

A Pareto Archive Evolutionary Strategy Based Radial Basis Function Neural Network Training Algorithm for Failure Rate Prediction in Overhead Feeders

Grant Cochenour Kansas State University 2061 Rathbone Hall Manhattan, KS 66506 00-1-785-532-5600 grc3484@ksu.edu	Jerad Simon Kansas State University 2061 Rathbone Hall Manhattan, KS 66506 00-1-785-532-5600 jesimon@ksu.edu	Sanjoy Das Kansas State University 2061 Rathbone Hall Manhattan, KS 66506 00-1-785-532-4642 sdas@ksu.edu	Anil Pahwa Kansas State University 2061 Rathbone Hall Manhattan, KS 66506 00-1-785-532-5600 pahwa@ksu.edu	Surasish Nag Kansas State University 2061 Rathbone Hall Manhattan, KS 66506 00-1-785-532-5600 surasish@ksu.edu
------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------

ABSTRACT

This paper outlines a radial basis function neural network approach to predict the failures in overhead distribution lines of power delivery systems. The RBF networks are trained using historical data. The network sizes and errors are simultaneously minimized using the Pareto Archive Evolutionary Strategy algorithm. Mutation of the network is carried out by invoking an orthogonal least square procedure. The performance of the proposed method was compared to a fuzzy inference approach and with multilayered perceptrons. The results suggest that this approach outperforms the other techniques for the prediction of failure rates.

Categories and Subject Descriptors

I.2.6 [Artificial Intelligence]: Learning – *connectionism and neural nets, parameter learning.*

General Terms: Algorithms

Keywords: Pareto Archive Evolutionary Strategy, multi-objective optimization, radial basis function, neural networks, power system reliability.

1. INTRODUCTION

Reliable power distribution systems are essential to deliver electric power from electrical substations to customers. A failure in any component of a distribution system will generally cause a power interruption. Since a very large part of the distribution systems in the world consist of overhead feeders with radial configuration, failures on distribution systems are responsible for a large part of the interruptions experienced by customers.

It is therefore essential to incorporate an effective maintenance schedule for these overhead feeders, which includes the ability to predict failure rates based on historical data. Wind, lightning, ice, tornadoes and other weather are some of the factors that could cause

line failures. High tree density near a line or lack of regular tree trimming can also cause frequent interruptions.

In a previous work, an adaptive fuzzy inference system approach was used for failure rate prediction of overhead lines [1, 2, 3]. The present work proposes the use of radial basis function (RBF) neural networks for failure rate prediction. RBF networks were selected for the model, as the data set that we have considered was of low dimensionality. RBF neural networks are particularly well suited for function interpolation in these cases, and have consistently outperformed other approaches in a variety of tasks [4, 5]. Further motivation was drawn from the fact that RBF networks are functionally equivalent to fuzzy logic systems which were already shown to predict failure rates with a small error rate [6]. As these networks make use of units having highly localized responses, called kernels, many methods exist to train such networks that use clustering [4]. Evolutionary algorithms [7] have also been a popular choice amongst researchers [8, 9, 10, 11, 12, 13].

For failure rate prediction, it is especially important to accurately predict failure rates of transmission lines that are more prone to failures. Unlike local algorithms, evolutionary approaches can easily place kernels outside the convex hull of the input data set, thereby reducing edge effects making room for more accurate predictions in these boundary cases. Typically, RBF networks use a large number of kernels in their hidden layer. However, to avoid over-fitting of the training data, a parsimonious network is always desirable. In this paper, a multi-objective optimization approach has been adopted with the number of kernels being a second objective (in addition to the error). Multi-objective optimization [14, 15, 16, 17, 18] is particularly well suited in our case as historical data of failure rates is extremely difficult to obtain. Moreover, preprocessing the data is a cumbersome task [2]. A multi-objective approach yields a number of RBF networks of varying sizes, and after the training has been accomplished using training data, the best network can be selected using a separate test data set.

