–	79.00	–	–	–	74.40	80.58	3610
Glass	81.31	78.38	78.04	82.2	70.5	69.0	70.00	86.14[a]	590
Heart disease	90.88	87.97	88.17	–	85.3	–	85.19	91.20[a]	235
Housing	96.19	99.90[a]	99.90[a]	–	86.3	–	–	99.71	2198
Ionosphere structure	97.66	97.50	97.50	90.1	94.3	–	91.14	97.72[a]	3764
Iris	–	–	–	–	–		97.33[a]	95.06	72
Pima Indians diabetes	82.16[a]	80.19	80.19	76.6	76.3	–	77.50	82.16[a]	446
Segmentation	–	–	–	–	–		99.57[a]	98.32	13566
Sonar	91.88[a]	91.85	88.32	–	83.0	–	82.00	89.93	1161
Vehicle	89.45	88.76	88.71	78.4	65.4	–	85.36	90.41[a]	3586
Vowel	97.85	99.27	99.27	–	69.8	–	99.90[a]	98.18	3167
Waveform with noise	–	–	–	–	–	–	84.88	87.26[a]	49350
Wine	–	–	99.56	–	–	–	–	100.00[a]	295
Zoo	–	–	–	–	–	–	97.00[a]	97.00[a]	438

– Approach did not use this dataset for test

[a] The higher classification accuracy rate among approaches

99.57% in NNPGA) Sonar (89.93% in PSOBPN and 91.88% in SABPN), and Vowel (98.18% in PSOBPN and 99.90% in NNPGA), obtained by PSOBPN are inferior to the classification accuracy rates of other approaches. The remaining classification accuracy rates obtained by the proposed PSOBPN approach are higher than those of other approaches, and the computation time is within an acceptable range.

Table 5 presents the results obtained by PSOBPN with feature selection and a number of published results using other machine learning algorithms. Comparing the classification accuracy rates of the proposed PSOBPN approach with those of SABPN [36], SASVM [37], PSOSVM [38], MONNA [32], GA+DISTAL [68], GAP (genetic algorithm and programming) [59], C4.5 (J48, the WEKA implementation of C4.5) [59], HIDER [10], XCS (Wilson's XCS classifier) [10], OFA (ordered fuzzy ARTMAP) [2], LVSM (Lagrangian support vector machines) [44], and R-FC-DENN [63] only three two-class problems and five multi-class problems whose classification accuracy rates obtained by PSOBPN are lower than those of other approaches. Comparing the PSOBPN without feature selection with feature selection, better results are obtained by implementing the feature selection at the expense of some computation cost (Table 5). Although PSOBPN did not achieve the best classification results across these eight problems, the difference is not great. In general, the PSOBPN approach with feature selection performs well in both two-class and multi-class problems. Further, with feature selection, the proposed PSOBPN approach can also effectively delete certain moder-

Table 5 Comparison between the proposed PSOBPN approach with feature selection and other approaches on the UCI data sets (%)

