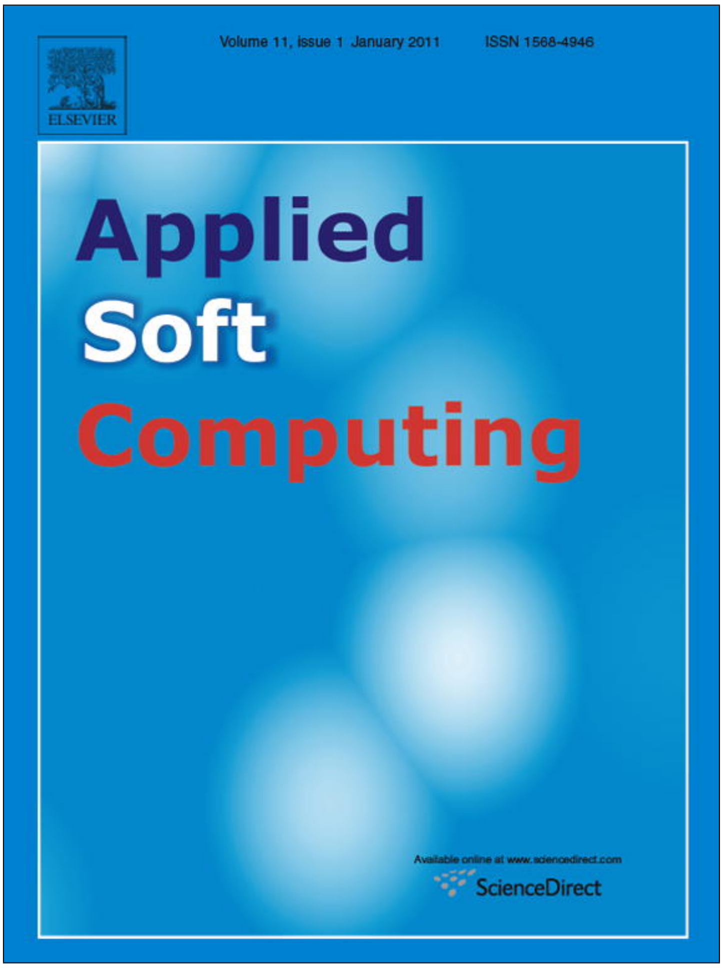
Volume 11, issue 1 January 2011
ISSN 1568-4946
ELSEVIER
Applied
Soft
Computing
Available online at www.sciencedirect.com
ScienceDirect

Applied Soft Computing 11 (2011) 1427–1438

Contents lists available at ScienceDirect

Applied Soft Computing

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

Radial basis function network based on time variant multi-objective particle swarm optimization for medical diseases diagnosis

Sultan Noman Qasem*, Siti Mariyam Shamsuddin

Soft Computing Research Group, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Skudai, 81310 Johor, Malaysia

ARTICLE INFO

Article history:
Received 18 November 2009
Received in revised form 26 February 2010
Accepted 28 April 2010
Available online 5 May 2010

Keywords:
Radial basis function network
Hybrid learning
Particle swarm optimization
Time variant multi-objective particle swarm optimization
Multi-objective particle swarm optimization
Elitist non-dominated sorting genetic algorithm

ABSTRACT

This paper proposes an adaptive evolutionary radial basis function (RBF) network algorithm to evolve accuracy and connections (centers and weights) of RBF networks simultaneously. The problem of hybrid learning of RBF network is discussed with the multi-objective optimization methods to improve classification accuracy for medical disease diagnosis. In this paper, we introduce a time variant multi-objective particle swarm optimization (TVMOPSO) of radial basis function (RBF) network for diagnosing the medical diseases. This study applied RBF network training to determine whether RBF networks can be developed using TVMOPSO, and the performance is validated based on accuracy and complexity. Our approach is tested on three standard data sets from UCI machine learning repository. The results show that our approach is a viable alternative and provides an effective means to solve multi-objective RBF network for medical disease diagnosis. It is better than RBF network based on MOPSO and NSGA-II, and also competitive with other methods in the literature.

1. Introduction

Artificial Neural Networks (ANNs) have been developed in the form of parallel-distributed network models based on the biological learning process of the human brain. ANNs are computational models inspired by a biological nervous system with applications in science and engineering. Radial Basis Function (RBF) networks are typed of ANNs, and they were introduced into the neural network literature by Broomhead and Lowe [1]. RBF networks are motivated by observation on the local response in biologic neurons. Due to their better approximation capabilities, simpler network structures and faster learning algorithms, RBF networks have certain advantages over other types of ANNs and have been widely applied in many science and engineering fields. It has three layered feed-forward and fully connected network, which uses RBFs as the only nonlinearly in the hidden layer neurons. The output layer has no nonlinearly and the connections of the output layer are only weighted; the connections from the input to the hidden layer are not weighted [2].

In recent years, there have been many studies in solving the problem of ANN training and structure optimization. Most of them have been applied to feed-forward models. The study in [3] provides a general framework for using evolutionary algorithms for ANNs. Other authors have used single-objective evolutionary algorithms to evolve a set of networks of different sizes in the same population. In recent years, Pareto-based multi-objective algorithms have proven to be more promising tools for training and optimizing the size of a neural network. For example, multi-objective approaches may force the search process to find a set of optimal solutions instead of a single one. Furthermore, a Pareto-based approach may be preferred to a linear weight aggregation procedure since this last type of method may entail some undesirable characteristics when combining different error measures such as those mentioned in [4]. Considering the set of Pareto-based multi-objective procedures [5,6], population-based ones might be preferred since they may speed up the search and optimization process.

Other studies attempted to use Pareto-based multi-objective algorithms to evolve the structure of neural networks and train them simultaneously. For example, the studies in [7,8] introduce the use of multi-objectivity to coevolve ensembles to build feed-forward networks. Liu and Kadirkamanathan [9] studied the benefits of multi-objective optimization for identifying nonlinear systems while optimizing the size of neural networks. In [10], the authors provide one of the first approximations for optimizing the size and parameters of RBF networks. The study in [11] extended this paper [10], using multi-objective optimization to

* Corresponding author. Tel.: +60 177885930.
E-mail addresses: sultannoman@hotmail.com (S.N. Qasem), mariyam@utm.my (S.M. Shamsuddin).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.04.014

find the best of radial basis function, number of hidden units, centers and widths of RBF networks. Other approaches use the Pareto-optimality criterion to train a multilayer perceptron, considering different error measures as objectives to be achieved [4]. In [12], the multi-objective method proposed includes differential evolution to find the optimal size of the hidden units and also train the network for a single layer perceptron. The same author improved his work in [13] with multi-objective ANNs (MPANN), which are a multi-objective algorithm that combines Pareto-based multi-objective algorithms with local search to optimize the number of hidden nodes and the perceptron training. In [14], MPANN is applied to breast cancer diagnosis, and promising results have been obtained. Abbass [15] studied the benefits of hybridizing Pareto-based evolutionary algorithms with the back-propagation training method. The same author also studied the improvement in network performance using different formulations for multi-objective optimization in [16].

Other authors have focused on the problem of multi-objective optimization of feed-forward NNs as a solution for the regularization problems in the network's complexity [17]. In this case, the optimization of the structure is carried out by minimizing the number of network connections. The same authors extend these ideas by including local search in the evolutionary process in [18] and to improve the generalization capabilities of the networks and their interpretability in [19,20]. Although, many authors offer many competitive solutions for feed-forward networks, this is not the case for RBF network, where hybrid learning in this network increases the complexity.

Although there are few studies on the implementation of multi-objective RBF network training, but research on training of RBF network with multi-objective swarm intelligence is still new. This section presents some existing work of training RBF network based on Multi-Objective Evolutionary Algorithms (MOEAs).

In [21], a multi-objective (MOBJ) optimization algorithm has been applied to the problem of inductive supervised learning depended on smoothness based apparent complexity measure for RBF networks. However, the computational complexity of the proposed algorithm is high in comparison with other state-of-the-art machine learning methods. A multi-objective genetic algorithm based design procedure for the RBF network has been proposed in [22]. A Hierarchical Rank Density Genetic Algorithm (HRDGA) has been developed to evolve both the neural network's topology and its parameters simultaneously.

A method in which RBF network ensemble is constructed from Pareto-optimal set obtained by multi-objective evolutionary computation has been proposed in [23]. Pareto-optimal set of RBF networks has been acquired by multi-objective GA based on three criteria; model complexity, representation ability, and model smoothness. A new evolutionary algorithm, the RBF-Gene algorithm, has been applied to optimize RBF networks [24]. Unlike other works, the algorithm can evolve both from the structure and the numerical parameters of the network: it can evolve the number of neurons and their weights.

A study in [25] presented RBF network optimization from training examples as a multi-objective problem and an evolutionary algorithm has been proposed to solve it. This algorithm incorporates mutation operators to guide the search for good solutions. A method of obtaining Pareto-optimal RBF network set based on multi-objective evolutionary algorithms has been proposed in [26]. On the other hand, Ferreira et al. [27] proposed a multi-objective genetic algorithm for identification of RBF network couple models of humidity and temperature in a greenhouse. Two combinations of performance and complexity criteria were used to steer the selection of model structures, resulting in distinct sets of solutions.

