A Self-Adaptive Fuzzy C-means based Radial Basis Function Network to Solve Economic Load Dispatch Problems

Surekha P
Research Scholar
EEE Department,
PSG College of Technology, India

Dr.S.Sumathi
Associate Professor
EEE Department,
PSG College of Technology, India

ABSTRACT
In recent decades, with a large increase in power demand, fuel cost, and limited fuel supply it has become very essential to run the power systems with minimum cost so that the committed units serve the expected load demand. The basic objective of Economic Load Dispatch (ELD) is to distribute the total generation among the generation units in operation, in order to meet the load demand at minimum operating cost while satisfying the system equality and inequality constraints. Nature inspired computing techniques like Artificial Neural Networks (ANN) are preferred for solving ELD problems because they do not impose any restrictions on the shape of the fuel cost curve and are capable of providing good solution quality, and higher precision solutions very close to the global optimum. In this paper, the application of Fuzzy c-means based Radial Basis Function Network (RBFN) to ELD is proposed in order to minimize the error function through a self adaptive process until the error is less than a given tolerance leading to a best solution. The applicability and viability for practical applications has been tested on two different power systems, viz., a IEEE 30 bus 6 unit test system and a 20 unit test system and the experiments were carried out on MATLAB R2008b software. Comparison of the results with the conventional Lambda Iteration method demonstrates the effectiveness of RBFN in solving ELD problems based on fuel cost, power loss, total generated power, algorithmic efficiency, and computational time.

Keywords
Economic load dispatch, Fuzzy c-means clustering, Radial Basis Function Network, algorithmic efficiency, computational time, APAE.

1. INTRODUCTION
Economic Load Dispatch (ELD) is one of the most significant optimization problems in modern computer aided power system design. With increasing fuel cost and power demands, optimization of economic dispatch brings a lot of revenue to the network operator. So, it has become important to allocate the total load between the available generating units in such a way that the total cost of operation is kept at a minimum [(Lakshmi Devi A. 2008)]. In traditional methods like lambda iteration (LI) method [(C.L. 2000)], and gradient-based method, [(J. B. Park 1993)], the solution to ELD is obtained by approximately representing the cost function for individual generators in terms of single quadratic function. These techniques require incremental fuel cost curves which are piecewise linear and monotonically increasing to find the global optimal solution [(Rayapudi 2011)]. For generators with non-monotonically incremental cost curves, conventional methods ignores or flattens out portions of incremental cost curve that are not continuous or monotonically increasing. Thus these methods require approximation of characteristics to meet the requirements, which in turn leads to an increase in the revenue over time. [(H. T. Yang 1996)], [(T. Jayabarathi 1999)]. Other classical methods like Newton-based techniques are not capable of performing for ELD problems with highly non-linear characteristics and a large number of constraints. Though dynamic programming is capable of solving non-linear and discontinuous problems, it suffers from the problem of curse of dimensionality. [(Glover 1992)].

Among these conventional methods, the LI method has been applied to find near optimal solutions to ELD problems for a very long time. The initial choice of lambda is an important factor which decides the convergence of the iterations. This method solves the ELD with two types of iterations – first, the value of lambda is changed iteratively from its initial assumed value to the final optimal value; second, for every value of lambda chosen by trial, the power generated by the generating units have to be acquired using sub-iterations. Hence the sub-iterations have to be run several times leading to a time consuming job [(Aravindhababu P. 2001)]. Over the years several efforts have been made to find optimized solutions to the ELD problem based on Artificial Intelligence such as Artificial Neural Networks, Fuzzy Logic, and Evolutionary Algorithms. Some of the most popular ANNs are Hopfield, Multilayer Perceptron, Learning vector quantization, Radial basis function, Adaptive resonance theory, and Back propagation networks. Several neural networks like Hopfield network, back propagation network and Perceptron network have been proposed for solving the ELD problem [(Djukanovic.M. 1996), (Singh. G. 1995), (Matuda.S. 1989)]. In back propagation networks, the training is based on non-linear optimization technique and hence the solution to ELD problem is obtained at a very slow rate. In addition, there is no standard rule to fix the number of hidden neurons and hence the network may not be able to provide a general optimal modeling for the given ELD system. Hopfield networks converge slowly and normally take more than thousand iterations to dispatch the power optimally. The above mentioned drawbacks of back propagation and Hopfield network are overcome by the Radial Basis Function Network (RBFN), due to its Gaussian activation function.

This paper focuses on solving ELD using Fuzzy c-means (FCM) clustering and Radial Basis Function Network (RBFN), with the objective to obtain minimum fuel cost, and an optimized load dispatch with less computational time. The choice of hidden layer neurons for the RBFN is a very important factor that is capable of approximating any given function with arbitrary precision. Hence Fuzzy c-means clustering was adopted as a pre-processing algorithm to the RBFN in order to dimensionally reduce the data allowing a simpler RBF model for ELD problems. RBFN were first introduced by Powell to solve the real multivariate interpolation problem [(Powell.M.J.D 1977)]. In contrast to ANN, the advantages of RBF network are its compact topology and the algorithm requires less training time for learning patterns. The learning strategy is based on random selection of input data sets as RBF centers in the hidden layer [(Chao-Ming Huang 2007)]. The weights between hidden and output layer can then be estimated by using the Gaussian activation function. Moreover, this network is mathematically simple with relatively low computational effort.