Dataset	SABPN	SASVM	PSOSVM	MONNA (SFFS)	MONNA (SBS)	MONNA (SFS)	GA DISTAL	GAP	C 4.5 (J48)	HIDER	XCS	OFA	LVSM	R-FC-DENN	PSOBPN	Time (se)
Australian	92.07	92.19	91.03	–	–	–	–	–	–	–	–	–	–	–	92.80[a]	506
Breast cancer (old)	98.69	99.38[a]	99.18	98.54	98.50	98.50	99.30	95.56	94.42	95.71	96.27	94.39	–	74.33	98.98	316
Breast cancer (new)	–	–	–	–	–	–	–	95.71	93.88	–	–	–	–	–	99.28[a]	294
Bupa liver	81.52	83.13	82.05	–	–	–	–	66.55	66.37	64.29	67.85	57.01	68.68	–	81.79[a]	151
Car evaluation	–	–	–	–	–	–	–	–	–	–	–	–	–	66.96	99.93[a]	4,386
CMC	–	–	–	–	–	–	–	–	–	–	–	–	–	68.66[a]	62.52	2,301
German	82.06[a]	–	81.62	–	–	–	–	–	–	–	–	–	–	–	81.88	5,171
Glass	83.29	87.93[a]	85.26	84.40	84.40	84.40	80.80	71.84	68.28	70.59	72.53	69.56	–	–	86.66	1019
Heart disease	93.57	93.33	92.83	–	–	–	92.90	–	–	–	–	–	–	–	94.24[a]	247
Housing	95.55	100.00[a]	100.00[a]	–	–	–	92.60	–	–	–	–	–	–	–	99.51	2,389
Ionosphere structure	98.50	99.07	99.01	95.80	92.90	94.40	98.60	90.69	89.82	–	–	–	87.75	–	99.19[a]	988
Iris	–	–	–	–	–	–	–	–	–	–	–	–	–	80.00	97.56[a]	125
New thyroid	–	–	–	–	–	–	–	96.49	–	–	–	–	–	–	99.52[a]	211
Pima Indians diabetes	83.08	82.22	82.68	77.90	76.60	76.60	79.50	73.64	73.32	74.10	68.62	69.80	78.12	–	83.33[a]	338
Sonar	93.90	95.99	96.26	–	–	–	97.20	75.89	73.86	59.93	53.41	79.96	–	–	97.61[a]	915
Vehicle	89.47	90.14	89.83	92.45[a]	84.21	80.11	68.80	72.11	72.22	–	–	–	–	–	90.65	4,876
Vowel	97.51	100.00[a]	100.00[a]	–	–	–	78.40	–	–	–	–	–	–	–	98.55	8158
Wine	–	–	100.00[a]	–	–	–	–	96.10	93.27	96.05	92.74	98.27	–	–	100.00[a]	418
Yeast	–	–	–	–	–	–	–	–	–	–	–	–	–	67.16[a]	64.42	9,048
Zoo	–	–	–	–	–	–	–	–	–	–	–	–	–	81.36	100.00[a]	447

– Approach did not use this dataset for test

[a] The higher classification accuracy rate among approaches

Table 6 Comparison of the classification accuracy rate for PSOBPN approach with/without feature selection (%)

Dataset	With feature selection	Without feature selection	*P* value
Australian	92.80	93.85	–
Balance scale	99.36	99.36	$0.500 > \alpha$
Breast cancer (old)	98.98	98.83	$0.062 > \alpha$
Breast cancer (new)	99.68	98.02	$0.009 < \alpha$
Bupa liver	81.97	81.39	$0.024 < \alpha$
Car evaluation	99.93	99.93	$0.500 > \alpha$
CMC	62.52	60.82	$0.001 < \alpha$
German	81.88	80.38	$0.001 < \alpha$
Glass	86.66	86.14	$0.045 < \alpha$
Heart disease	94.24	91.20	$0.001 < \alpha$
Housing	99.51	99.71	–
Ionosphere structure	99.19	97.72	$0.001 < \alpha$
Iris	97.56	95.06	$0.002 < \alpha$
Pima Indians diabetes	83.33	82.16	$0.001 < \alpha$
Segmentation	98.49	98.32	$0.021 < \alpha$
Sonar	97.61	89.93	$0.001 < \alpha$
Vehicle	90.65	90.41	$0.110 > \alpha$
Vowel	98.55	98.18	$0.045 < \alpha$
Wine	100.00	100.00	$0.500 > \alpha$
Waveform with noise	87.18	87.26	–
Yeast	64.42	64.42	$0.500 > \alpha$
Zoo	100.00	97.00	$0.001 < \alpha$

Confidence level $\alpha = 0.05$
– The classification accuracy rate is reduced with feature selection

ating or non-affecting features while maintaining the same or better classification accuracy rate. The importance of the relationship of the remaining features for classification may be examined in the future.