Unlike previous studies, our approach deals with the problem of RBF network hybrid learning (unsupervised learning and supervised learning) that shares with TVMOPSO. This mechanism evolves toward Pareto-optimal front defined by several objective functions with model accuracy and complexity. This proposed approach is simple with faster convergence to Pareto-optimal solutions. In this study, three benchmark data sets and two well-known MOEAs (i.e., MOPSO [35] and NSGA-II [34]) for RBF network are conducted. The comparison with previous methods is implemented to examine the efficiency and effectiveness of our proposed method. There are two main advantages of the proposed method; first, the proposed method can be applied to any real-world problem and second, it shows better performance in terms of error, sensitivity, specificity and accuracy for benchmark data sets. In other cases, the proposed method still has acceptable performance and produces the results that are comparable with the results produced by the existing methods. The better performance of the proposed method may be attributed to all features i.e., mutation operator, diversity measure and the adaptive control parameters. In addition, our proposed method is good not only in approximating the Pareto-optimal RBF network, but also in terms of diversity of the solutions on the front.

The structure of this paper is organized as follows: Section 2 provides the Background Materials of the study, and follows by the explanation on the RBFN-TVMOPSO in Section 3. Experimental Results and Discussions are given in Section 4, and Conclusions are drawn in Section 5.

2. Background materials

2.1. RBF network

RBF network is one of the most important ANN paradigms in machine learning field. It is a feed-forward network with a single layer of hidden units, called radial basis functions (RBFs). RBF outputs show the maximum value at its center point and decrease its output value as the input leaves the center. Typically, the Gaussian function is used for the activation function. The RBF network is constructed with three layers: input layer, hidden layer and output layer (Fig. 1). In input layer, the number of neurons is the same with the number of input dimension. The input layer neuron will transmit data to the hidden layer, and calculates a value of the RBFs received from the input layer. These values will be transmitted to the output layer which calculates the values of linear sum of the hidden neuron. In this study, the Gaussian function is used as RBF [10,11,21–28].

Let $\Phi_j(x)$ be the jth radial basis function. $\Phi_j(x)$ is represented as:

$$\Phi_j(x) = \exp\left(-\frac{\left(x - c_j\right)^2}{2\sigma_j^2}\right) \tag{1}$$

Here, $x = (x_1, x_2, \ldots, x_d)^T$ is the input vector, $c_j = (c_{1j}, c_{2j}, \ldots, c_{dj})^T$ and σ_j^2 are the jth center vector and the width parameter, respectively. The output of RBF network y which is the linear sum of radial basis function, is given as follows:

$$y = \sum_{j=1}^{p} w_j \Phi_j(x), \tag{2}$$

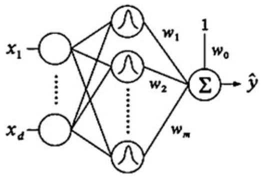

Fig. 1. Structure of RBF network.

where y is the output of the RBF network, p is the number of the hidden layer neuron, and w_j is the weight from jth neuron to the output layer.

To construct RBF network, the number of the hidden layer neuron m must be set, and the centers c_j, the widths σ_j and the weights w_j must be estimated. In RBF typical learning, the network structure will be determined based on prior knowledge or the experiences of experts. The parameters are estimated using either the clustering or the least mean squared method. On the other hand, there are approaches in which the network structure and its parameters are estimated by the evolutionary computation [28,29].

2.2. Multi-Objective Optimization (MOO)

Many real-world problems involve simultaneous optimization of several objective functions. Generally, these functions are non-commensurable and often with conflicting objectives. MOO with such conflicting objective functions gave rise to a set of optimal solutions, instead of one optimal solution. The reason for the optimality of many solutions is that no one can be considered to be better than any other with respect to all objective functions. These optimal solutions are known as Pareto-optimal solutions.

A general MOO problem consists of a number of objectives to be optimized simultaneously and is associated with a number of equality and inequality constraints. It can be formulated as follows:

$$\begin{aligned}&\text{Minimize/Maximize}\, f_i(x) \quad i = 1, \ldots, M\\ &\text{Subject to}: \begin{cases} g_j(x) = 0 & j = 1, \ldots, N \\ h_k(x) \le 0 & k = 1, \ldots, K \end{cases}\end{aligned} \tag{3}$$

where f_i, g_j and h_k are the ith objective function, jth equality and kth inequality constraints respectively, x is a decision vector that represents a solution, and M, N and K are the number of objectives, equality and inequality constraints.

For a MOO problem, any two solutions x_1 and x_2 can have one of the two possibilities—one dominates the other or none dominates the other. In a minimization problem, without loss of generality, a solution x_1 dominates x_2 if the following two conditions are satisfied:

$$\begin{aligned}&\forall i \in \{1, 2, \ldots, M\} : f_i(x_1) \le f_i(x_2)\\ &\exists j \in \{1, 2, \ldots, M\} : f_j(x_1) \prec f_j(x_2)\end{aligned} \tag{4}$$

The operators $\le$ and $\prec$ can be seen as the “less than or equal to” and “less than” operators respectively. The solutions which are not dominated by any other solutions are called the Pareto-optimal solution or non-dominated solution. Generally, many Pareto-optimal solutions exist. The set of Pareto-optimal solutions is called Pareto-optimal front. A non-dominated set is required to be near to the true Pareto front and distributed uniformly.

2.3. Time Variant Multi-Objective Particle Swarm Optimization (TVMOPSO)

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique developed by Kennedy and Eberhart in 1995 [30], inspired by social behavior of bird flocking or fish schooling, in which each individual is treated as an infinitesimal particle in the n-dimensional space, with the position vector and velocity vector of particle i being represented as $X_i(t) = (X_{i1}(t), X_{i2}(t), \ldots, X_{in}(t))$ and $V_i(t) = (V_{i1}(t), V_{i2}(t), \ldots, V_{in}(t))$. The particles move according to the following equations:

$$V_{id}(t+1) = W \times V_{id}(t) + c_1 r_1 (P_{id}(t) - X_{id}(t)) + c_2 r_2 (P_{gd}(t) - X_{id}(t)) \tag{5}$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1), \quad i = 1, 2, \ldots, M; \quad d = 1, 2, \ldots, n, \tag{6}$$

where c_1 and c_2 are the acceleration coefficients, vector $P_i = (P_{i1}, P_{i2}, \ldots, P_{in})$ is the best previous position (the position giving the best fitness value) of particle i known as the personal best position (*pbest*), vector $P_g = (P_{g1}, P_{g2}, \ldots, P_{gn})$ is the best position among the personal best positions of the particles in the population and is known as the global best position (*gbest*).

The parameters r_1 and r_2 are two random numbers distributed uniformly in (0, 1). Generally, the value of V_{id} is restricted in the interval $[-V_{\max}, V_{\max}]$. Inertia weight w, introduced by Shi and Eberhart commonly used to accelerate the convergence speed of the algorithm [31]. A particle swarm algorithm for the solution of Multi-Objective (MO) problems was presented in [32]. In MOPSO, there are many fitness functions, and there is no global best; however, a repository with the non-dominated solutions will be found.

TVMOPSO is adaptive in nature by allowing its vital parameters (inertia weight and acceleration coefficients) to adjust with iterations. This adaptiveness helps the algorithm to explore the search space more efficiently. A diversity parameter has been used to ensure sufficient diversity among the solutions of the non-dominated fronts, while retaining at the same time the convergence to the Pareto-optimal front. TVMOPSO extends the algorithm of the single-objective PSO to handle MOO problems. It incorporates the mechanism of crowding distance computation into the algorithm of PSO, specifically on global best selection, in the deletion method of an external archive of non-dominated solutions and adjusts the parameters of PSO. The crowding distance mechanism together with a mutation operator maintains the diversity of non-dominated solutions in the external archive. The algorithm of TVMOPSO is given in the next subsection.

2.3.1. Main algorithm

The TVMOPSO algorithm is presented below:

Step 1: Generate an initial population P (Population size = N) and velocity for each particle V in a feasible space; evaluate the population and initialize the personal best (*pbest*) and global best (*gbest*) of each particle respectively.
Step 2: Initialize the iteration counter $t = 0$.
Step 3: Store the non-dominated vectors found in P into the archive A (A is the external archive that stores non-dominated solutions found in P).
Step 4: Repeat

a) Compute the crowding distance values of each non-dominated solution in the archive A.
b) Sort the non-dominated solutions in A in descending crowding distance values.
c) Repeat
   i. Randomly select the global best guide for P from a specified top portion of the sorted archive A and store its position to *gbest*.
   ii. Adjust the parameters of PSO (inertia weight and acceleration coefficients) at iteration t. The value of w_t is allowed to decrease linearly with iteration from $w_{\max}$ to $w_{\min}$. The value of inertia weight at iteration t, w_t is obtained as

$$w_t = (w_{\max} - w_{\min}) \frac{\max t - t}{\max t} + w_{\min}, \tag{7}$$

where maxt is the maximum number of iterations and t is the iteration number.