2. LITERATURE SURVEY

Several conventional techniques are available in literature for solving the ELD problem. They include the conventional Lambda Iteration method [(C.L. 2000)], dynamic programming [(Lowery 1983)], mixed integer programming [(Wilson 1968)], branch and bound [(Yoshimura 1983)], and Newton's method [(Wood J. 1984)]. Heuristic methods like Tabu search [(W.-M. Lin 2002)], Simulated Annealing [(Wong 1994)] etc., were also applied for ELD problems. Ching-Tzong Su et. al [(Ching-Tzong Su 2000)] presented a new Hopfield model based approach for the economic dispatch problem, by including the computational procedures with a series weighting factor adjustments associated with the transmission line losses, updating the unit generations and power losses inorder to minimize the value of the energy function. Aravindhababu et. al. presented an on-line approach for solving the ELD using RBFN which directly produced the optimal lambda value. This value was applied further to compute the economic generations iteratively [(Aravindhababu P. 2001)]. In [(Chao-Ming Huang 2007)] Chao-Ming Huang et al., proposed a novel technique that combines orthogonal least-squares (OLS) and enhanced particle swarm optimization (EPSO) algorithms to construct the radial basis function (RBF) network for real-time power dispatch.

3. PROBLEM DEFINITION

The principal objective of the economic load dispatch problem is to find a set of active power delivered by the committed generators to satisfy the required demand subject to the unit technical limits at the lowest production cost. The optimization of the ELD problem is formulated in terms of the fuel cost expressed as,

$$F_T = \sum_{i=1}^{n} F_i(P_{Gi}) = \sum_{i=1}^{n} a_i + b_i P_{Gi} + c_i P_{Gi}{}^2 \qquad [1]$$

Where F_T is the fuel cost of the system, F_i = fuel cost of the i^{th} generating unit of the system, P_{Gi} = power generated in the i^{th} generating unit, n = number of generators, a_i, b_i, c_i = cost coefficients of the i^{th} generator.
Subject to the equality constraint,

$$\sum_{i=1}^{N} P_{Gi} = P_D + P_L \qquad [2]$$

where P_{Gi} represents the generated power, P_D is the total active power demand and P_L represents the transmission losses.
Subject to the inequality constraint,

$$P_{Gi\min} \le P_{Gi} \le P_{Gi\max} \qquad [3]$$

where, $P_{Gi\min}$ is the minimum value of the real power, $P_{Gi\max}$ is the maximum value of the real power and P indicates the generated real output power.
Taking these constraints into consideration, the biologically inspired artificial neural network, FCM based RBFN is proposed to obtain well-distributed dispatch solutions for ELD. The effectiveness of the techniques is investigated on two test systems consisting of six and twenty generating units, yielding higher quality solution including fast convergence, diversity maintenance, robustness and scalability. The results obtained are compared with the conventional Lambda Iteration Method.

4. IMPLEMENTATION OF PROPOSED METHODOLOGY

The proposed methodology of implementing the RBF network to solve the ELD problem is shown in Figure 1. The training data based on the selected test systems for different power demands with varying weights are set by the Lambda Iteration (LI) method. The values generated should be capable of satisfying all load profiles.

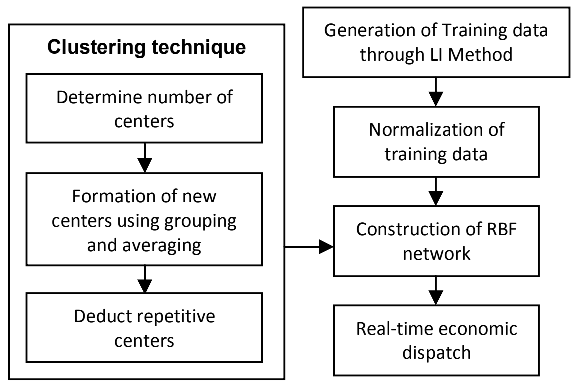

Figure 1. Schematic of proposed methodology

4.1 RBF Network

A typical RBF network model (Figure 2) consists of three layers, the input, hidden and the output layers [(Sivanandam S. N. 2006)]. The nodes within each layer are fully connected to the previous layer. The input nodes pass the incoming input vector directly to the hidden nodes without weights. The connections between the input nodes and the hidden nodes are called the first layer connections. The Gaussian functions are chosen as the activation function in the hidden units. The connections between the hidden nodes and the output nodes are weighted and are called second layer connections. The Gaussian activation function $\phi_j(X)$ for RBF networks is given by Equ. 4.

$$\phi_j(X) = \exp[-(X-\mu_j)^T \sum_{j=1}^{L} (X-\mu_j)] \qquad [4]$$

where, X = input feature vector, L = number of hidden units, μ_j = mean vector of the j^{th} Gaussian function, and $\sum_{j=1}^{L}(X-\mu_j)$ = covariance matrix of the j^{th} Gaussian function.

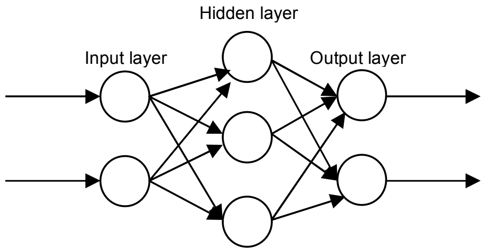

Figure 2. Architecture of RBF network

The output layer implements a weighted sum of hidden-unit outputs as given by Equ. 5:

$$\psi_k(X) = \sum_{j=1}^{L} \lambda_{jk}\phi_j(X) \text{ , for k=1,…, M} \qquad [5]$$

where, M = number of output units, λ_{jk} = output weights, $\phi_j(X)$ = Gaussian activation function, j = 1,2,…L, whwre L is the number of hidden units and k = number of output units. The centers for the radial basis functions are chosen from the set of input training data. A sufficient number of centers have to be selected in order to ensure adequate sampling of the input vector space. The output of i_m unit $v_i(x_i)$ in the hidden layer is calculated from the equation