Compared with the SABPN [36], one variable is added for determining the number of learning cycle of BPN in PSOBPN. Experimental results showed that the proposed PSOBPN can achieve better classification accuracy rate than those of SABPN in most of datasets. That is, the BPN may need more learning cycle to learn the pattern in some complex problems. The experimental results showed that the classification accuracy rates obtained by PSOBPN are better than those of SABPN in general. The result is accordance with the observation provided by Schittenkopf et al. [50].

Finally, in order to verify whether a significant difference between the proposed PSOBPN approach with feature selection and without feature selection exists, the results of the proposed PSOBPN approach with and without feature selection are compared, as shown in Table 6. It can be noted in this Table, although the classification accuracy rates of three datasets (Australian, Housing, and Waveform with noise) are reduced and four dataset (Balance scale, Car Evaluation, Wine, and Yeast) are same, those of the remaining datasets are increased. For the dataset whose classification accuracy rate increased, only the Breast (old) and Vehicle dataset does not have significant difference; all other datasets have p-value lower than 0.05, which means the significant difference exist. Therefore, the proposed PSOBPN approach with feature selection is better than the proposed PSOBPN approach without feature selection.

5 Conclusions and future research

This study applied the particle swarm optimization-based approach to search for appropriate parameter values and beneficial features for BPN. The main contributions of this study include

(1) The trial-and-error method traditionally used for BPN in determining the parameter is time-consuming and cannot guarantee better results. The proposed approach can be used to automatically determine the parameter values for BPN.

(2) When the feature selection is taken into account it can improve the performance, reduce dimensionality, and remove noise of the BPN. The result showed that the classification accuracy rates are significantly increased in many data sets when feature selection is taken into account in BPN.

Compared with the previous studies, the classification accuracy rates of the proposed PSOBPN approach are better than those of other approaches. Furthermore, with the feature selection the proposed PSOPBN approach can also effectively delete some moderating or non-affecting features while maintaining the same or better classification accuracy rate. The results are by no means an exhaustive list of current machine learning algorithms, nor are they guaranteed to be the best performing algorithms available, but they give some results of the relative performance of our approach, which appears to be very good.

More studies can be done in the future. First, since the proposed approach is only applied and compared with others using classification problems, the efficiency of the proposed approach in the forecast of continuous values will be examined in the future. Second, the proposed PSO-based meta-heuristic is sensitive to parameter settings, and sometimes it may result in local optimal. Therefore, to perform a comprehensive study on alternative parameter tuning policies and to customize the algorithm by developing new parameter and new mechanism is room for further investigation. For example, other parameters, such as time interval, time-varying inertia weight, may be added to the PSO algorithm, while particle grouping may be used to split the population of particles into subgroups. These techniques may help PSO to avoid falling into local optimal and improve the performance. Third, since PSO is a meta-heuristics, it can be applied to other architectures of network, such as cascade neural network and RBF network (radial basis function network). Moreover, PSOBPN may be applied to a greater range of problems in the real world.

Acknowledgments This work was partially funded by National Science Council of the Republic of China, Taiwan, and Chang Gung University, Taiwan, under Contract No. NSC97-2410-H-211-001-MY2 and Contract No. UARPD370101, respectively. The authors are indebted to two anonymous referees for their thorough reviews and the insights they provided.