1.1 Pareto Archive Evolutionary Strategy

Evolutionary algorithms have emerged as one of the most popular approaches for complex optimization problems [7]. They draw upon Darwinian paradigms of evolution to search through the solution space (the set of all possible solutions). Starting with a set (or population) of solutions, in each generation of the algorithm, new solutions are created from older ones by means of a few

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '05, June 25-29, 2005, Washington, DC, USA.
Copyright 2005 ACM 1-59593-010-8/05/0006...\$5.00.

evolutionary operators, the commonest of them being mutation. Mutation is accomplished by imparting a small, usually random perturbation to the solution. In a manner similar to the Darwinian paradigm of survival of the fittest, only the better solutions are allowed to remain in a population, the degree of optimality of the solution being assessed through a measure called fitness.

When dealing with optimization problems with multiple objectives, the conventional concept of optimality does not hold good [14, 15, 16]. Hence, the concepts of dominance and Pareto-optimality are applied. Without a loss of generality, if we assume that the optimization problem involves minimizing each objective $e_i(\cdot)$, $i = 1 \dots M$, a solution u is said to dominate over another solution v iff $\forall i \in \{1, 2, \dots, M\}$, $e_i(u) \leq e_i(v)$ with at least one of the inequalities being strict, i.e. for each objective, u is better than or equal to v and better in at least one objective. This relationship is represented as $u \prec v$. In a population of solution vectors, the set of all non-dominating solutions is called the Pareto front. In other words, if S is the population, the Pareto Front is the set,

$$\Gamma = \{u \in S \mid \forall v \in S, \neg(v \prec u)\}. \quad (1)$$

Given any set S of solutions, we will use the notation $u \prec S$ to mean $v \in S \Rightarrow u \prec v$.

The simplistic approach of aggregating multiple objectives into a single one often fails to produce good results as it produces only a single optimal solution. Multi-objective optimization on the other hand involves extracting the entire Pareto front from the solution space. In recent years, many evolutionary algorithms for multi-objective optimization have been proposed [17, 19].

The Pareto Archive Evolutionary Algorithm (PAES) is a multi-objective evolutionary algorithm [20, 21, 22]. PAES maintains an archive of non-dominated solutions that is initialized at the beginning from one or more randomly generated solutions. In each iteration of PAES, a solution is selected from the archive. The solution is then mutated to produce an offspring solution. If the parent dominates the offspring, the latter is automatically discarded. However, when the offspring dominates, it replaces its parent in the archive. If neither parent nor offspring dominate, the algorithm picks the one that lies in a less crowded region of the archive. Crowdedness in PAES is measured using an adaptive grid method, the details of which are outlined in [22]. Arguably, PAES is the simplest of all multi-objective evolutionary algorithms. Other versions of PAES, such as the $(1+\lambda)$ -PAES and the $(\mu+\lambda)$ -PAES have also been proposed [22]. The present research incorporates a variation of the basis PAES algorithm, a detailed description of which is included in Section 2.

1.2 Radial Basis Function Networks

RBF networks considered in this paper are capable of performing mappings from an n -dimensional input \mathfrak{R}^n to a one-dimensional output. The network consists of a set of N nonlinear elements with local responses, called kernels. For an input $\mathbf{x}(q)$, $q=1, 2, \dots, Q$, drawn from a set of Q samples, the response of the i^{th} kernel is given by,

$$\phi_i(\mathbf{x}(q)) = e^{-\frac{(\mathbf{x}(q) - \mu_i)^2}{\sigma_i^2}}. \quad (2)$$