$$v_i(x_i) = e(-\sum_{j=1}^{r}[x_{ji} - \hat{x}_{ji}]^2 / \sigma_1^2) \qquad [6]$$

where, x_{ji} = center of the RBF unit for input variables, σ_i = width of the i^{th} RBF unit and $\hat{x}_{ji}$ = j^{th} variable of input pattern. The output of the neural network is computed by using the equation

$$y_{net} = \sum_{i=1}^{H} w_{im} v_i(x_i) + w_0 \qquad [7]$$

where, y_{net} = output value of m^{th} node in output layer for the n^{th} incoming pattern, H = number of hidden layer nodes, w_{im} = weight between i^{th} RBF unit and m^{th} output node and w_0 = biasing term at n^{th} output node. The error rate E is calculated as the difference between the achieved and desired outputs for all output patterns and nodes using

$$E = \frac{1}{2}\sum_n \sum_k y_k(x^n) - (t_k^n)^2 \qquad [8]$$

where, n = number of input patterns, k = sum of the values for each output node, $y_k(x^n)$ is the achieved output for the given input x^n and t_k^n are the desired output for the given input n. The iterations are continued until the stopping condition is reached, which may be the weight change in the hidden layer or number of epochs.

4.2 Fuzzy c-means clustering

The choice of selecting the number of hidden units in a neural network is one of the most challenging tasks, requiring more experimentation. Application of clustering methods requires the number of known clusters in advance. There are two options for clustering – validity measures and compatible clustering. The data samples are clustered several times, each time with a different number of clusters $k \in [2, n]$ validity measures, while in compatible clustering, the algorithms starts with a large number of clusters then proceeding by gradually merging similar clusters to obtain fewer clusters. [(Ke Meng 2010)]. Inorder to validate the non-linearity of the system, the value of *k* should be large enough.

In this paper, a fuzzy c-means clustering approach is adopted to specify the range of hidden layer neurons in the RBF network. Consider $x_i \in \Re$ be the data patterns in the feature space. Let the initial cluster number be $k = n/2$, and test whether a new center should be added based on the performance of the network. The new cluster center c_{k+1} is added from the remaining samples $[c_1, c_2, \ldots, c_k]$. The fuzzy membership matrix is then updated with new centers and the process is repeated until the condition $k < n$ is reached. The clustering algorithm is performed by solving

$$Min.J_m(u, c; x) = \sum_{i=1}^{n}\sum_{j=1}^{k} u_{ji}^m \left\| x_i - c_j \right\|^2 \qquad [9]$$

$$\text{Subject to } \begin{cases} u = [u_{ji}], u_{ji} \in [0,1] \\ \sum_{j=1}^{k} u_{ji} = 1, \sum_{j=1}^{k} u_{ji} > 0 \end{cases}, \begin{cases} i = 1,2,\cdots,n \\ j = 1,2,\cdots,k \end{cases}$$

The algorithm of the FCM is as follows

Step 1: For the given data set, initialise $k \in [n/2, n]$, tolerance $\varepsilon > 0$, initial cluster center c_0, fuzzification constant m, such that $1 < m < \infty$. If $m \to 1$, the membership degrees of the data pattern tend to be either 0 or 1 thus approaching the hard means clustering, and if $m \to \infty$, the membership degrees of the data pattern tend to $1/k$, leading to a high level of fuzziness. Based on several experiments, the most common optimal choice of m is 2.

Step 2: Calculate $u(t) = [u_{ji}(t)]$, where $u_{ji}(t)$ is the membership value of vector x_i to the cluster center c_j; with Euclidean distance $d_{ji} = \left\| x_i - c_j \right\|^2$ between x_i and c_j,

$$u_{ji}(t) = \frac{1}{\sum_{r=1}^{k} \left\{ \left[\frac{d_{ji}(t-1)}{d_{ri}(t-1)} \right]^{\frac{2}{m-1}} \right\}} \qquad [10]$$

Step 3: Compute the center $c(t)$, given $c = [c_1, c_2, \ldots, c_k]$ is the array of clusters for $\forall j$,

$$c_j(t) = \frac{\sum_{i=1}^{n}[u_{ji}(t)]^m x_i}{\sum_{i=1}^{n}[u_{ji}(t)]^m} \qquad [11]$$

Step 4: Test for stopping condition else go to step 2. The stopping condition may be maximum number of iterations or until the condition $\left\| c(t) - c(t-1) < \varepsilon \right\|$ is met.

4.3 Parameters

From equations 5 and 6 the major governing parameters for implementing the mapping of the RBF network are

- Number of centers in the hidden layer
- Position of the RBF centers

- Width of the RBF centers
- Weights applied to the RBF function outputs as they are passed to the summation layer

The number of hidden neurons or equivalently radial basis centers needs to be much larger than the number of clusters in the data. The choice of number of hidden neurons is determined through the FCM algorithm. The output of the hidden neuron is significant only if the Euclidean distance from the cluster center is within a radius of $2\sigma_i$ around the cluster center. The width of the RBF centers are set once the clustering procedure is complete satisfying the condition that the basis functions should overlap to some extent in order to give a relatively smooth representation of the data. Typically, the width for a given cluster center is set to the average Euclidean distance between the center and the training vectors which belong to that cluster.

4.4 Algorithm

The application of RBF network consists of two phases, training and testing. The accuracy of RBF network model depends on the proper selection of training data. The inputs of the training network are power demand, weights w_1 and w_2, while the outputs constitute the power generated by the generating units. The step-by-step procedure involved in the implementation of ELD using FCM based RBF network is elaborated below:

Step 1: The data set is divided into training, and testing sets to evaluate the proposed network performance.

Step 2: Initialize suitable values for the range of cluster, initial cluster center, tolerance value for FCM, and number of maximum iterations.