References

1. Abe N, Kudo M (2006) Non-parametric classifier-independent feature selection. Pattern Recognit 39: 737–746
2. Ahluwalia M, Bull L (1999) Coevolving functions in genetic programming: classification using k-nearest-neighbour, In: Banzhaf W, Daida J, Eiben G et al (eds) GECCO-99. In: Proceedings of the genetic and evolutionary computation conference. Morgan Kaufmann, Florida, pp 947–952
3. Barletta M, Gisario A, Guarino S (2007) Modeling of electrostatic fluidized bed (EFB) coating process using artificial neural networks. Eng Appl Artif Intell 20:721–733
4. Berry MJA, Linoff G (eds) (2001) Data mining techniques: for marking, sales and customer support. Wiley, NY
5. Castillo PA, Carpio J, Merelo JJ et al (2000a) Evolving multilayer perceptrons. Neural Process Lett 12:115–127
6. Castillo PA, Merelo JJ, Prieto A et al (2000b) G-prop: global optimization of multilayer perceptrons using GAs. Neurocomputing 35:149–163
7. Chen W-C, Fu GL, Tai P-H et al (2009) Process parameter optimization for MIMO plastic injection molding via soft computing. Expert Syst Appl 36:1114–1122
8. Choi YS (2001) Discovering text databases with neural nets. Knowl Inf Syst 3:356–373
9. Danaher S, Datta S, Waddle I et al (2004) Erosion modeling using bayesian regulated artificial neural networks. Wear 256:879–888
10. Dixon PW, Corne DW, Oates MJ (2002) A preliminary investigation of modified XCS as a generic data mining tool. In: Lanai PL, Stolzmann W, Wilson S (eds) Advertences in learning classifier systems. Springer, Berlin, pp 257–269
11. Ebrahimzadeh A, Ranjbar A (2008) Intelligent digital signal-type identification. Eng Appl Artif Intell 21:569–577
12. Fung G, Mangasarian OL (2003) Finite Newton method for Lagrangian support vector machine classification. Neurocomputing 55:39–55
13. Ghosh A, Jain LC (2004) Evolutionary computation in data mining. Springer, Germany
14. Ghosh R, Verma B (2003) A hierarchical method for finding optimal architecture and weights using evolutionary least square based learning. Int J Neural Syst 13:13–24
15. Gupta JND, Sexton RS (1999) Comparing backpropagation with a genetic algorithm for neural network training. Int J Manag Sci 27:679–684
16. Gusikhin O, Rychtyckyj N, Filev D (2007) Intelligent systems in the automotive industry: applications and trends. Knowl Inf Syst 12:147–168
17. Han J, Kamber M (2007) Data mining: concepts and techniques. Morgan Kaufmann, San Francisco
18. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20:89–99
19. Heo G-S, Oh I-S (2008) Simultaneous node pruning of input and hidden layers using genetic algorithms. In: Proceedings of IEEE the seventh international conference on machine learning and cybernetics. pp 3428–3433
20. Hettich S, Blake CL, Merz CJ (1998) UCI repository of machine learning databases, Department of Information and Computer Sciences, University of California, Irvine, CA. http://www.ics.uci.edu/~mlearn/MLRepository.html
21. Hou T-H, Su C-H, Chang H-Z (2008) Using neural networks and immune algorithms to find the optimal parameters for an IC wire bonding process. Expert Syst Appl 34:427–436
22. Huang C-Y, Chen L-H, Chen Y-L et al (2009) Evaluating the process of a genetic algorithm to improve the back-propagation network: a Monte Carlo study. Expert Syst Appl 36:1459–1465
23. Javadi AA, Farmani R, Tan TP (2005) A hybrid intelligent genetic algorithm. Adv Eng Inform 19: 225–262
24. Karpenko M, Sepehri N (2002) Neural network classifiers applied to condition monitoring of a pneumatic process valve actuator. Eng Appl Artif Intell 15:273–283
25. Kennedy J, Eberhart R (1995) Particle swarm optimization. Proc IEEE Conf Neural Netw 4:1942–1948
26. Khaw JFC, Lim BS, Lim LEN (1995) Optimal design of neural networks using the Taguchi method. Neurocomputing 7:225–245
27. Kim K-J, Han I (2000) Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. Expert Syst Appl 19:125–132
28. Ko Y-D, Moon P, Kim CE et al (2009) Modeling and optimization of the growth rate for ZnO thin films using neural networks and genetic algorithms. Expert Syst Appl 36:4061–4066
29. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97:273–324