The value of c_1 has been allowed to decrease from its initial value of c_{1i} to c_{1f} while the value of c_2 has been increased from c_{2i} to c_{2f} using the following equations as in [33]. The values of c_{1t} and c_{2t} are evaluated as follows:

$$c_{1t} = (c_{1f} - c_{1i})\frac{t}{\max t} + c_{1i}, \tag{8}$$

$$c_{2t} = (c_{2f} - c_{2i})\frac{t}{\max t} + c_{2i}. \tag{9}$$

iii. Modify each searching point using previous PSO formula and the *gbest*:

$$V_{i+1} = W_t \times V_i + r_1 c_{1t}(pbest_i - P_i) + r_2 c_{2t}(A(gbest) - P_i) \tag{10}$$

$$P_{i+1} = P_i + V_{i+1}, \tag{11}$$

where w_t is inertia weight at iteration t, V_{i+1}, P_{i+1} are new velocity and position of particle i, c_{1t} and c_{2t} are the acceleration coefficients at iteration t, r_1 and r_2 are random numbers in the range [0,1], $pbest_i$ is the best position that the particle i have reached and $A(gbest)$ is the global best guide for each non-dominated solution.

iv. If the current position outside the boundaries, then it takes the upper bound or lower bound, and it is the velocity is multiplied by −1 so that it searches in the opposite direction.

v. If (t < (maxt*Pmut), then perform mutation on P_{i+1}, where maxt is the maximum number of iterations and Pmut is the probability of mutation.

vi. Evaluate the population P_{i+1}.

d) Until the population size is N.

e) Insert all new non-dominated solutions in P into A, if they are not dominated by any of the stored solutions. All dominated solutions in the archive by the new solution are removed from the archive. If the archive is full, then the solution to be replaced is determined by the following steps:

a. Compute the crowding distance values of each non-dominated solution in the archive A.

b. Sort the non-dominated solutions in A in descending crowding distance values.

c. Randomly select a particle from a specified bottom portion which comprises the most crowded particles in the archive then replace it with the new solution.

f) Modify the *pbest* of each particle in P. If the current position dominates the *pbest*, replace the *pbest* with current position; otherwise keep the previous *pbest*.

g) Increment iteration counter t.

Step 5: Until maximum number of iterations is reached.

The main differences of the proposed approach with respect to the other proposals existing in the literature [35,36] are:

1. Selection of *gbest*: Based on the crowding distances' values, the non-dominated solution with the highest crowding distance values is done to pick a solution as the *gbest* instead of roulette wheel selection.
2. Computation of the diversity: The diversity measurement has been done using crowding distance measure.
3. Updating of the archive: whenever the archive is full, crowding distance is used in selecting which solution to be replaced in the archive. This promotes diversity among the stored solutions in the archive since those solutions which are in the most crowded areas are most likely to be replaced by a new solution.

These differences are similar to [37]; however, we adjust PSO parameters accordingly (w_t: the inertia coefficient, c_{1t}: the local acceleration coefficient and c_{2t}: the global acceleration coefficient) in each iteration to ensure better search to improve the convergence to Pareto-optimal solutions.

2.3.2. *External archive*

As archive stores non-dominated solutions found in the previous iteration, any one of the solutions can be used as global guide. However to ensure that the particles in the population move towards the sparse regions of the non-dominated solutions and speed up the convergence towards the true Pareto-optimal region, the global best guide of the particles is selected from a restricted variable size archive. This restriction on archive is done using the crowding distance operator. This operator will ensure that those non-dominated solutions with the highest crowding distance values are always preferred to be in the external archive. The other advantage of this variable size external archive is that it saves considerable computational time during optimization. As the archive size increases, the computing requirement becomes greater for sorting and crowding value calculations. Selecting different guides for each particle from a restricted archive allows the particles to explore better to the true Pareto-optimal region, hence, improve improves the performance of the algorithm effectively.

2.3.3. *Crowding distance assignment operator*

In order to improve the diversity in the Pareto-optimal solutions, a crowding distance operator is adopted from [34]. Unlike [35], this operator for measuring diversity has an advantage that it needs no parameter specification. The crowding distance value of a solution provides density estimation of surrounding solutions. Crowding distance is calculated by sorting the set of solutions in ascending objective function values. The crowding distance value of a particular solution is the average distance of its two neighboring solutions. The boundary solutions that have the lowest and highest objective function values are given infinite crowding distance values; hence, they are always selected. This process is done for each objective function. The final crowding distance value of a solution is computed by adding all the individual crowding distance values in each objective function. For sorting, an efficient quick sorting procedure is used. The pseudo-code of crowding distance computation is given below.

```
1. Get the number of non-dominated solutions in the external archive $A$
      $n = |A|$

2. Initialize distance
      For $i = 1$ to $n$
         $A_i$.dist = 0

3. Compute the crowding distance of each solution
      For each objective $m$,
      Sort using each objective value.
           $A = \mathrm{sort}(A, m)$
      Set the maximum distance to the boundary points so that they are
         always selected
           $A_1$.distance = $S_n$.distance = maximum distance
      For $i = 2$ to $(n - 1)$
           $A_i.\mathrm{dist} = A_i.\mathrm{dist} + (A_{i+1}.\mathrm{dist} - A_{i-1}.\mathrm{dist})/(f_m^{\max} - f_m^{\min})$
```

2.3.4. *Mutation operator*

The single-objective PSO algorithms have been found to show good convergence properties. However, for the multi-objective PSOs, this convergence is usually achieved at the cost of the diversity [35]. To allow the multi-objective PSO algorithm to explore the search space to a greater extent, while obtaining better diversity, a mutation operator has been used in TVMOPSO. This is helpful in terms of preventing premature convergence due to existing local Pareto fronts in some optimization problem. The pseudo-code of mutation operator is given below.

```
% particle = particle to be mutated
% dims = number of dimensions
% t = current iteration
% maxt = total number of iterations
% Pmut = mutation rate
```

```
function Mutation-Operator(particle,dims,t,maxt,Pmut)
 begin
  if flip($(1 - t/maxt)^{5/Pmut}$) then
  begin
    Wichdim = random (0, dims-1)
    Mutrange = (upperbound[wichdim]−lowerbound[wichdim])*
      $(1-t/maxt)^{5/Pmut}$
    ub = particle[wichdim]+mutrange
    lb = particle[wichdim]−mutrange
    if lb < lowerbound[wichdim] then lb=lowerbound[wichdim]
    if ub > upperbound[wichdim]then ub=upperbound [wichdim]
    particle[wichdim]=RealRandom(lb,ub)
  end if
end function
```

3. The RBFN–TVMOPSO algorithm

3.1. RBFN representation

The RBF network is represented as a vector with dimension D containing the connections (centers and weights) of the network. The dimension of a particle is:

$$D = (I \times H) + (H \times O) + H + O, \qquad (12)$$

where I, H and O are referred the number of input, hidden and output neurons of RBF network respectively. The centers of RBF network in hidden layer are initialized from k-means clustering algorithm and the weights of RBF network are initialized from the Least Mean Squared (LMS) algorithm or random values. The number of input and output neurons is problem-specific and there is no exact way of knowing the best number of hidden neurons. However, there are rules-of-thumb [38] to obtain this value. The number of hidden neurons (RBFs) is depended on the number of classes of the problem to be solved. In this paper, a particle represents a one-hidden layer of RBF network and the swarm consists of a population of one-hidden layer of networks.

3.2. RBFN-TVMOPSO implementation

The proposed approach is a multi-objective optimization approach to RBF network training with TVMOPSO as the multi-objective optimizer (RBFN-TVMOPSO). The algorithm will simultaneously determine the set of connection (centers of RBF, weights) and its corresponding accuracy by treating this problem as a multi-objective minimization problem.

RBFN-TVMOPSO starts by collecting, normalizing and reading the data set. This is followed by setting the desired number of hidden neurons and the maximum number of generations for TVMOPSO. The next step is to determine the dimension of the particles and to initialize the population with feed-forward RBF network. In each generation, every particle is evaluated based on the two objective functions (Eqs. (13) and (14)). Subsequently, the algorithm outputs a set of non-dominated Pareto RBF networks (Fig. 2).

3.3. Objective functions

Two objective functions are used to evaluate the RBF network particle's performance. The two objective functions for minimization problems are:

1. Accuracy based on Mean Square Error (MSE) on the training set.

$$f_1 = \frac{1}{N}\sum_{j=1}^{N}\left(t_j - o_j\right)^2, \qquad (13)$$

   where o_j and t_j are the network output and the desired output, and N is the number of samples.
2. Complexity is computed based on the sum of the squared weights, and it is a concept of regularization and represents the smoothness of the model [17,20,41]. The equation is given as:

$$f_2 = \frac{1}{2}\sum_{j=1}^{Q} w_j^2, \qquad (14)$$

   where w_j, $j = 1,\ldots,Q$ is a weight in the network, and Q is the number of weights in total.

4. Experimental studies

Several experiments were conducted on three real-world medical data sets listed in Table 1 to evaluate the performance of RBFN-TVMOPSO. All data sets have been loaded from the University of California at Irvine (UCI) machine learning benchmark repository [39]. These problems have been the subjects of many studies in ANNs and machine learning.