Step 3: Compute the membership matrix and update iteratively based on,

$$u_{ji}(t) = \frac{1}{\sum_{r=1}^{k}\left\{\left[\frac{d_{ji}(t-1)}{d_{ri}(t-1)}\right]^{\frac{2}{m-1}}\right\}} \qquad [12]$$

Similarly, the clusters center matrix (Eqn 11) is computed and updated. If the maximum number of iterations or the tolerance level has reached then the clustering process stops.

Step 4: Compute the cluster radius and weights between the hidden layer and output layer. The feasible results based on the training and testing data are saved and the performance metric Average Percentage Absolute Error (APAE) is computed,

$$APAE\ \% = \frac{1}{m}\sum_{i=1}^{m}\frac{\left|Actual\ \ Output\ - Estimated\ \ Output\right|}{Actual\ \ Output}\times 100 \qquad [13]$$

where m is the number of generating units.

Step 5: Based on the current membership matrix, new cluster centers c_{k+1} are determined using

$$Min \sum_{1\le i,j\le k, i\ne j}(u_{ni} - u_{nj}) \qquad [14]$$

Go to Step 3.

Step 6: The center model that produces minimum error is selected and the output results are computed based on the testing data.

Figure 3 shows the steps involved in solving ELD problem using RBF network. The parameters such as cost coefficients a_i, b_i and c_i, minimum and maximum power generated in the i^{th} unit, $P_{Gi\min}$ and $P_{Gi\max}$, are given as input to the input nodes. Along with the input parameters, the test data of the inputs are also provided. While propagating along the hidden layers, the weights are updated and the centers are chosen using random selection method. The network is trained through the training algorithm and the error values are computed. The difference between the target and the trained data are computed. If the difference is below the tolerance value, the algorithm is stopped and the results are displayed, otherwise the process is repeated until the error converges. The accuracy of the RBF network also depends upon the proper selection of the training data. The more uniform the training data are distributed, the faster the network converges thus providing the optimal solution.

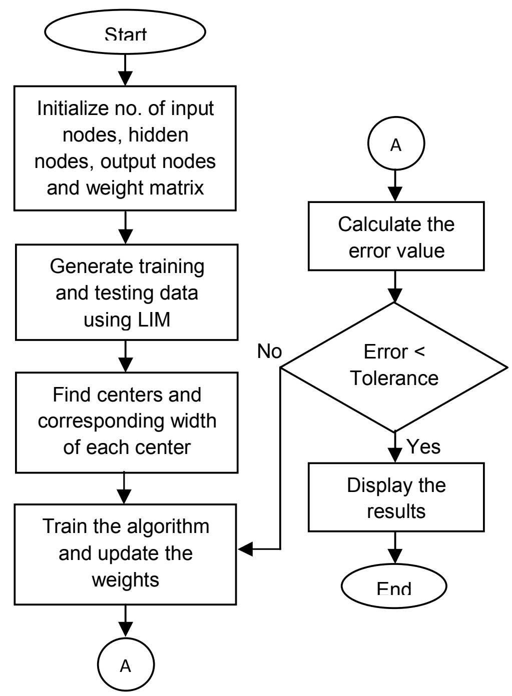

Figure 3. Flow chart of ELD using RBF network

5. EXPERIMENTAL RESULTS

Experimental results show the applicability and effectiveness of a real time project. The main objective of the economic dispatch is to minimize fuel costs while satisfying constraints such as power balance equation and generating power limit for each unit. The pertinence and practicality of the FCM based RBF network for solving Economic Load Dispatch (ELD) problem has been tested on two different power generating units – the IEEE 30 bus 6 units and the 20 units system including the transmission losses. The solution to ELD is obtained through LI method and further through the CM based RBFN. The algorithms are implemented in MATLAB R2008b platform on i3, 2.53 GHz, 4 GB RAM personal computer.

5.1 CASE I: IEEE 30 Bus system

The IEEE 30 bus six unit test system has been adopted from [(Sailaja Kumari M. 2009)], in which the fuel cost coefficients, and power limits are known. The specifications of the system for six generator test system are detailed in Table 1. The system is found to have minimum and maximum generation capacity of 117 MW and 435 MW, respectively.

Table 1. Fuel cost coefficients and power limits for six unit test system

Unit no.	a_i (\$/hr)	b_i (\$/MW hr)	c_i (\$/MW^2 hr)	P_{Gimax} (MW)	P_{Gimin} (MW)
1	.00375	2	0	50	200
2	.01750	1.75	0	20	80
3	.06250	1	0	15	50
4	.00834	3.25	0	10	35
5	.02500	3	0	10	30
6	.02500	3	0	12	40

The transmission loss coefficient denoted as B_{ij} is given according to Equ. 15 as,

$$B_{ij} = \begin{bmatrix} 0.000218 & 0.000103 & 0.000009 & -.000010 & 0.000002 & 0.000027 \\ 0.000103 & 0.000181 & 0.000004 & -.000015 & 0.000002 & 0.000030 \\ 0.000009 & 0.000004 & 0.000417 & -.000131 & -.000153 & -.000107 \\ -.000140 & -.000015 & -.000131 & 0.000221 & 0.000094 & 0.000050 \\ 0.000002 & 0.000002 & -.000153 & 0.000094 & 0.000243 & 0.000000 \\ 0.000027 & 0.000030 & -.000107 & 0.000050 & 0.000000 & 0.000358 \end{bmatrix}$$

[15]

In the LI Method, the program does not impose any restriction on the range of the lambda in order to obtain optimal distribution of power among the power generating units. For experimental analysis, the power demand for the IEEE 30 bus system was varied between 117 MW to 400 MW with random intervals and generated power in each unit, the total cost, total losses and computational time were evaluated. The lambda value is chosen based on the derivative of the cost function in order to achieve better convergence. The rate of change of lambda $\Delta\lambda$ is chosen as 0.00005 in this study. Table 2 shows the computed results for the 6-unit test system using LIM Method.