30. Kwok NM, Liu DK, Dissanayake G (2006) Evolutionary computing based mobile robot localization. Eng Appl Artif Intell 19:857–868
31. Lee J, Kang S (2007) GA based meta-modeling of BPN architecture for constrained approximate optimization. Int J Solids Struct 44:5980–5993
32. Lezoray O, Cardot H (2001) A neural network architecture for data classification. Int J Neural Syst 11:33–42
33. Li S, Li Y, Liu Y et al (2007) A GA-based NN approach for makespan estimation. Appl Math Comput 185:1003–1014
34. Li W-S, Clifton C, Liu S-Y (2000) Database integration using neural networks: implementation and experiences. Knowl Inf Syst 2:73–96
35. Lin SC, Ting CJ (1996) Drill wear monitoring using neural networks. Int J Mach Tools Manuf 36:465–475
36. Lin S-W, Tseng T-Y, Chou S-Y et al (2008) A simulated annealing based approach for simultaneous parameter optimization and feature selection of back-propagation networks. Expert Syst Appl 34: 1491–1499
37. Lin S-W, Lee Z-J, Chen S-C et al (2008) Parameter determination of support vector machines and feature selection using simulated annealing approach. Appl Soft Comput 8:1505–1512
38. Lin S-W, Ying K-C, Chen S-C et al (2008) Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Syst Appl 35:1817–1824
39. Lin S-W, Chou S-Y, Chen S-C (2007) Irregular shapes classification by back-propagation neural network. Int J Adv Manuf Technol 34:1164–1172
40. Liu H, Motoda H (1998) Feature selection for knowledge discovery and data mining. Kluwer Academic, Boston
41. Liu L, Liu W, Cartes DA (2008) Particle swarm optimization-based parameter identification applied to permanent magnet synchronous motors. Eng Appl Artif Intell 21:1092–1100
42. Mahesh Parappagoudar B, Pratihar DK, Datta GL (2008) Forward and reverse mappings in green sand mould system using neural networks. Appl Soft Comput 8:239–260
43. Malhotra R, Malhotra DK (2003) Evaluating consumer loans using neural networks. Int J Manag Sci 31:83–96
44. Mangasarian OL, Musicant DR (2001) Lagrangian support vector machine. J Mach Res 1:161–177
45. Pudil P, Novovičová J, Kittler J (1994) Floating search methods in feature selection. Pattern Recognit Lett 15:1119–1125
46. Peng T, Zuo W, He F (2008) SVM based adaptive learning method for text classification from positive and unlabeled documents. Knowl Inf Syst 16:281–301
47. Qiao W, Gao Z, Harley RG et al (2008) Robust neuro-identification of nonlinear plants in electric power systems with missing sensor measurements. Eng Appl Artif Intell 21:604–618
48. Sasaki T, Tokoro M (2000) Comparison between Lamarckian and Darwinian evolution on a model using neural networks and genetic algorithms. Knowl Inf Syst 2:201–222
49. Salzberg SL (1997) On comparing classifiers: pitfalls to avoid and a recommended approach. Data Min Knowl Discov 1:317–327
50. Schittenkopf C, Deco G, Brauer W (1997) Two strategies to avoid overfitting in feedforward networks. Neural Netw 10:505–516
51. Senthil Arumugam M, Rao MVC, Chandramohan A (2008) A new and improved version of particle swarm optimization algorithm with global-local best parameters. Knowl Inf Syst 16:331–357
52. Sedki A, Ouazar D, El Mazoudi E (2009) Evolving neural network using real coded genetic algorithm for daily rainfall-runoff forecasting. Expert Syst Appl 36:4523–4527
53. Sexton RS, Alidaee B, Dorsey RE et al (1998) Global optimization for artificial neural networks: a tabu search application. Eur J Oper Res 106:570–584
54. Sexton RS, Dorsey RE, Johnson JD (1999) Optimization of neural networks: a comparative analysis of the genetic algorithm and simulated annealing. Eur J Oper Res 114:589–601
55. Sexton RS, McMurtrey S, Cleavenger DJ (2006) Knowledge discovery using a neural network simultaneous optimization algorithm on a real world classification problem. Eur J Oper Res 168:1009–1018
56. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: Proceeding of the IEEE congress on Evolutionary Computation. pp 69–73
57. Shi Y, Eberhart R (1998) Parameter selection in particle swarm optimization. Lecture notes in computer science, vol 1447. pp 591–600
58. Sivagaminathan RK, Ramakrishnan S (2007) A hybrid approach for feature subset selection using neural networks and ant colony optimization. Expert Syst Appl 33:49–60
59. Smith M, Bull L (2005) GAP: constructing and selection features with evolutionary computing, In: Ghosh A, Jain LC (eds) Evolutionary computation in data mining. Springer, Heidelberg