In the above equation μ_i and σ_i are parameters associated with each kernel, usually referred to as its location and width. The network output is a linear combination of the kernel responses,

$$y(q) = \sum_{i=1}^K w_i \phi_i(\mathbf{x}(q)), \quad (3)$$

where each w_k is a weight linking the kernel to an output summing unit. Arranging the kernel responses as a $Q \times N$ matrix, Φ , whose $(q, i)^{\text{th}}$ element is $\phi_i(\mathbf{x}(q))$, and the weights and outputs as $N \times 1$ and $Q \times 1$ vectors, \mathbf{w} and \mathbf{y} we get,

$$\mathbf{y} = \Phi \mathbf{w}. \quad (4)$$

Suppose \mathbf{t} is a vector of desired outputs, called targets, the mean square error between the actual outputs and targets is,

$$E = \frac{1}{2} (\mathbf{t} - \mathbf{y})(\mathbf{t} - \mathbf{y})^T \quad (5)$$

The optimal weight vector to minimize the mean square error is given by [4],

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t} = \Phi^+ \mathbf{t}, \quad (6)$$

where Φ^+ is the pseudoinverse of Φ . A number of RBF training methods apply orthogonal least square (OLS) techniques to the response matrix Φ . We postpone a fuller discussion on this approach until Section 2.

Evolutionary algorithms have been very popular in determining the parameters of RBF networks. A genetic algorithm was used in [9] to evolve the network for function approximation. The genetic algorithm population consisted of individual kernels instead of entire networks. The kernel locations and widths were evolved, while the weights were determined separately. The network size was a constant. A similar strategy has been used in [8] which makes use of an OLS procedure and singular value decomposition to evaluate the contribution of individual kernels. Other methods [e.g. 10] evolve an entire population of RBF neural networks. A two-level algorithm was proposed in [12] where a micro genetic algorithm at the upper level is used to evolve the kernel widths, while a lower level OLS process determines the other parameters. In [11] the number of kernels is also determined dynamically using a genetic algorithm. In the first application of multi-objective optimization to RBF training [13], the network size is the second objective to be minimized. The proposed method makes use of an OLS algorithm and singular value decomposition on the kernel response matrix to perform mutation.

PAES was selected as the training algorithm in this paper as it is arguably the simplest of all multi-objective optimization algorithms. In terms of algorithm complexity, it compares favorably with other popular approaches such as NSGA-II and SPEA [17, 19]. Furthermore, preliminary simulations by us suggested that applying crossover, one of the standard operators in evolutionary algorithms, failed to improve the convergence towards the Pareto front, while

adding to computer time. Unlike genetic algorithms, evolutionary strategies rely almost solely on mutation, the other primary evolutionary operator. Keeping in view the general philosophy of simplicity in PAES, we have devised a simple but effective mutation operator to evolve accurately trained RBF networks.

2. PROBLEM DESCRIPTION

Reliable power distribution systems are essential to deliver electric power from generating stations to customers. A failure in any component of a distribution system will cause a power interruption. Since a very large part of the distribution systems in the world consist of overhead feeders with radial configuration, failures on distribution systems are responsible for a large part of the interruptions experienced by customers. It is therefore essential to incorporate an effective maintenance schedule for these overhead feeders, which includes the ability to predict failure rates based on historical data. Wind, lightning, ice, tornadoes and other weather are some of the factors that could cause line failures. High tree density near a line or lack of regular tree trimming can also cause frequent interruptions.

Previous work on the prediction of the failure rates of overhead lines has used an adaptive fuzzy inference system approach [3]. The data to train the model was gathered from various sources, including a large utility company based in the western United States, and the National Climatic Data Center, for a seven year period, 1990-1996. A total of 37 feeder sections were involved. The input data into this system consisted of four fields, which were, (i) the lightning rate, i.e. the number of days experiencing lightning each year, (ii) the tree density, i.e. the number of trees per mile length of feeder-section, (iii) the years since tree trimming was carried out in that area, and (iv) the wind index, which incorporates information on the wind velocity as well their duration. The output data, used for training, was the failure rate, i.e. the average number of failures per mile of feeder, per year. 75% of the data was used to train the fuzzy model, while the rest was used to evaluate its performance. For further details, one is referred to [2]. Since it is known *a priori* that less tree trimming, or excess lightning activity would increase the likelihood of failures, such domain knowledge was effectively used to design a fuzzy rule base. The rule base was further fine-tuned using a gradient descent algorithm to minimize the mean square error between the model's predicted failure rate, and the actual values obtained from the data. The research in this paper uses the same data for predicting failure rates. The results of that earlier research have also been used here for the sake of comparison.