The accuracy of RBF network model depends on the proper selection of training data. The inputs of the training network are power demand, weights w_1 and w_2, while the outputs constitute the power generated by the 6 generating units. Table 3 shows the various parameters and their values used in RBFN based ELD.

The learning rate (α) controls the rate at which the weights are modified due to previous weight updates. It acts as a smoothing parameter that reduces oscillation and helps attain convergence. This must be a real value between 0.0 and 1.0. In this experiment, convergence was attained for α = 0.997. The step size controls the weights during the training process, larger the learning rate, larger the rate of change of weights. Hence to maintain stability in the updation of weights, the value of 0.0002 was chosen.

Table 3 Parameters of ANN used to implement ELD for six unit system

Parameters	Notations used	Values
Initial cluster number	k	3
Fuzzification constant	m	2
Input Nodes	Input node	3
Output Nodes	Output node	6
No. of patterns	n	171
No. of RBF centers	Centers	55
Momentum factor	m	0.0002
Learning rate	α	0.997
Step size/tolerance	ϵ	0.002
No. of iterations	Iter	500

The training data are generated using lambda iteration method, by changing the total power demand in from minimum to maximum generation capacity taking into account the generator power limits and transmission power losses. A total of 171 training samples were created in this case and the 5.2% of the training data was chosen as testing data in a trial and error basis.

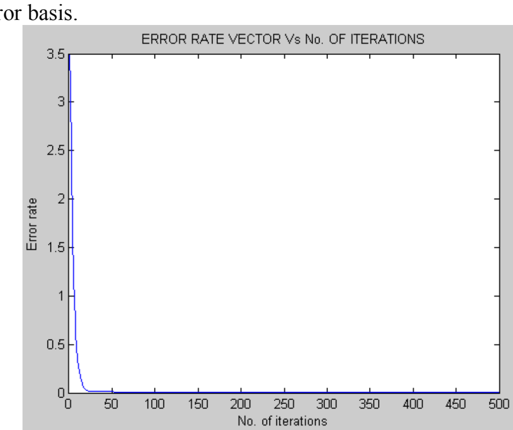

Figure 4. Error Rate Vs No. of Iterations

Figure 4 shows the typical relationship between the number of iterations and the error rate for the 6 unit generator system. While increasing the number of iterations the error rate decreases and becomes constant after a set of iterations. The optimized results were obtained when the RBFN converged towards the best value at the end of 500 iterations. Table 4 shows the computational results of the RBFN for 6 unit generator system for different values of power demand.

In order to verify the effectiveness of RBF in solving Economic Load Dispatch problems, the results obtained in the above sections are compared with those obtained through literature in terms of cost, total power, loss, algorithmic efficiency and computational time for a power demand of 283.4 MW (Table 5). It is clear that the proposed Fuzzy c-means based RBFN is superior to the other techniques for the IEEE 30 bus system.

Table 2 Results using LIM for six generator test system

Power demand (MW)	117	150	200	250	283.4	300	350	400
P1 (MW)	50	68.25161	107.6139	147.4569	174.3403	187.782987	200	200
P2 (MW)	20	30.40913	40.0768	50.05122	56.89421	60.351453	75.66692	80
P3 (MW)	15.90416	21.0729	24.16534	27.40669	29.66026	30.808059	35.73562	43.01804
P4 (MW)	10	10	10	10	10	10	26.8413	35
P5 (MW)	10	10	10	10	10	10	12.48831	29.5571
P6 (MW)	12	12	12	12	12	12	12	25.75381
Fuel Cost ($/hr)	288.5221	377.6314	522.9947	687.6676	808.9491	872.718503	1072.37	1288.999
Total Power (MW)	117.9042	151.7336	203.856	256.9148	292.8948	310.942499	362.7322	413.329
Power loss (MW)	0.90345	1.731668	3.853857	6.908351	9.48889	10.937331	12.72418	13.32627
CPU time (s)	15.8281	22.25	23.9219	24.2969	25.9063	30.0156	33.6406	34.9063
Lambda	2.0062	2.34235	2.5464	2.7617	2.91225	2.9892	3.30145	3.7501

Table 4 Results using ANN for six generator test system

Power demand	117	150	200	250	283.4	300	350	400	435
w1	0.5	0.15	0.6	0.7	0.35	0.85	0.45	0.45	0.95
w2	0.5	0.85	0.4	0.3	0.65	0.15	0.55	0.55	0.05
P1	49.974	77.788	116.846	152.132	184.177	195.126	201.834	198.632	195.810
P2	21.0196	27.791	37.0151	43.3293	50.9106	55.4441	75.6078	76.7924	84.6246
P3	15.8205	17.6391	15.4882	17.758	22.0863	23.0039	19.7934	33.9414	50.1861
P4	9.4746	10.5882	8.3501	10.9038	9.9015	9.9048	12.0989	14.6543	31.7837
P5	10.665	10.816	9.7967	10.4438	12.7022	12.629	23.509	18.1916	30.2631
P6	12.5306	12.6609	12.8123	12.3629	10.6345	13.317	23.0391	24.5869	43.1272
FUEL COST	293.188	390.212	506.175	648.601	792.451	859.199	1040.34	1089.89	1417.71
Total Power	119.4838	157.2832	200.3086	246.93	290.4117	309.4253	355.8818	366.799	435.795
Power loss	0.3116	0.5056	0.7704	1.1355	1.5455	1.7628	2.5199	2.6669	4.0361
CPU time	0.886	0.783	0.924	1.092	0.985	0.889	0.9864	1.027	1.031
APAE	-1.44332	-5.83526	2.61625	-0.49836	-3.67958	-4.19765	-1.55165	9.728697	-0.59397