60. Srinivas V, Ramanjaneyulu K (2007) An integrated approach for optimum design of bridge decks using genetic algorithms and artificial neural networks. Adv Eng Softw 38:475–487
61. Tsai C-Y, Chou S-Y, Lin S-W et al (2009) Location determination of mobile devices for an indoor WLAN application using a neural network. Knowl Inf Syst 20:81–93
62. Verikas A, Bacauskiene M (2002) Feature selection with neural networks. Pattern Recognit Lett 23:1323–1335
63. Wang G, Zhang C, Huang L (2008) A study of classification algorithm for data mining based on hybrid intelligent systems. In: Proceeding of IEEE Ninth ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing. pp 371–375
64. Wang T-Y, Huang C-Y (2007) Applying optimized BPN to a chaotic time series problem. Expert Syst Appl 32:193–200
65. Wang X, Wang W, Huang Y et al (2008) Design of neural network-based estimator for tool wear modeling in hard turning. J Intell Manuf 19:383–396
66. Yamazaki A, Souto MCPde , Ludermir TB (2002) Optimization of neural network weights and architectures for odor recognition using simulated annealing. Proc IEEE Neural Netw 1:547–552
67. Yan H, Jiang Y, Zheng J et al (2006) A multilayer perceptorn-based medical decision support system for heart disease diagnosis. Expert Syst Appl 30:272–281
68. Yang J, Honavar V (1998) Feature subset selection using a genetic algorithm. IEEE Intell Syst 34:44–49
69. Yeung DS, Zeng X-Q (2002) Hidden neuron pruning for multilayer perceptrons using a sensitivity measure. Proc IEEE Mach Learn Cybern 4:1751–1757
70. Yu S, Zhu K, Diao F (2008) A dynamic all parameters adaptive BP neural networks model and its application on oil reservoir prediction. Appl Math Comput 195:66–75
71. Yuen CWM, Wong WK, Qian SQ et al (2009) A hybrid model using genetic algorithm and neural network for classifying garment defects. Expert Syst Appl 36:2037–2047
72. Zhang L, Jack LB, Nandi AK (2005) Fault detection using genetic programming. Mech Syst Signal Process 19:271–289

Author biographies

Shih-Wei Lin is an Associate Professor in Department of Information Management, Chang Gung University, Taiwan. He received his Ph.D. in Industrial Management from the National Taiwan University of Science and Technology in 2002. His current research interests include scheduling and data mining. His papers have appeared in Computers and Operations Research, European Journal of Operational Research, Journal of the Operational Research Society, International Journal of Production Research, International Journal of Advanced Manufacturing Technology, Knowledge and Information Systems, Applied Soft Computing Applied Intelligence and Expert Systems with Applications, etc.

Shih-Chieh Chen is a Ph.D. candidate at National Taiwan University of Science and Technology. He received his MS degree from Hua-fan University of Taiwan in 2005. His current research interest is data mining.

Wen-Jie Wu is an assistant professor in Department of Information Management, Chang Gung University, Taiwan. He received the B.S. Degree in computer science and engineering from Tatung University, Taipei, Taiwan, in 1996, and the M.S. degree and Ph.D. degree in computer science and information engineering from Nation Chung Cheng University, Chiayi, Taiwan, in 1998 and 2003, respectively. His research interests include image processing, medical computer-aided diagnosis system, and data mining.

Chih-Hsien Chen is an Associate Professor in Department of Management, Fo-Guang University, Taiwan. His research interests include performance evaluation, continuous improvement, and supply chain management. His papers have appeared in Total Quality Management and Business Excellent, Journal of Industrial Technology, etc. He received his Ph.D. in industrial management from the National Taiwan University of Science and Technology in 2005.