3. APPROACH

3.1 Algorithm Description

The proposed algorithm begins with by generating several RBF networks and picking out the non-dominated set as the initial archive. The network size, and the kernel location and width parameters of the initial RBF networks are generated randomly, but the weights are computed using Equation (6). The archive records the size, as well as the kernel locations, widths and weights up to a limit. Within each iteration of the algorithm, a solution u is picked at random from the archive A . The solution is mutated to produce an offspring v . The offspring is compared with the parent solution. If $v \prec u$, v is inserted into A . Any archived solution w , including u

that is dominated by v is also removed. On the other hand, if v is dominated by u , v is discarded. When neither $u \prec v$ nor $v \prec u$, the algorithm checks to see if v dominates any other solution in the archive A . If so, then v is inserted into A while u as well as other solutions in A that are dominated by v are discarded. If none of these cases apply, then the algorithm uses the approach adopted by most multi-objective evolutionary algorithms to prefer solutions that lie in sparser regions of the non-dominated front. In a manner similar to NSGA-II [19] picks the solution that lies in a more sparsely located region in the archive A . The crowdedness may be estimated by first sorting the solutions according to their errors, and then considering the perimeter of a rectangle that encloses the solution, and whose corners are the latter's neighbors in the sorted list [19]. For any RBF solution v , is given by,

$$\text{crowd}(v) = \frac{|E_u - E_w|}{\max(E) - \min(E)} + \frac{|N_u - N_w|}{\max(N) - \min(N)}. \quad (7)$$

In the above equation, u and w are v 's neighbors in the sorted list, and E_u , E_v , and N_u , N_v are the associated errors as given in Equation (5) and the number of kernels. The $\min(\cdot)$ and $\max(\cdot)$ operators are carried out over the entire archive A . The pseudocode for the algorithm is provided below.

```

populate A with initial RBFs;
for each w in A that is dominated,
    A = A - {w};
end
t=0;
while t<maxIterations
    select u randomly from A;
    v = mutate(u);
    if u < v
        do nothing
    elseif v < u
        A = A ∪ {v};
        for each w in A that is dominated by v
            A = A - {w};
        end
    elseif neither v < u nor u < v
        if v < w for some w in A
            A = A ∪ {v};
            for each w in A that is dominated
                A = A - {w};
            end
        endif
    else z = argmin(crowd(u), crowd(v))
        A = A - {u};
        A = A ∪ {z};
    endif
    t=t+1;
end

```

3.2 Mutation

Mutation is invoked by applying the Gram-Schmidt orthogonalization algorithm to the kernel response matrix, Φ . Using this method Φ can be factored into two matrices, \mathbf{B} and \mathbf{A} of size $Q \times N$ and $N \times N$, as,

$$\Phi = \mathbf{B}\mathbf{A} = [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \dots \quad \mathbf{b}_i \quad \dots \quad \mathbf{b}_N], \quad (8)$$

where the columns of \mathbf{B} are mutually orthogonal (i.e. $\mathbf{b}_i^T \mathbf{b}_j = 0$).

Under these circumstances it has been shown in [5] that when the weights are optimally chosen based on Equation (6),

$$\mathbf{t}^T \mathbf{t} = \sum_{i=1}^N \frac{(\mathbf{b}_i^T \mathbf{t})^2}{\mathbf{b}_i^T \mathbf{b}_i} + \mathbf{e}^T \mathbf{e}, \quad (9)$$

where \mathbf{e} is a vector of errors. Each term in the summation is the contribution of the i^{th} kernel to the target and can be isolated.