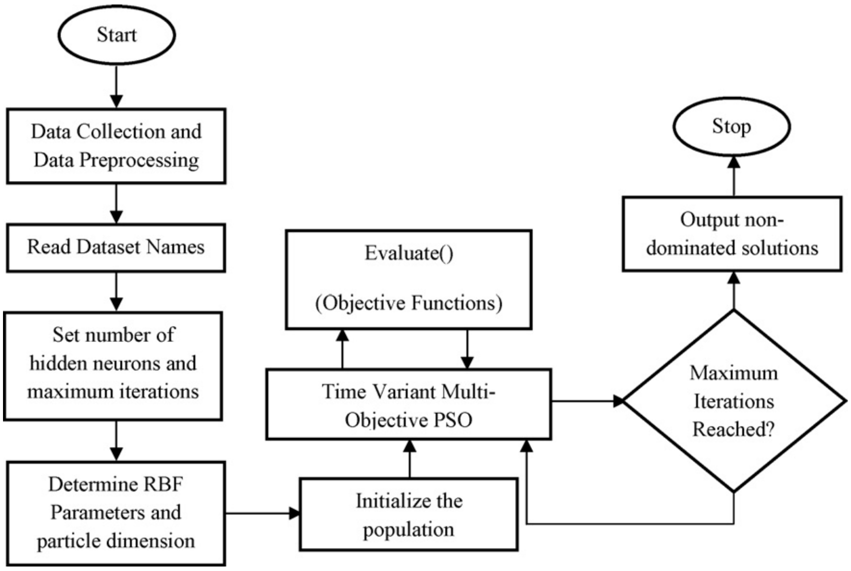
Start
Data Collection and Data Preprocessing
Read Dataset Names
Set number of hidden neurons and maximum iterations
Determine RBF Parameters and particle dimension
Initialize the population
Time Variant Multi-Objective PSO
Evaluate()
(Objective Functions)
Maximum Iterations Reached?
Output non-dominated solutions
Stop

Fig. 2. RBFN-TVMOPSO procedure.

Table 1
Description of data sets.

Data Set	Attributes	Classes	Samples	Input	Hidden	Output
Breast Cancer	9	2	699	9	2	1
Diabetes	8	2	768	8	2	1
Hepatitis	19	2	155	19	2	1

4.1. Data sets design

Data sets from the medical field are being used to validate the proposed algorithm. The cancer, diabetes and hepatitis data sets represent binary class classification problems.

The objective of the cancer problem is to diagnose breast cancer in patients by classifying a tumor as benign or malignant. The “Breast Cancer Wisconsin” problem data set was originally collected in the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg [42]. 458 (65.5%) of the patterns in the data sets are benign while 241 (34.5%) of the patterns are malignant. There are nine attributes/inputs (clump thickness, uniformity of cell size and shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses) and two output classes (benign or malignant).

The diabetes problem is to diagnose a Pima Indian individual based on personal data and medical examination. There are eight attributes/inputs (no. of times pregnant, plasma glucose concentration, diastolic blood pressure, triceps skin fold thickness, serum insulin, BMI and age) and two output classes (diabetes positive or diabetes negative).

The hepatitis problem is a complex and noisy data as it contains a large number of missing data (there are 167 missing values in total in this data set). The learning task is to predict whether a patient with hepatitis will live or die. There are nineteen attributes/inputs (age, sex, steroid, antivirals, fatigue, malaise, anorexia, liver big, liver film, spleen palpable, spiders, ascites, varices, bilirubin, alk. phosphate, SGOT, albumin, protime and histology) and two output classes (live or die).

4.2. Experimental setup

All data sets in this study are partitioned into three sets: a training set, a validation set and a testing set. The training set is used to train the network in order to get the Pareto-optimal solutions. The validation set is used to select the best one from the Pareto-optimal solutions, while the testing set is used to test the generalization performance of Pareto RBF network. It is known that the results may vary significantly for different partitions of the same data set. For each data set, we analyze the evolutionary process of RBFN-TVMOPSO and evaluate the performance of it on the breast cancer, diabetes and hepatitis data sets. These data are partitioned randomly into three. For each data set, 50% of data were used for the training set, 25% of data for the validation set and the rest 25% for the testing set. Values of the data sets are normalized in the range of [0,1].

Table 2
Parameters settings for RBFN-TVMOPSO.

Parameter	RBFN-TVMOPSO
Optimization type	Minimization
Population size	100
Archive size	100
Objective functions	2
Constraints	0
Lower limit of variable	−0.5
Upper limit of variable	0.5
Probability of mutation (pM)	0.5

For each data set, the experiments were implemented to minimize the influence of random effects. In addition, the number of input and output nodes is problem-dependent but the number of hidden nodes is the number of classes of data set. The structure of RBF network in these data sets is shown in Table 1. The number of iterations is the same for all data sets (1000) except for the breast cancer data set (500). There are some parameters in RBFN-TVMOPSO, which need to be specified by the user. Therefore, these parameters were set equivalently for all data: the inertia weights $w_{\max}$ and $w_{\min}$ (0.7 and 0.4), the initial acceleration coefficients c_1 and c_2 (2.5 and 0.5) [36]. c_1 has been allowed to decrease from its initial value of 2.5 to 0.5, while c_2 has been increased from 0.5 to 2.5. The other various parameters' settings of RBFN-TVMOPSO are presented in Table 2.

4.3. Results and discussions

This section presents the results of study on TVMOPSO based RBF network. The experiments are conducted using three data sets. The results for each data set are compared to RBF network based on MOPSO [35] and NSGA-II [34] algorithms and analyzed based on the convergence to Pareto-optimal set with their classification performance. One advantage of evolutionary multi-objective optimization approach to RBF network generation is that a number of

Table 3
Results of RBFN-TVMOPSO, RBFN-NSGA-II and RBFN-MOPSO for all data sets.

Data set	RBFN-TVMOPSO			RBFN-MOPSO			RBFN-NSGA-II		
	Training error	Validation error	Testing error	Training error	Validation error	Testing error	Training error	Validation error	Testing error
Breast cancer									
Mean	0.0241	0.0251	**0.0267**	0.0747	0.0724	0.0767	0.1049	0.0998	0.1052
SD	0.0307	0.0299	**0.0299**	0.0653	0.0620	0.0645	0.0664	0.0664	0.0666
Min	0.0142	0.0157	0.0149	0.0133	0.0145	0.0121	0.0199	0.0187	0.0203
Max	0.2854	0.2912	0.2834	0.3201	0.3312	0.3163	0.2513	0.2517	0.2512
Diabetes									
Mean	0.1786	0.1730	**0.1591**	0.2050	0.1994	0.1928	0.2061	0.1984	0.1942
SD	0.0136	0.0127	**0.0167**	0.0234	0.0227	0.0294	0.0195	0.0220	0.0250
Min	0.1592	0.1631	0.1411	0.1605	0.1656	0.1470	0.1831	0.1733	0.1629
Max	0.2556	0.2570	0.2567	0.2617	0.2645	0.2636	0.2506	0.2507	0.2507
Hepatitis									
Mean	0.1464	0.1970	**0.1226**	0.1774	0.1809	0.1472	0.1856	0.1866	0.1637
SD	0.0153	0.0282	**0.0209**	0.0253	0.0236	0.0368	0.0302	0.0276	0.0389
Min	0.1201	0.1522	0.1034	0.1434	0.1527	0.1033	0.1458	0.1576	0.1202
Max	0.2526	0.2750	0.2644	0.2725	0.3268	0.3436	0.2505	0.2505	0.2506

The best results are highlighted in bold.

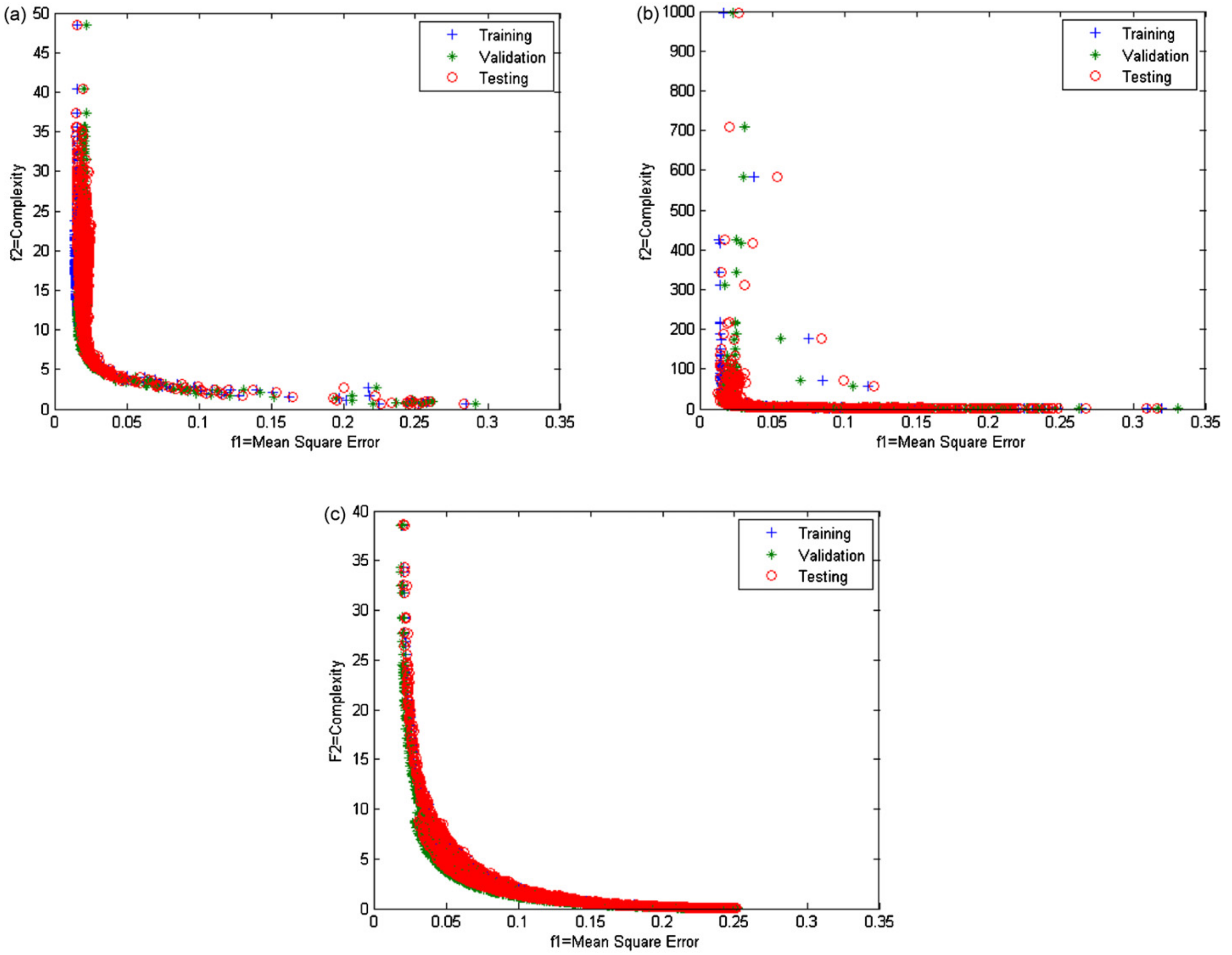

Fig. 3. Pareto front of (a) RBFN-TVMOPSO, (b) RBFN-MOPSO and (c) RBFN-NSGA-II on breast cancer data set.

