A PAES Based Optimization Of RBF Networks To Predict Overhead Feeder Failures

Grant Cochenour[1], Jerad Simon[1], Surashish Nag[1], Oluwarotimi Odeh[2], Jan Poupe[1], Anil Pahwa[1]

[1]Department of Electrical and Computer Engineering
Kansas State University
Manhattan, KS 66506

[2] Department of Agricultural Economics
Kansas State University
Manhattan, KS 66506

Abstract

This paper details a method for the multi-objective optimization of radial basis function (RBF) networks to predict failures in an overhead power distribution system. The error and number of kernels are both minimized using the Pareto Archive Evolutionary Strategy. The networks are trained to predict the number of failures based upon daily weather data and outage histories from a Midwestern utility. The results indicate that this method is capable of predicting the daily number of failures.

Keywords: Pareto Archive Evolutionary Strategy, Radial Basis Function

1. Introduction

Reliable distribution systems are important for the delivery of electric power to customers. Common factors that cause outages are trees, lightning, animals, and wind. The present work uses RBF networks to predict the daily number of failures based upon daily wind and lightning measurements.

There are many methods available to train RBF networks like clustering [1]. Evolutionary algorithms have also been used extensively [3, 4, 5, 6, 7, 8].

For the prediction of power outages, it is more important to predict days with many failures that are rare. Unlike clustering and other local algorithms, an evolutionary algorithm can place kernels outside the convex hull of the dataset, improving the performance of the RBF network on boundary cases. RBF networks use a number of localized response units called kernels. However, simple networks are desirable so as to have sufficient generality. In this paper a multi-objective evolutionary algorithm (MOEA) has been adopted. The two objectives to minimize are mean squared error (MSE) and the number of kernels. This approach is well suited [9, 10, 11, 12, 13] to this application as historical outage data is hard to find. The approach creates networks of different sizes. After the networks are optimized, the best network can be selected based upon some other expert knowledge.

2. Algorithm

RBF networks can map an *n*-dimensional input to a one-dimensional output. An RBF network consists of N local response units called kernels. For an input $\mathbf{x}(q)$, q=1, 2, … Q, drawn from Q training samples, the response of the ith kernel is given by,

$$\phi_i\left(\mathbf{x}(q)\right) = e^{-(\mathbf{x}(q)-\mu_i)^2/\sigma_i^2}. \tag{1}$$

In the above equation, μ_i and σ_i are parameters called location and width and are specific to each kernel. The output of the RBF network, y, in response to an input vector $\mathbf{x}(q)$ is a weighted sum of the kernel outputs,

$$y(q) = \sum_{i=1}^{N} w_i \phi_i\left(\mathbf{x}(q)\right). \tag{2}$$

Each w_i is a weight, linking the ith kernel to the output. When the kernel responses are used to construct a $Q \times N$ matrix, Φ, whose (q, i)^th^ element is $\phi_i\left(\mathbf{x}(q)\right)$ and the network output for each training sample and the kernel weights are arranged as $Q \times 1$ and $N \times 1$ vectors, $\mathbf{y}$ and $\mathbf{w}$, then the output vector $\mathbf{y}$ is

$$\mathbf{y} = \Phi \mathbf{w}. \tag{3}$$

If $\mathbf{t}$ is a $Q \times 1$ vector of desired outputs, then the sum-squared error between the network outputs and desired outputs is

$$E = \tfrac{1}{2}(\mathbf{t}-\mathbf{y})(\mathbf{t}-\mathbf{y})^T. \tag{4}$$

It was shown in [1] that there is exactly one weight vector that will minimize Equation (4). If $Q>N$, then the optimal weight vector is given by,

$$\mathbf{w} = \left(\Phi^T \Phi\right)^{-1} \Phi^T \mathbf{t} = \Phi^+ \mathbf{t}, \tag{5}$$

where Φ^+ is called the pseudoinverse of Φ.

The Pareto Archive Evolutionary Strategy (PAES) was outlined in [15, 16, 17] and is one of the simplest MOEAs. In multi-objective optimization problems, the conventional concept of optimality or fitness is not as useful [9, 10, 11]. So, the concepts of dominance and Pareto optimality are applied. If we assume that the optimization problem involves minimizing each objective $e_i(.)$, $i = 1...M$, a solution j is said to dominate another solution k if and only if $\forall i \in \{1,2,\ldots,M\}, e_i(j) \le e_i(k)$ with at least one of the inequalities being strict. This relationship is written as $j \prec k$. In a population of solution vectors, the set of all non-dominated solutions is called the Pareto optimal set or Pareto Front. In other words, if P is a population of solutions, then the Pareto front is a set defined as

$$\Psi = \left\{ j \in P \mid \forall k \in P, \neg(k \prec j) \right\}. \quad (6)$$