Table 5 Comparative Analysis

Parameters	LIM	Hybrid GA [(Mary 2004)]	EP [(J. Yuryevich 1999)]	Simple GA [(Sailaja Kumari M. 2009)]	Fast GA [(Sailaja Kumari M. 2009)]	PS [(Y. LABBI 2010)]	GA- PS [(Y. LABBI 2010)]	Proposed RBFN
P_{G1} (MW)	174.3403	176.2358	176.1522	189.5200	189.6130	175.727	175.6627	184.1766
P_{G2} (MW)	56.89421	49.0093	48.8391	47.7240	47.7450	48.6812	48.6413	50.9106
P_{G3} (MW)	29.66026	21.5023	21.5144	19.5719	19.5761	21.4282	21.4222	22.0863
P_{G4} (MW)	10.0000	21.8115	22.1299	13.8642	13.8752	22.8313	22.6219	9.9015
P_{G5} (MW)	10.0000	12.3387	12.2435	10.0000	10.0000	12.0667	12.3806	12.7022
P_{G6} (MW)	12.0000	12.0129	12.0000	12.0000	12.0000	12.0000	12.0000	10.6345
Fuel cost ($/hr)	808.9491	802.465	802.404	799.3840	799.8230	802.0150	802.0138	792.4514
Total power P_G (MW)	292.8948	292.9105	292.8791	292.6801	292.8093	292.7344	292.7287	290.4117
Power loss (MW)	9.48889	9.5105	9.4791	9.6825	9.6897	9.3349	9.3286	1.5455
CPU time (s)	25.9063	NA	NA	0.483	0.125	NA	NA	0.985

*NA – Data Not Available

5.2 CASE II: 20 UNIT TEST SYSTEM

In order to demonstrate the effectiveness of the algorithms, several tests have been performed on a benchmark consisting of twenty generator units [(Ching-Tzong Su 2000)]. The details of fuel cost coefficients and generating limits for each unit are given in Table 6. The maximum and minimum power generating limits of the system are 3865 MW and 1010 MW, respectively. The experiments were conducted using the conventional Lambda Iteration Technique and the RBF network by varying the power demand within the range [1010, 3865].

The Transmission Loss Coefficient Matrix for calculating power loss of 20 Unit test system can be obtained from [(Ching-Tzong Su 2000)]. Table 7 illustrates results of lambda iteration method for the twenty unit system such as the generated power of each unit, the fuel cost, power loss and CPU time for various values of power demand.

The structural design of the RBFN is modified for 20 unit test system with three input nodes, four hidden nodes and twenty output nodes. The twenty output nodes correspond to optimal power generated for each generating units and three input nodes represents weights w_1 and w_2, and power demand. The RBF network was trained with 133 patterns generated through LIM method for 500 iterations with network parameters initialized as shown in Table 8. In this experiment, 56 centers were selected in random with a learning rate of 0.997 and step size of 0.002 through the Fuzzy c-means algorithm. The momentum factor controls the number of weights changed during the updation process and also acts as a smoothing parameter that reduces oscillation and helps attain convergence. This must be a real value between 0.0 and 1.0, and was set to 0.0002. Step size is the tolerance value in the range [0.0, 1.0], which decides the acceptable difference between the desired output value and the actual output value. Since this is the deciding parameter, it was set to a very low value 0.002 in this study.

Table 6 Fuel cost coefficients and power limits for twenty unit test system

Unit no.	a_i (\$/hr)	b_i (\$/MW hr)	c_i (\$/MW^2 hr)	P_{Gimax} (MW)	P_{Gimin} (MW)
1	0.00068	18.19	1000	600	150
2	0.00071	19.26	970	200	50
3	0.00650	19.80	600	200	50
4	0.00500	19.10	700	200	50
5	0.00738	18.10	420	160	50
6	0.00612	19.26	360	100	20
7	0.0079	17.14	490	125	25
8	0.00813	18.92	660	150	50
9	0.00522	18.27	765	200	50
10	0.00573	18.92	770	150	30
11	0.00480	16.69	800	300	100
12	0.00310	16.76	970	500	150
13	0.00850	17.36	900	160	40
14	0.00511	18.70	700	130	20
15	0.00398	18.70	450	185	25
16	0.00712	14.26	370	80	20
17	0.0089	19.14	480	85	30
18	0.00713	18.92	680	120	30
19	0.00622	18.47	700	120	40
20	0.00773	19.79	850	100	30

From Figure 5, it is shown that error rate decreases with increase in number of iterations and finally attains a constant value at zero. Results using ANN for twenty unit test system such as the power generated in each unit, power loss, and CPU time for various values of power demand are shown in Table 9.

Computational results of both Lambda iteration method and the Radial basis function method are compared in terms of generated power per unit, fuel cost, total power generated, power loss and computational time with LI method, and results obtained through other algorithms in literature. The results are illustrated in Table 10 for a power demand of 2500 MW.

5.3 Summary of discussions

The results obtained for the 6 unit and the 20 unit systems have proved that the Fuzzy c-means based RBF is more efficient in producing the optimal dispatch when compared with LI Method. The consequences of the output based on the solution quality, and computational efficiency are summarized in this section.

Solution quality: From the results obtained through the 6 unit test system in Table 5, for a power demand of 283.4 MW, it is noticed that the optimized fuel cost obtained by RBFN is 0.848%, less than LIM. Likewise, for the 20 unit system from Table 10, the minimum cost obtained by LIM is 0.0323% higher than the cost obtained through RBF for a power demand of 2500 MW.