RBF networks with a spectrum of complexity can be obtained in one single run.

Table 3 shows the statistical results of RBF network based on TVMOPSO, MOPSO and NSGA-II algorithms over 10 independent runs on breast cancer, diabetes and hepatitis data sets (the mean, SD, Max and Min indicate the mean value, standard deviation, maximum value and minimum value, respectively). The result of these algorithms is Pareto-optimal solutions to improve the generalization on unseen data. We report the results in terms of average error rates over 10 independent runs on the three data sets. These results demonstrate that TVMOPSO has the capability to evolve compact RBF networks which generalize well on unseen data. As illustrated in Table 3, testing error values signify that RBFN-TVMOPSO has resulted in better convergence, compared to other algorithms for all data sets. It is interesting to see that small standard deviations are obtained for the testing error in all data sets. These indicate the consistency, stability and accuracy of our proposed method.

In order to evaluate the performance of RBFN-TVMOPSO for generalization and classification, the comparison was carried out using RBFN-MOPSO and RBFN-NSGA-II. Figs. 3–5 show the results of non-dominated fronts of all algorithms on training, validation and testing corresponding to these data sets. Fig. 3 provides the non-dominated solutions returned by three algorithms for the breast cancer data set. The best performance of RBFN-TVMOPSO is clearly illustrated. Fig. 3a shows that RBFN-TVMOPSO is better than RBFN-MOPSO and RBFN-NSGA-II in terms of convergence to Pareto front solutions, while RBFN-TVMOPSO maintains diversity of the Pareto front solutions. The poor performance of RBFN-MOPSO and RBFN-NSGA-II is shown in Fig. 3b and c. From Table 3 and Fig. 3, RBFN-MOPSO and RBFN-NSGA-II have failed to converge to the true Pareto front properly. From Fig. 3b, the highest value of function $f2$ (complexity) is found to be 1000, whereas for the results obtained by RBFN-TVMOPSO and RBFN-NSGA-II, this value is set to 50 and 40 respectively.

Similarly, Fig. 4 represents the final fronts obtained by three algorithms for the diabetes data set. It can be seen that RBFN-MOPSO and RBFN-NSGA-II have failed to converge to the true Pareto front properly. The highest value of function $f2$ (complexity) for RBFN-MOPSO is found to be 800, whereas for the results obtained by RBN-TVMOPSO and RBFN-NSGA-II are set to 30 and 10 respectively (Table 3). Among these algorithms, RBFN-TVMOPSO has given better convergence, spread and distributed of solutions on this data set. However, RBFN-NSGA-II in Fig. 4c has good solutions diversity. The comparative performance of the above algorithms in terms of their performance in hepatitis data set can be seen in Fig. 5 which shows the final non-dominated fronts obtained by all the algorithms. From Fig. 5b and c and Table 3, it can be seen that RBFN-MOPSO and RBFN-NSGA-II have failed considerably

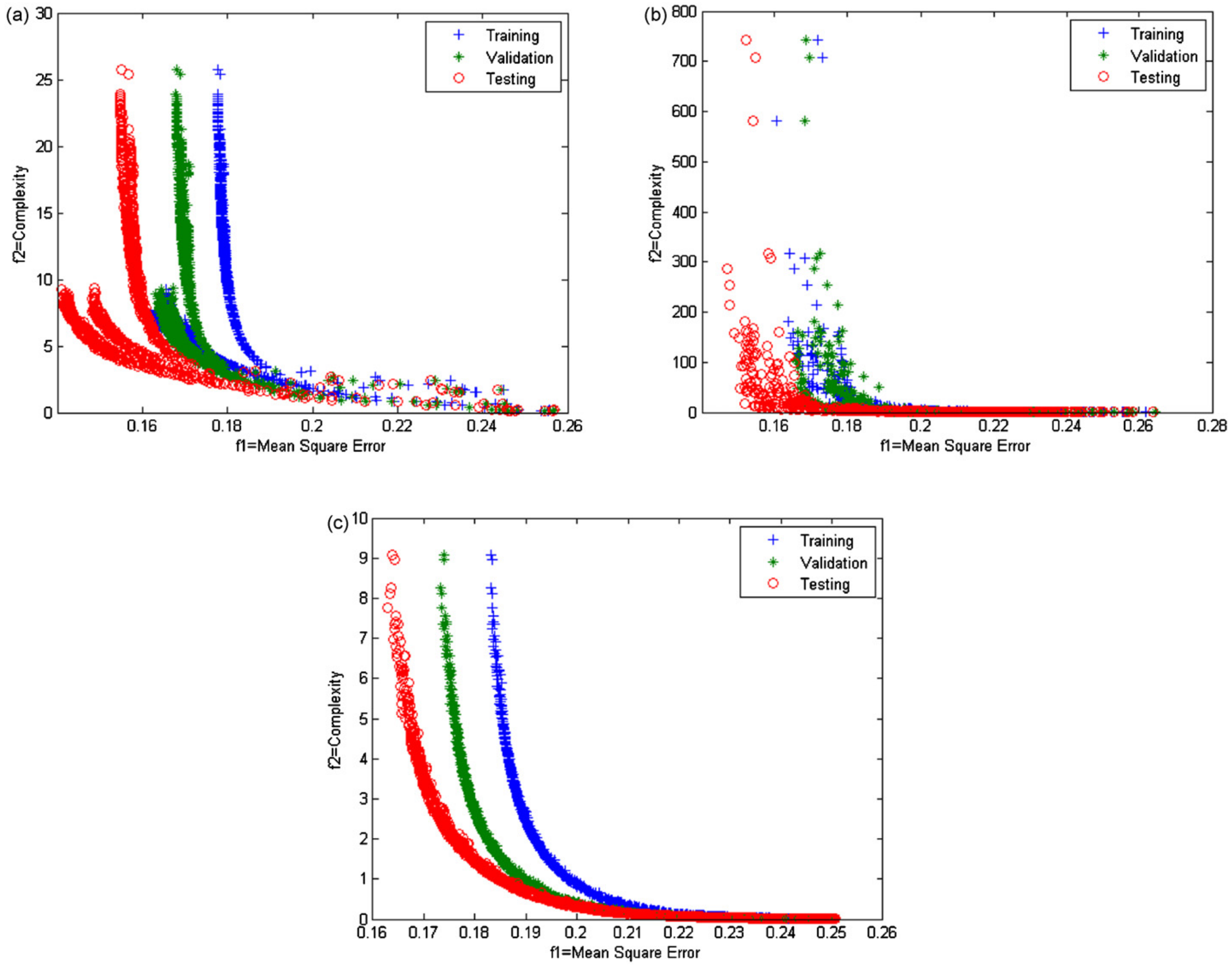

Fig. 4. Pareto front of (a) RBFN-TVMOPSO, (b) RBFN-MOPSO and (c) RBFN-NSGA-II on diabetes data set.

in attaining the non-dominated set properly in terms of convergence. In addition, RBFN-TVMOPSO in Fig. 5a provides good spread and distributed solution compared to RBFN-MOPSO for this data set.

From these comparisons, it shows that RBFN-TVMOPSO is better than RBFN-MOPSO and RBFN-NSGA-II in terms of convergence to Pareto front solutions. RBFN-TVMOPSO maintains diversity of the Pareto front solutions for all data sets. The average number of Pareto-optimal solutions over 10 independent runs for each data set is obtained by all methods (Table 4). It can be observed that the proposed RBFN-TVMOPSO provides higher number of Pareto-optimal solutions than RBFN-MOPSO. However, the proposed algorithm has maintained diversity and has competitive Pareto-optimal solutions compared to RBFN-NSGA-II for all data sets.

Table 4
Average of number of Pareto-optimal solutions in set of non-dominated solutions for all data sets.