The mutation operator acting upon a kernel of size N can either reduce its size to $N-2$, increase it to $N+2$, or not alter the size, each outcome being equally probable. The mutation operator inserts three new candidate kernels into the RBF network and applies the OLS algorithm to determine the contribution of the N existing kernels, as well as the three new ones if each were to be individually added to the network. Out of the $N+3$ candidate kernels, the operator retains the $N-2$, $N-1$, N , $N+1$, or $N+2$ kernels with the highest contribution, discarding the remainder.

The kernel widths are computed using the standard nearest neighbor heuristic,

$$\sigma_i = \min_j (\|\mu_i - \mu_j\|) / \sqrt{N}, \quad (10)$$

where \min is the minimum operation. The kernel locations are subjected to small perturbations from their original locations by adding uniformly distributed random vector,

$$\mu_i = \mu_i + \text{rnd}(-\varepsilon, \varepsilon). \quad (11)$$

In the above equation, $\text{rnd}(\cdot)$ generates random vectors whose elements lie in the range $(-\varepsilon, \varepsilon)$. The kernel weights are determined optimally using (6).

4. RESULTS

4.1 Experiment Setup

Testing of the proposed algorithm was implemented in MATLAB Release 14. The maximum archive size was set to 20. The archive is initialized with 10 non-dominated networks of with random number of kernels, K , and random kernel centers with the widths determined by Equation (10). The number of kernels, K , for each candidate solution is limited to the range of 4 to 50 for the duration of the experiment. The nearest neighbor heuristic in Equation (10) has a uniformly distributed random scaling factor on the interval $[1, 10]$. The norm of the perturbation described in Equation (11) for each kernel in a solution, μ_i , is always less than the kernel's width, σ_i , as in [13]. The selection of a solution from the archive for mutation is biased towards larger networks because larger networks take more time to converge. The algorithm was run for 5000 iterations.

The data for the experiment is the power system reliability data described earlier. This data was broken up in the same manner as Case 4 from [2] into test and training data sets. The data sets have 4 input variables and 1 output. The training data set has 1775 samples, and the test data set has 591 samples. This data is then normalized. The training data is used in the algorithm, and the test data is used to verify the generality of the resulting solutions.

4.2 Results

The proposed algorithm performs very well on the supplied data. Figure 1 shows the Pareto front evolving. The RBF networks are gradually improved. Small networks converge within only a few perturbations while the larger networks may take many steps to arrive at a good solution. Use of the nearest neighbor heuristic generally assures that the final solutions will have sufficient

generality as shown in Figure 2. The test data actually has lower error than the training data. This unusual finding was also observed in [2].

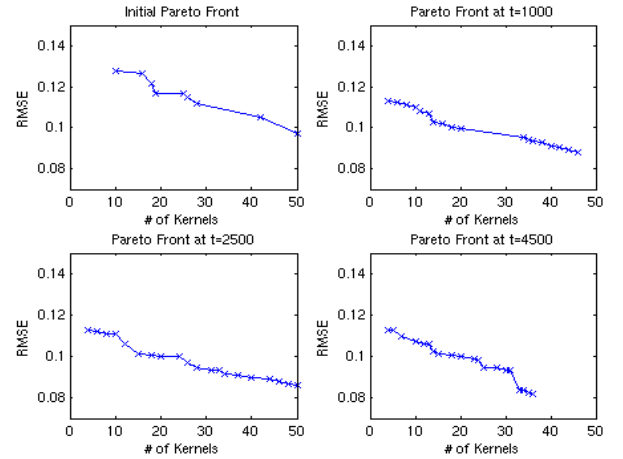


Figure 1. Pareto front at select iterations

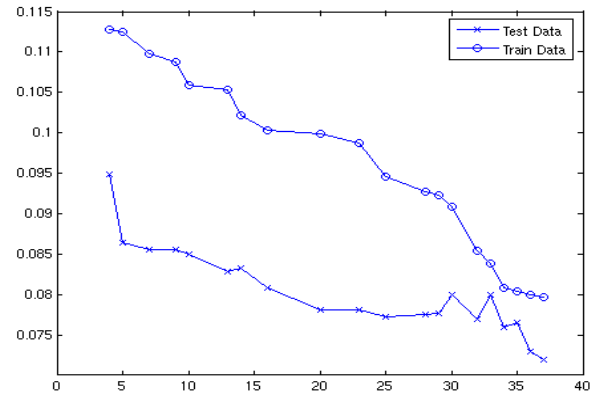


Figure 2. Final Pareto front generated by the proposed algorithm and the performance of the generated solutions on the test data set