PAES keeps non-dominated solutions found up to that point in an archive of some predetermined maximum size. In each iteration, a single solution is selected from the archive and mutated to produce a new candidate solution. If the parent solution dominates the new solution, then the new solution is thrown away. However, if the new solution dominates the parent, the new solution replaces the parent in the archive as well as any other solutions in the archive that are dominated by the new solution. If neither the parent nor offspring dominates, the algorithm retains the solution that is lies in the sparsest region to encourage a uniformly spread Pareto Front. Crowdedness is measured by a grid method that is outlined in [17]. Other versions of PAES have also been proposed, but the present research utilizes a variation of this basic PAES algorithm.

The proposed algorithm begins by generating several networks of random size and keeping those solutions that are not dominated. The number of kernels and their locations are randomly assigned; the widths are assigned with Equation (10); the weights are computed using Equation (5). Within each iteration, a solution j is picked at random from the archive A. The solution is mutated to produce a new solution k. If $k \prec j$, then k is inserted into the archive. Any archived solution l, including j, that is dominated by k is removed from the archive. If $j \prec k$, then k is discarded. If neither j or k dominate each other, the algorithm checks to see if the mutant dominates any other solutions currently in the archive and removes any dominated solutions. If the mutant and all archived solutions are non-dominated and the archive is full, the algorithm favors solutions that lie in sparsely populated regions of the objective space. If the archive is not full, then k is added to the archive regardless of any crowding considerations. The crowdedness can be estimated by sorting the solutions according to the number of kernels and then considering the perimeter of the rectangle that encloses the solution and whose corners are the latter's neighbors in the sorted list [14]. For any RBF solution k, the crowdedness is given by

$$\text{crowd}(k) = \frac{\left|E_j - E_l\right|}{\max(E) - \min(E)} + \frac{\left|N_j - N_l\right|}{\max(N) - \min(N)}. \quad (7)$$

In the above equation, j and l are neighbors of k in the sorted list, and E_j, E_l, N_j, and N_l are the errors as given in Equation (4) and the number of kernels for each network. The min(.) and max(.) operators are applied to the entire archive.

PAES was selected over other algorithms for this work like NSGA-II and SPEA because initial work by us suggested that a crossover operation was not effective and only added to the runtime of the algorithm. Additionally, the algorithmic complexity of PAES is favorable compared to NSGA-II and SPEA [13, 14].

Evolutionary strategies like PAES rely solely on mutation. We have devised a simple but effective mutation operator to evolve well trained RBF networks using PAES.

Mutation is accomplished with the Gram-Schmidt orthogonalization algorithm to the matrix, $\mathbf{\Phi}$. Using this method, $\mathbf{\Phi}$ can be decomposed into two matrices **A** and **B**, of size $N \times N$ and $Q \times N$, as,

$$\mathbf{\Phi} = \mathbf{BA}, \quad (8)$$

where the columns of **B** are orthogonal to each other. It has been shown in [2] that when the weights are optimally chosen based on Equation (5),

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

where $\boldsymbol{e}$ is a vector of error terms. Each term in the summation is the contribution of the ith kernel to the target.

The mutation operator acting upon a network of size, N, adds three new candidate kernels to the network. Then the contribution of each of the N+3 kernels is determined with the orthogonalization algorithm. The algorithm retains N-2, N-1, N, N+1, or N+2 kernels whose contributions are the largest and throws the remaining kernels away. The kernel widths are assigned by using the 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 randomly subjected to small disturbances from their original locations by adding a uniformly distributed random vector,

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

In the above equation, rnd(.) generates random numbers whose elements lie in the range of (-ε,ε). The kernel weights for the new solution are then found with Equation (5).

The pseudocode for the algorithm is provided below:

```
populate archive A with random networks;
for each l in A that is dominated,
    A = A - {l};
end
until done{
select j randomly from A;
k= mutate(j);
if j≺k
    discard k;
elseif k≺j
    A = A ∪ {k};
    for each l in A that is dominated
        A = A-{l};
elseif neither k≺j nor j≺k
    if k≺l  for some l in A
A = A ∪ {k};
        for each l in A that is dominated
            A = A-{l};
    endif
else    t = argmin(crowd(j), crowd(k))
        A = A-{j};
        A = A∪ {t};
endif
end
```

3. Results and Discussion

The outage history used for training was provided by a Midwest utility. Outages that were not caused by wind and lightning were excluded. Lightning and wind information for the same time period and location was also collected on a daily basis. There are three input variables: sum of daily lightning current magnitudes, maximum daily 5-second gust speed, and the product of the previous variables. The model output is the expected number of weather-related failures for the day. There are 2147 training samples.