Data set	RBFN-TVMOPSO	RBFN-MOPSO	RBFN-NSGA-II
Breast cancer	98	54	100
Diabetes	96	50	100
Hepatitis	97	71	100

When we minimize both accuracy (error) and complexity of the network in a Pareto-based approach, we are able to achieve a number of Pareto-optimal solutions with a complexity ranging from simple networks to highly complex ones. From all data sets, we can conclude that by trading off accuracy against complexity, the Pareto-based multi-objective optimization algorithm is able to find the simplest structures that solve the problem best. Furthermore, the simple Pareto-optimal networks are able to generalize well on unseen data. From Figs. 3–5, we can see that when the network complexity is increasing, the testing error is decreasing. This phenomenon can be observed from the results by all selected testing approaches. However, this phenomenon is only partially maintained for the relationship between the test accuracy and the network complexity. Test error is still decreasing as the network complexity is increasing. From these figures, it is very difficult to find a single optimal network that can offer the best accuracy for all data sets. Therefore, instead of searching for a single optimal RBF network, an algorithm that can result in a complete set of near optimal networks would be more reasonable for real applications.

Table 5 shows the statistical results for classification accuracy of RBF network based on TVMOPSO, MOPSO and NSGA-II over 10 independent runs on breast cancer, diabetes, and hepatitis data sets. The accuracy refers to the percentage of correct classification on training, validation and testing data sets respectively. Fig. 6

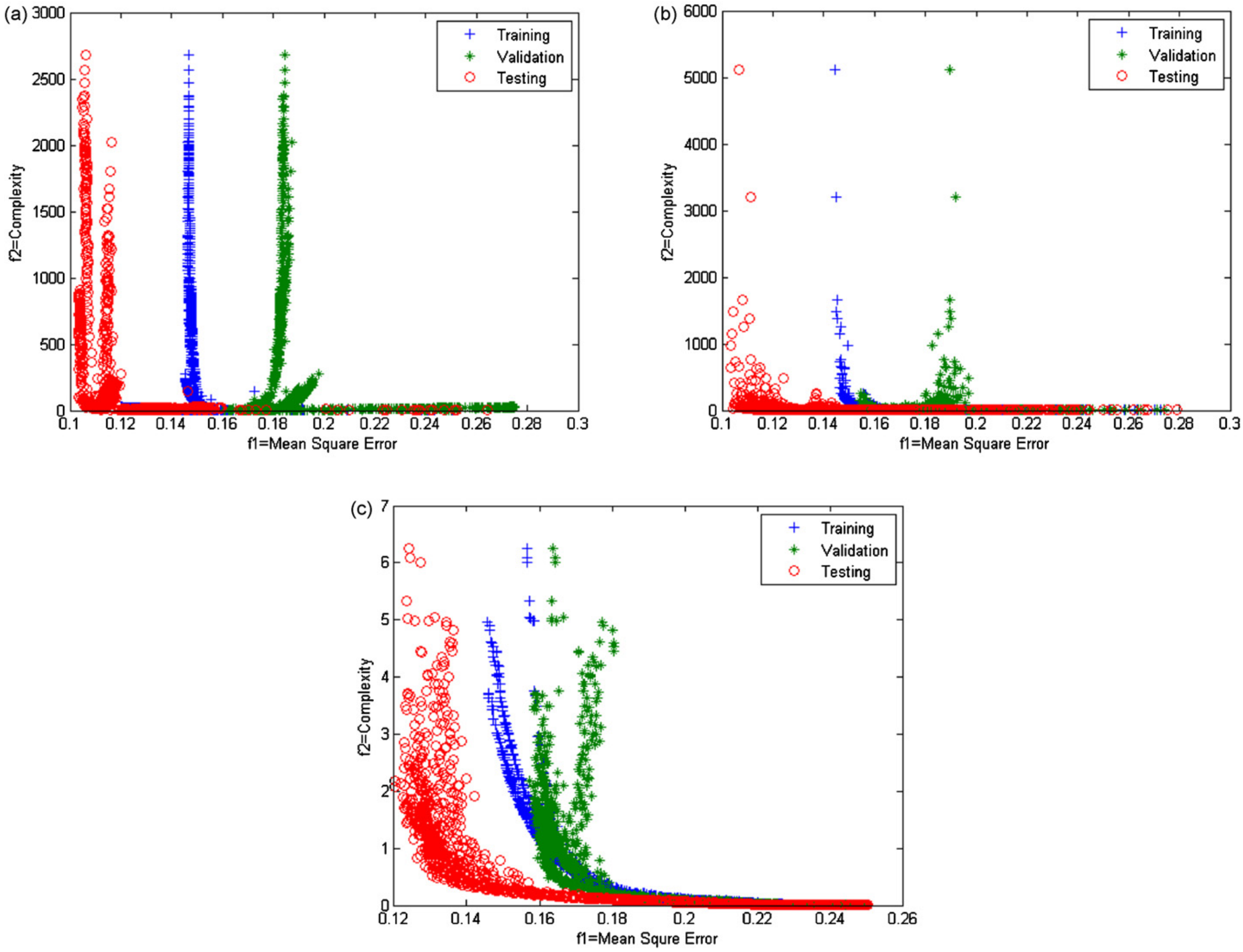

Fig. 5. Pareto front of (a) RBFN-TVMOPSO, (b) RBFN-MOPSO and (c) RBFN-NSGA-II on hepatitis data set.

shows correct classification of Pareto front solutions for testing phases which are produced from RBFN-TVMOPSO, RBFN-MOPSO and RBFN-NSGA-II. Table 5 and Fig. 6 show also that RBF networks based on TVMOPSO have higher percentages of accuracy compared to the other algorithms on breast cancer and diabetes data sets. However, RBFN-NSGA-II has the highest classification accuracy for Hepatitis data set compared to other algorithms. RBFN-TVMOPSO provides best results and maintains diversity of Pareto fronts solu-

Table 5
Results of RBFN-TVMOPSO, RBFN-NSGA-II and RBFN-MOPSO for all data sets.

Data set	RBFN-TVMOPSO			RBFN-MOPSO			RBFN-NSGA-II		
	Training accuracy	Validation accuracy	Testing accuracy	Training accuracy	Validation accuracy	Testing accuracy	Training accuracy	Validation accuracy	Testing accuracy
Breast cancer									
Mean	97.24	96.46	**96.53**	91.76	92.05	90.95	87.86	89.11	87.30
SD	6.35	6.42	**6.19**	11.54	10.12	12.09	14.21	12.59	14.83
Min	10.82	11.70	10.00	33.04	28.66	28.82	33.04	28.66	34.71
Max	98.83	98.25	98.24	98.83	98.25	98.82	97.95	97.66	98.24
Diabetes									
Mean	72.87	73.41	**78.02**	67.09	68.91	70.58	66.28	68.59	69.59
SD	3.39	2.93	**3.95**	5.77	5.13	6.09	4.41	4.02	4.98
Min	36.98	33.85	33.85	36.98	32.81	33.85	36.98	33.85	33.85
Max	78.13	77.60	82.81	77.34	76.56	81.25	74.22	73.96	77.60
Hepatitis									
Mean	78.19	72.54	82.26	74.12	75.61	82.32	75.82	76.36	**83.78**
SD	4.79	4.80	3.85	6.96	7.62	9.00	3.08	3.34	**3.81**
Min	25.64	23.08	18.42	24.36	23.08	13.16	24.36	23.08	15.79
Max	88.46	82.05	86.84	83.33	84.62	86.84	83.33	79.49	84.21

The best results are highlighted in bold.