Computational efficiency: Apart from yielding the optimal solution, it may also be noted that RBFN yields the minimum cost (Table 5 and Table 10) at a comparatively lesser time of execution. Computational efficiency of FCM based RBFN is 96.19% higher than LIM in case of a power demand of 283.4 MW for 6 unit test system. Similarly, for twenty unit test system, RBFN has higher computational efficiency by the factor of 97.09%, for power demand of 2500 MW. Thus, the FCM based RBF approach is more efficient than Lambda iteration method in terms of computational time.

Table 8 Parameters of ANN used to implement ELD for twenty unit system

Parameters	Notations used	Values
Initial cluster number	K	3
Fuzzification constant	M	2
Input Nodes	I	3
Hidden Nodes	H	4
Output Nodes	O	20
No. of patterns	N	133
No. of RBF centers	Centers	56
Learning rate	α	0.997
Momentum factor	M	0.0002
Step size	ϵ	0.002
No. of iterations	Iter	500

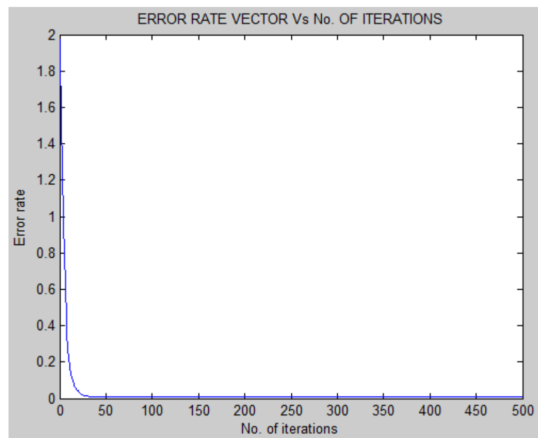

Figure 5 Error Rate Vs No. of Iterations

6. CONCLUSION

Economic load dispatch (ELD) in electric power system is the task of allocating generation among the committed units thus minimizing the total cost of production subject to the system equality and inequality constraints. For the considered ELD systems including transmission losses, FCM based RBF found solutions better than the conventional lambda iteration method in terms of fuel cost, computational time, and power loss. It was observed that in all the conducted experiments, the average performance of RBF was exceptional and the required time proved that FCM based RBF is most suitable for online solving of ELD problems. In future, efforts will be taken to impose more realistic constraints on the problem structure and large size real-time problems would be attempted by the proposed methodology. It would be of considerable interest to incorporate several practical constraints such as security, emission, and fuel reserve to the ELD problems and addition of these constraints while solving the ELD problem will be the subject of future works in this area.

Table 7 Results using LIM for twenty generator test system

Power demand (MW)	1010	1500	2000	2500	3000	3500
Lambda (λ)	15.02	19.45585	19.97445	20.3892	20.8129	21.9647
Fuel cost ($/MW hr)	34411.52	41800.38	51919.58	63295.81	72977.61	84881.91
P_{G1} (MW)	150	233.4626	350.5274	470.6366	577.9241	600
P_{G2} (MW)	50	50	50	50	126.3088	200
P_{G3} (MW)	50	50	81.51073	151.1845	200	200
P_{G4} (MW)	50	50	53.72806	97.11856	145.7151	200
P_{G5} (MW)	50	50	71.58732	97.77008	123.4743	160
P_{G6} (MW)	20	20	32.87442	55.68459	80.24465	100
P_{G7} (MW)	25	115.1665	125	125	125	125
P_{G8} (MW)	50	53.46564	115.3661	150	150	150
P_{G9} (MW)	50	50	50	68.82129	101.04	200
P_{G10} (MW)	30	37.78153	118.283	150	150	150
P_{G11} (MW)	100	152.9464	187.0447	194.5108	195.673	236.1011
P_{G12} (MW)	150	292.7372	317.9294	337.2191	357.3664	453.7171
P_{G13} (MW)	40	108.5256	135.8477	151.1625	160	160
P_{G14} (MW)	20	20	20	20	20	59.1824
P_{G15} (MW)	25	51.82681	78.91401	103.9979	134.5984	185
P_{G16} (MW)	36.0695	80	80	80	80	80
P_{G17} (MW)	30	30	30	51.67328	85	85
P_{G18} (MW)	30	30	72.24207	98.43284	120	120
P_{G19} (MW)	40	40	68.32917	98.48716	120	120
P_{G20} (MW)	30	30	30	42.17147	74.26862	100
Total P_G (MW)	1026.069	1545.912	2069.184	2593.871	3126.614	3684.001
Power loss (MW)	16.06525	45.90553	69.13084	93.83006	126.5954	183.9935
CPU time (s)	1023.8	1251.9	1340.1	1232.1	1231.9	1291.7