Figure 3 shows the convergence of selected network sizes. These plots are more discreet in nature because a network of with a selected number of kernels is not always in the archive nor is it mutated at each iteration. Because a network size is not always present it is possible for plots like in Figure 3 to not be monotonically decreasing when the Pareto front has made only minimal improvement and a solution N kernels is removed and then later reinserted into the archive.

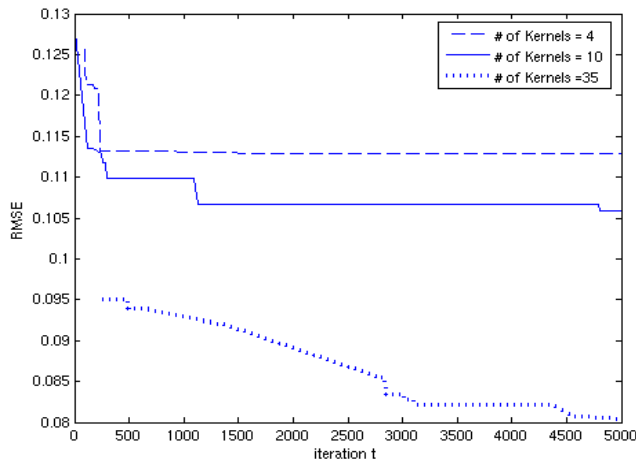


Figure 3. Error of select networks through time

4.3 Performance Comparison

The proposed algorithm was compared against other established methods in Table 1. It was compared against the fuzzy inference system presented in [2] and against standard multi-layered perceptrons (MLPs) with various network sizes. The results reported in Table 1 for the proposed algorithms and MLPs are the averaged results over 10 trials.

The MLPs were trained with standard backpropagation and momentum using the standard MATLAB neural network toolbox. The best learning rate and momentum for the MLP were found to be 0.1 and 0.5 respectively, which were significantly better than the default values the toolbox provides. The MLP training was halted after the error for the test data shows no further improvement.

Table 1. Average RMS error of proposed algorithm in comparison with other methods.

	RMSE Training	RMSE Test
Proposed Algorithm (N=5)	0.110	0.093
Proposed Algorithm (N=10)	0.103	0.085
Proposed Algorithm (N=16)	0.100	0.081
FIS	0.142	0.121
MLP (4x2x1)	0.129	0.123
MLP (4x3x1)	0.122	0.119
MLP (4x4x1)	0.118	0.114

The proposed algorithm outperforms both the MLP and the fuzzy inference approaches. The runtime for the proposed algorithm is 2127 seconds, and for the three MLP models collectively, the runtime is 2640 seconds. Larger MLPs showed no significant improvement in the performance for the training data, while the error for the test data increased. The error reported here is the

standard RMS error, which is $\sqrt{E/Q}$, where E is computed using Equation (5) and Q is number of training samples.

5. CONCLUSION

In this paper a modified version of the PAES algorithm has been presented for the optimization of RBF networks used in the prediction of overhead distribution failure rates. The presented algorithm is simple and outperforms other methods in the prediction of failure rates on overhead distribution feeders.