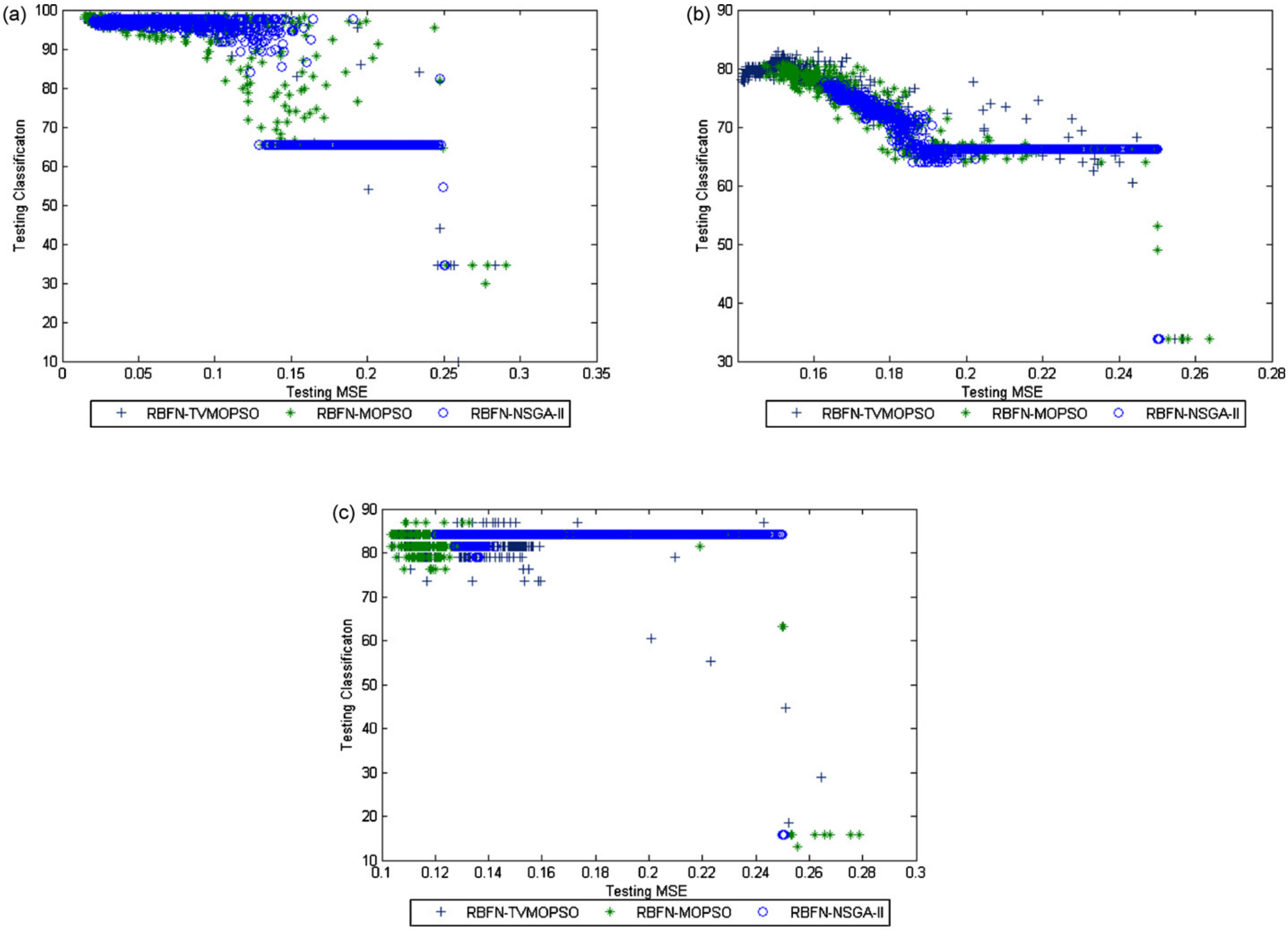

Fig. 6. Correct classification of Pareto front of testing data for (a) breast cancer, (b) diabetes and (c) hepatitis.

Table 6
Results of RBFN-TVMOPSO, RBFN-NSGA-II and RBFN-MOPSO for all data sets.

Data set	RBFN-TVMOPSO			RBFN-MOPSO			RBFN-NSGA-II		
	Sensitivity	Specificity	AUC	Sensitivity	Specificity	AUC	Sensitivity	Specificity	AUC
Breast cancer	97.25	96.20	**0.967**	91.26	95.83	0.936	88.24	94.77	0.915
Diabetes	83.09	68.54	**0.758**	72.61	64.44	0.685	72.14	60.14	0.661
Hepatitis	88.47	41.92	**0.652**	86.18	43.67	0.649	84.17	0.59	0.424

The best results are highlighted in bold.

tions. It is interesting to see that the small standard deviations are obtained for the testing classification accuracy in breast cancer and diabetes data sets. This indicates the consistency, stability and accuracy of the proposed method.

In order to evaluate the classification capabilities of RBFN-TVMOPSO, RBFN-MOPSO and RBFN-NSGA-II, a comparison of these three algorithms and average of sensitivity, specificity, Area Under Curve (AUC) of ROC analysis was performed as depicted in Table 6. The Receiver Operating Characteristic (ROC) curve is a graphical display that gives the measure of the predictive accuracy of a logistic model. The curve displays the true positive rate (sensitivity) and false positive rate (1-specificity). Sensitivity is a measure of accuracy for predicting events that is equal to the true positive/total actual positive. Specificity is a measure of accuracy for predicting nonevents that is equal to the true negative/total actual negative of a classifier for a range of cutoffs. Fig. 7 displays the ROC curve for all algorithms.

4.4. Comparisons with other work

Having validated the effectiveness of multi-objective optimization, the performance of the proposed method is compared against RBFN-MOPSO, RBFN-NSGA-II and other methods in the literature

Table 7
Comparison of classification accuracy of our proposed approach with the previous methods.

Method/reference	Breast cancer	Diabetes	Hepatitis
RBFN-TVMOPSO	96.53	78.02	82.26
RBFN-MOPSO	90.95	70.58	82.32
RBFN-NSGA-II	87.30	69.59	83.78
HMOEN_L2 [40]	96.30	78.50	80.30
HMOEN_HN [40]	96.82	75.36	75.51
MPANN [15]	98.10	74.90	–
MOBNET [8]	–	77.85	–

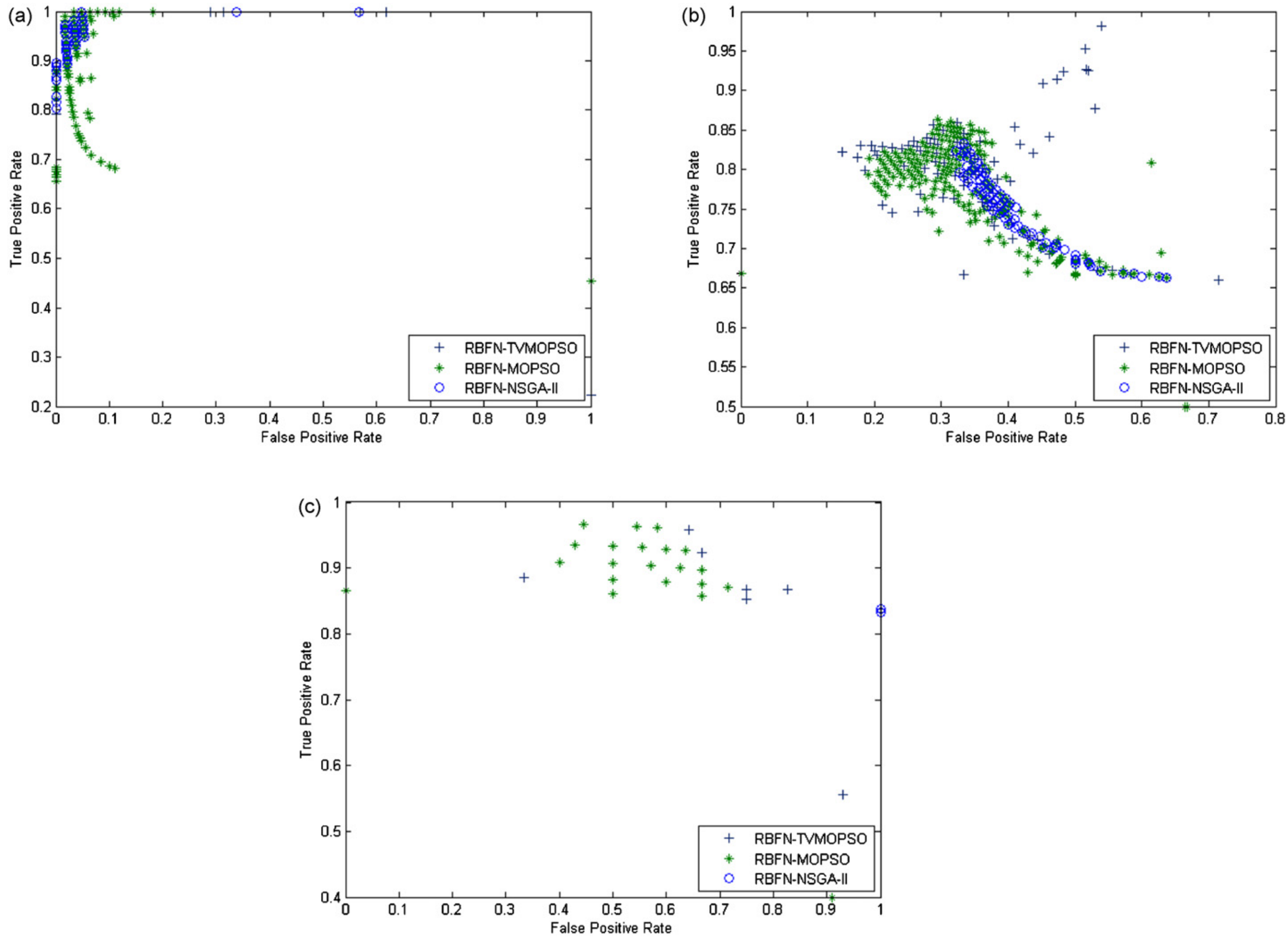

Fig. 7. ROC curve of Pareto front of testing data for (a) breast cancer, (b) diabetes and (c) hepatitis.

using these data sets. The summary of the results is shown in Table 7 and Fig. 8. The results that are presented here are not fine-tuned in any manner, i.e., the same parameter and experimental settings are used for all data sets. Nonetheless, it can be observed that the proposed method is at least competitive for diabetes and hepatitis data sets. Breast cancer results are outperformed by MPANN [15]. On the other hand, MPANN performs poorly for diabetes data set with respect to our proposed RBFN-TVMOPSO. MPANN reported the average Pareto-optimal front of the ANN with the lowest classification error for the breast cancer and diabetes data sets. The HMOEN_L2 [40] and HMOEN_HN [40] results are based on the ANN with the best training accuracy on the data set for each run. However, our proposed method, RBFN-MOPSO and RBFN-NSGA-II reported that average of all Pareto-optimal fronts over 10 independent runs for all data sets.

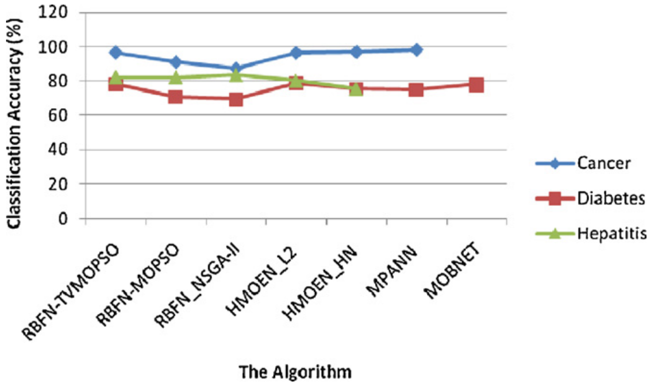

Fig. 8. Comparison of proposed algorithm against existing methods for breast cancer, diabetes and hepatitis data sets.