The archive was given a maximum size of 20 and was initially populated with five undominated networks. The number of kernels for each of these networks was randomly assigned but was restricted to the range of four to 50; the locations and widths of the individual kernels were determined by Equations (10, 11). The norm of the perturbation applied by Equation (11) is always less than the kernel's width as in [8]. The selection of networks from the archive to mutate is biased towards the larger networks as they take longer to converge. The algorithm was allowed to run for 5000 iterations.

The proposed algorithm performs well on the supplied data. Figure 1 shows the improvement of the Pareto front at selected iterations.

A 25-kernel network predicts 91% of the data within +/- 1 failure. The benefits of evolutionary algorithms can be seen in Figure 2. The model predicts several rare, high failure events very well. This is important, as utilities are more interested in predicting high failure days than low ones. A localized algorithm would be heavily influenced by the large number of low failure days and would not accurately predict these boundary cases.

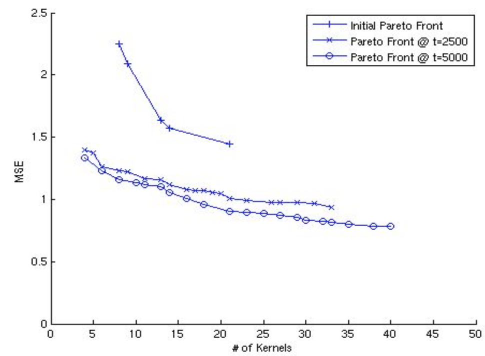

Figure 1. Pareto front at selected iterations

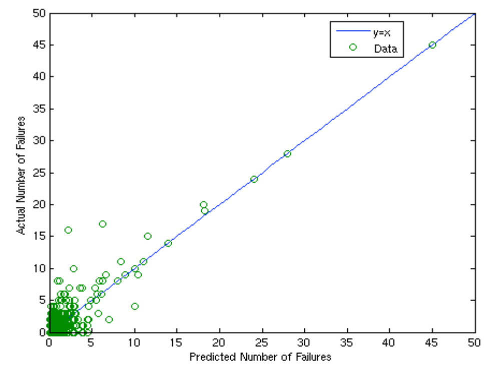

Figure 2. Actual # of Failures vs. 25 kernel Network Prediction

One potential application of daily failure prediction is the subsequent forecast of monthly failures by aggregating the daily predictions. Figure 3 demonstrates the effectiveness of this method at predicting monthly failures. When allowed sufficient time, the over and under predictions of the network average out and result in a very good monthly forecast.

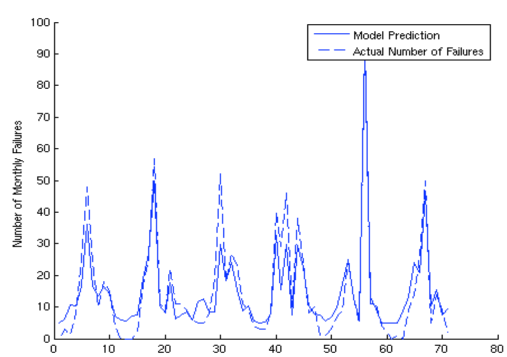

Figure 3. Aggregated Daily Prediction for Monthly Failures with a 25 kernel network

4. 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 failures. The proposed technique is a simple, yet effective method for simultaneously minimizing RBF network size and the error. The resulting networks are effective at predicting the number of overhead distribution feeder failures given weather information for the day. Using daily prediction to forecast the monthly number of failures is also effective.

References

[1] Broomhead, D. S., Lowe, D.: Multivariate functional interpolation and adaptive networks, Complex Systems, Vol. 2, (1988), pp. 321-355.

[2] 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.

[3] 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.

[4] 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.

[5] Billings, S. A., Zheng, G. L.: Radial basis function network configuration using genetic algorithms, Vol. 8, no. 6 (1995), 302-309.

[6] 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.

[7] 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.

[8] González J., Rojas, I., Ortega, J., Pomares, H., Díaz, A.: Multiobjective Evolutionary Optimization of the Size, Shape and Position Parameters of Radial Basin Function Networks for Function Approximation, IEEE Transactions on Neural Networks, Vol. 14, no. 6, (Nov 2003), 1478-1498.

[9] Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization, Evolutionary Computation, Vol. 3, no. 1 (Spring 1995) 1-16.

[10] 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.

[11] 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.

[12] 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).

[13] 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.

[14] 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).

[15] Knowles, J., Corne, D.: Approximating the nondominated front using the Pareto Archived Evolution Strategy, Evolutionary Computation, Vol. 8, no. 2 (2000) 149-172.

[16] 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.

[17] 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.