6. REFERENCES

- [1] Gupta, S., Pahwa, A., Brown, R. E., Das, S.: A Fuzzy Model for Overhead Distribution Feeders Failure Rates, Proceedings of the 34th North American Power Symposium, Arizona State University, (2002), pp. 248-253.
- [2] Gupta, S., Pahwa, A., Zhou, Y., Brown, R. E., Das, S.: Data Selection To Train A Fuzzy Model For Overhead Distribution Feeders Failure Rates, Proceedings of the ISAP 2003 Conference, Lemnos, Greece, (2003).
- [3] Gupta, S., Pahwa, A., Zhou, Y., Brown, R. E., Das, S.: An Adaptive Fuzzy Model For Overhead Distribution Feeders Failure Rates, Electric Power Components and Systems, in press.
- [4] Broomhead, D. S., Lowe, D.: Multivariate functional interpolation and adaptive networks, Complex Systems, Vol. 2, (1988), pp. 321-355.
- [5] Chen, S., Cowan, C.F.N., Grant, P.M.: Orthogonal least squares learning for radial basis function networks, IEEE Transactions on Neural Networks, Vol. 2, no. 2, (1991), 302-309.
- [6] Jang, J. -S. R., Sun, C.-T.: "Functional equivalence between radial basis function networks and fuzzy inference systems", IEEE Transactions on Neural Networks, Vol. 4, no. 1 (1993) 156-159.
- [7] Goldberg D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Reading, MA (1989).
- [8] Whitehead, B. A.: Genetic Evolution of Radial Basis Function Coverage Using Orthogonal Niches, IEEE Transactions on Neural Networks, Vol. 7, no. 6, (Nov 1996), 1525-1528.
- [9] Whitehead, B. A., Choate, T. D.: Cooperative-Competitive Genetic Evolution of Radial Basis Function Centers and Widths for Time Series Prediction, IEEE Transactions on Neural Networks, Vol. 7, no. 4, (July 1996), 869-880.
- [10] Billings, S. A., Zheng, G. L.: Radial basis function network configuration using genetic algorithms, Vol. 8, no. 6 (1995), 302-309.
- [11] Alexandridis, A., Sarimveis, H., Bafas, G.: A new algorithm for online structure and parameter adaptation of RBF networks. Neural Networks, Vol. 16, no. 5-6 (June 2003), pp. 529-535.
- [12] Chen, S., Wu, Y., Luk, B. L.: Combined Genetic Algorithm Optimization and Regularized Orthogonal Least Square Learning for Radial Basis Function Networks, IEEE Transactions on Neural Networks, Vol. 10, no. 5, (Sept 1999), 1239-1243.

- [13] González J., Rojas, I., Ortega, J., Pomares, H., Díaz, A.: Multiobjective Evolutionary Optimization of the Size, Shape and Position Parameters of Radial Basis Function Networks for Function Approximation, *IEEE Transactions on Neural Networks*, Vol. 14, no. 6, (Nov 2003), 1478-1498.
- [14] Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation*, Vol. 3, no. 1 (Spring 1995) 1-16.
- [15] Coello Coello, C.A.: A comprehensive survey of evolutionary-based multiobjective optimization techniques, *Knowledge and Information Systems*, Vol. 1, no. 3 (Aug 1999) 269-308.
- [16] Van Veldhuizen, D.A., Lamont, G.B.: Multiobjective evolutionary algorithms: Analyzing the state-of-the-art, *Evolutionary Computation*, Vol. 8, no. 2, (2000) 125-147.
- [17] Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3(4): 257-271 (1999).
- [18] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Transactions on Evolutionary Computation*, Vol. 7, no. 2 (Apr. 2003) 117-132.
- [19] Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2): 182-197 (2002).
- [20] Knowles, J., Corne, D.: Approximating the nondominated front using the Pareto Archived Evolution Strategy, *Evolutionary Computation*, Vol. 8, no. 2 (2000) 149-172.
- [21] Knowles, J., Corne, D.: Properties of an adaptive archiving algorithm for storing nondominated vectors, *IEEE Transactions on Evolutionary Computation*, Vol. 7, no. 2 (Apr. 2003) 100-116.
- [22] Knowles, J., Corne, D.: The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation, *Proceedings of the 1999 Congress on Evolutionary Computation* (1999), 98-105.