5. Conclusions

In this paper, an approach for multi-objective optimization based on swarm intelligence principles, called TVMOPSO, is proposed and applied to develop generalization and classification accuracy for multi-objective RBF network. TVMOPSO is adaptive in nature with respect to its inertia weight and acceleration coefficients.

This paper introduces time variant multi-objective PSO approach to RBF network design called time variant multi-objective PSO-RBF network Optimizer to concurrently optimize the accuracy and connections (centers and weights) of network. RBF network and its parameters are encoded to the particle, and Pareto-optimal set of RBF network is obtained by TVMOPSO based on two criteria, i.e. model accuracy and model complexity. The benchmark of medical diagnosis indicates that our proposed method provides better or comparable results to RBF network based on MOPSO and NSGA-II in terms of giving a wide spread of diverse solutions with good convergence to true Pareto-optimal fronts.

The main advantages of TVMOPSO are that it is simple algorithm, easy to implement, easy to use, and yet robust in yielding efficient Pareto frontiers. TVMOPSO converges fast to the true Pareto-optimal RBF network, and at same time maintains good diversity along the Pareto front. At this point, the proposed TVMOPSO significantly outperforms MOPSO and NSGA-II for RBF network and competitive with previous methods. Hence it can be concluded that, for RBF networks, the proposed technique is a viable tool for multi-objective analysis.

Acknowledgments

This work is supported by Ministry of Higher Education (MOHE) under Fundamental Research Grant Scheme. The authors would like to thank Research Management Centre (RMC) Universiti Teknologi Malaysia, for the research activities and *Soft Computing Research Group* (SCRG) for the support and incisive comments in making this study a success.

References

[1] D. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Systems 2 (1988) 321–355.
[2] J.A. Leonard, M.A. Kramer, Radial basis function networks for classifying process faults, Control Systems Magazine 11 (3) (1991) 31–38.
[3] X. Yao, Evolving artificial neural networks, Proceedings IEEE 87 (9) (1999) 1423–1447.
[4] J. Fieldsend, S. Singh, Pareto evolutionary neural networks, IEEE Transaction Neural Networks 16 (2) (2005) 338–354.
[5] C.A. Coello Coello, An updated survey of GA-based multi-objective optimization techniques, ACM Computing Surveys 32 (2) (2000) 109–143.
[6] C.A. Ceollo Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Multi-Objective Problems, Kluwer, New York, 2002.
[7] N. Garcia-Perdrajas, E. Sanz-tapia, D. Ortiz-Boyer, C. Hervas, Introducing multi-objective optimization in cooperative coevolution of neural networks, in: Proc. 6th Int. Work-Conference. Artificial Natural Neural Network, 2001, pp. 645–652.
[8] N. Garcia-Perdrajas, C. Hervas, J. Munoz-Perez, Multi-objective cooperative coevolution of artificial neural networks, Neural Networks 15 (10) (2002) 1259–1278.
[9] G. Liu, V. Kadirkamanathan, Multi-objective criteria for neural networks structure selection and identification of nonlinear systems using genetic algorithms, Proceedings Institute of Electrical Engineering: Control Theory and Applications 146 (5) (1999) 373–382.
[10] E. de Lacerda, A. de Carvallho, T. Ludermir, Evolutionary optimization of RBF networks, in: In Proc. 6th Braz. Symp. Neural Netw., 2000, pp. 219–224.
[11] J. Gonzalez, I. Rojas, J. Ortega, H. Pomares, F. Fernandez, A. Diaz, Multi-objective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation, IEEE Transactions on Neural Network 14 (6) (2003) 1478–1495.
[12] H.A. Abbass, R. Sarker, Simultaneous evolution of architectures and connection weights in ANNs, in: Proc. ANNES Conf., 2001, pp. 16–21.
[13] H.A. Abbass, A memetic pareto evolutionary approach to artificial neural networks, in: Australian Joint Conference on Artificial Intelligence, 2001, pp. 1–12.
[14] H.A. Abbass, An evolutionary artificial neural networks approach for breast cancer diagnosis, Artificial Intelligence in Medicine 25 (3) (2002) 265–281.
[15] H.A. Abbass, Speed up backpropagation using multi-objective evolutionary algorithms, Neural Computation 15 (11) (2003) 2705–2726.
[16] H.A. Abbass, Pareto neuro-evolution: constructing ensemble of neural networks using multi-objective optimization, in: Proc. CEC, vol. 3, 2003, pp. 2074–2080.
[17] Y. Jin, T. Okabe, B. Sendhoff, Neural network regularization and ensembling using multi-objective evolutionary algorithms, in: Proc. Cong. Evol. Comput., 2004, pp. 1–8.
[18] Y. Jin, B. Sendhoff, Alleviating catastrophic forgetting via multi-objective learning, in: Proc. Int. Joint Conference Neural Netw., 2006, pp. 3335–3342.
[19] L. Graning, Y. Jin, B. Sendhoff, Generalization improvement in multi-objective learning, in: Proc. Int. Joint Conf. Neural Netw., 2006, pp. 4839–4846.
[20] Y. Jin, B. Sendhoff, E. Korner, Evolutionary Multi-Objective Optimization for Simultaneous Generation of Signal-Type and Symbol-Type Representations, vol. 3410, Springer-Verlag, Berlin, Germany, 2005, pp. 752–766.
[21] I. Kokshenev, A.P. Braga, A multi-objective approach to RBF network learning, Neurocomputing 71 (2008) 203–1209.
[22] G.G. Yen, Multi-Objective Evolutionary Algorithm for Radial Basis Function Neural Network Design, 16, Springer-Verlag, 2006, pp. 221–239.
[23] N. Kondo, T. Hatanaka, K. Uosaki, Pattern classification by evolutionary RBF networks ensemble based on multi-objective optimization, in: International Joint Conference on Neural Networks, 2006, pp. 2919–2925.
[24] V. Lefort, C. Knibbe, G. Beslon, J. Favrel, Simultaneous Optimization of Weights and Structure of an RBF Neural Network, vol. 3871, Springer-Verlag, LNCS, 2006, pp. 49–60.
[25] J. Gonzalez, I.P.H. Rojas, J. Ortega, RBF Neural Networks, Multi-objective Optimization and Time Series Forecasting, vol. 2084, Springer-Verlag, LNCS, 2001, pp. 498–505.
[26] N. Kondo, T. Hatanaka, K. Uosaki, Nonlinear dynamic system identification based on multi-objectively selected RBF networks, in: Proceedings of the IEEE Symposium on Computational Intelligence in Multicriteria Decision Making, 2007, pp. 112–127.
[27] P.M. Ferreira, A.E. Ruano, C.M. Fonseca, Evolutionary multi-objective design of radial basis function networks for greenhouse environmental control, in: Proc. of the 16th IFAC World Congress, 2005.
[28] Y. Bai, L. Zhang, Genetic algorithm based self-growing training for RBF neural networks, in: Proceedings of International Joint Conference on Neural Networks, 2002, pp. 840–845.
[29] Z. Wenbo, H. De-Shuang, Y. Ge, The structure optimization of radial basis probabilistic neural networks based on genetic algorithms, in: Proceedings of International Joint Conference on Neural Networks, 2002, pp. 1086–1091.
[30] J. Kennedy, R.C. Eberhart, Particle Swarm Optimization, in: Proceedings of IEEE International Conference on Neural Networks, IV, Piscataway, 1995, pp. 1942–1948.
[31] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: IEEE Conference on Evolutionary Computation, 1998, pp. 69–73.
[32] C.A.C. Coello, M.S. Lechuga, MOPSO: a proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation part of the 2002 IEEE World Congress on Computational Intelligence, IEEE Press, Hawaii, 2002, pp. 1051–1056.
[33] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 240–255.
[34] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A. Fast, Elitist Multi-objective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
[35] C. Coello, G. Pulido, M. Salazar, Handling multi-objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation 8 (2004) 256–279.
[36] P.K. Tripathi, S. Bandyopadhyay, S.K. Pal, Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients, International Journal of Information Sciences 177 (3) (2007) 5033–5049.
[37] C.R. Raquel, P.C. Naval, An effective use of crowding distance in multiobjective particle swarm optimization, in: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, 2005, pp. 257–264.
[38] M. Shahin, M. Jaksa, H. Maier, Application of Neural Networks in Foundation Engineering, in: Theme Paper to the International e-Conference on Modern Trends in Foundation Engineering: Geotechnical Challenges and Solutions, Theme No. 5: Numerical Modelling and Analysis, Chennai, India, 2004.
[39] A. Asuncion, D. Newman, UCI machine learning repository, 2007, URL (http://www.ics.uci.edu/∼mlearn/MLRepository.html).
[40] C. Goh, E. Teoh, K.C. Tan, Hybrid multiobjective evolutionary design for artificial neural networks, IEEE Transactions on Neural Networks 19 (9) (2008) 1531–1548.
[41] Y. Jin, B. Sendhoff, Pareto based approach to machine learning: an overview and case studies, IEEE Transactions on Systems, Man, and Cybernetics, Part C 38 (3) (2008) 397–415.
[42] O.L. Mangasarian, W.H. Wolberg, Cancer diagnosis via linear programming, SIAM News 23 (5) (1990) 1–18.