Table 9 Results using ANN for twenty generator test system

Power demand (MW)	**1010**	**1500**	**2000**	**2500**	**3000**	**3500**	**3865**
w_1	0.25	0.3	0.9	0.55	0.55	0.65	0.1
w_2	0.75	0.7	0.1	0.45	0.45	0.35	0.9
Fuel cost ($/MW hr)	34058.68	41570	49993.4	62436.46	72941.23	84123.39	87405.29
P_{G1} (MW)	147.1636	290.4817	406.9764	472.7972	578.4828	595.7565	598.2086
P_{G2} (MW)	53.8524	56.9833	89.0204	164.8213	205.4831	213.2504	202.5966
P_{G3} (MW)	53.0988	52.6563	67.3398	111.0902	158.9411	202.1239	198.7915
P_{G4} (MW)	47.1719	40.9439	42.7814	72.2386	115.8517	185.9402	186.3426
P_{G5} (MW)	52.5952	66.3248	85.3925	120.02	142.5125	148.9922	182.1915
P_{G6} (MW)	26.829	25.4111	36.6882	75.8967	92.8074	102.6464	100.2029
P_{G7} (MW)	29.9063	81.4288	102.3298	118.6989	121.3111	138.4964	128.027
P_{G8} (MW)	53.0762	50.0647	73.4457	112.5673	135.2231	151.8175	128.7755
P_{G9} (MW)	49.6017	49.6433	75.1298	87.6202	119.9089	160.1948	190.9665
P_{G10} (MW)	33.1905	51.0884	90.0773	118.7859	138.5096	145.5974	150.5402
P_{G11} (MW)	102.8683	138.7033	142.853	161.9814	184.8945	253.5285	307.1777
P_{G12} (MW)	153.2964	226.4354	260.8681	270.073	327.8716	389.6394	488.7123
P_{G13} (MW)	48.9157	75.9085	86.268	87.7971	109.9513	164.2233	145.4075
P_{G14} (MW)	31.6477	29.7433	34.0271	52.0309	59.6217	102.6602	127.2653
P_{G15} (MW)	34.4998	40.1871	74.2373	99.4377	130.3887	188.2335	171.8812
P_{G16} (MW)	27.428	79.6814	85.4802	89.0737	77.6386	76.8944	88.1829
P_{G17} (MW)	33.257	28.1896	36.5791	73.3581	77.5748	82.7333	85.046
P_{G18} (MW)	35.9433	42.3985	59.4482	93.9469	111.8516	129.4469	118.6177
P_{G19} (MW)	55.8547	68.2277	93.7277	145.414	150.2787	146.5659	157.8982
P_{G20} (MW)	40.2033	35.9105	29.9237	76.194	95.952	116.3288	110.0647
Total power P_G (MW)	1110.4	1530.4	1972.6	2603.8	3135.1	3695.1	3866.9
Power loss (MW)	18.0642	40.3534	63.0818	92.3204	129.7901	181.5548	215.0577
APAE %	-0.26045	-1.26773	6.861934	-1.53346	0.105998	-1.83827	-1.2176
CPU time (s)	0.12	0.678	0.799	0.982	1.145	1.923	1.989

Table 10 Comparative Analysis

Parameters	LIM	Hopfield [(Ching-Tzong Su 2000)]	BBO [(Aniruddha Bhattacharya 2010)]	RBFN
P_{G1} (MW)	512.7805	512.7804	513.09	472.7972
P_{G2} (MW)	169.1033	169.1035	173.35	164.8213
P_{G3} (MW)	126.8898	126.8897	126.92	111.0902
P_{G4} (MW)	102.8657	102.8656	103.33	72.2386
P_{G5} (MW)	113.6836	113.6836	113.77	120.02
P_{G6} (MW)	73.5710	73.5709	73.07	75.8967
P_{G7} (MW)	115.2878	115.2876	114.98	118.6989
P_{G8} (MW)	116.3994	116.3994	116.42	112.5673
P_{G9} (MW)	100.4062	100.4067	100.69	87.6202
P_{G10} (MW)	106.0267	106.0267	100	118.7859
P_{G11} (MW)	150.2394	150.2395	148.98	161.9814
P_{G12} (MW)	292.7648	292.7647	294.02	270.073
P_{G13} (MW)	119.1154	119.1155	119.58	87.7971
P_{G14} (MW)	30.8340	30.8342	30.55	52.0309
P_{G15} (MW)	115.8057	115.8056	116.45	99.4377
P_{G16} (MW)	36.2545	36.2545	36.23	89.0737
P_{G17} (MW)	66.8590	66.8590	66.86	73.3581
P_{G18} (MW)	87.9720	87.9720	88.55	93.9469
P_{G19} (MW)	100.8033	100.8033	100.98	145.414
P_{G20} (MW)	54.3050	54.3050	54.27	76.194
Fuel cost ($/ hr)	62456.6391	62456.6341	62456.79	62436.46
Total power P_G (MW)	2537.662	2591.967	2592.11	2603.8
Power loss (MW)	91.9670	91.967	92.11	92.3204
CPU time (s)	33.757	6.355	6.93	0.982

7. REFERENCES

[1] Ching-Tzong Su, Chien-Tung Lin. "New Approach with a Hopfield Modeling Framework to Economic Dispatch." *IEEE Transactions on Power Systems* vol. 15, no. 2 (May 2000): 541 - 545.

[2] Aniruddha Bhattacharya, P.K. Chattopadhyay. "Solving complex economic load dispatch problems using biogeography-based optimization." *Expert Systems with Applications* vol.37 (2010): 3605-3615.

[3] Aravindhababu P., Nayar K.R. "Economic dispatch based on optimal lambda using radial basis function network." *Journal on Electrical Power and Energy systems* vol. 24 (August 2001): 551-556.

[4] C.L., Wadhwa. *Electrical Power Systems.* New Delhi: New Age International (p) Limited Publishers, 2000.

[5] Chao-Ming Huang, Fu-Lu Wang. "An RBF Network With OLS and EPSO Algorithms for Real-Time Power Dispatch." *IEEE Trans. Power Systems* vol. 22, no. 1 (February 2007): 96-104.

[6] Chih-Cheng Hung, Youngsup Kim, Coleman, T.L. "A comparative study of radial basis function neural networks and wavelet neural networks in classification of remotely sensed data." *IEEE Proc. of 5th Biannual World Automation Congress.* IEEE, 2002. 455 - 461.

[7] Djukanovic.M., Calovic.M.,Milosevic.B.,Sobajic.DJ. "Neural-net based real time economic dispatch for therma power plants." *IEEE Trans. Energy Conversion* vol. 11, no. 44 (1996): 755-762.

[8] Glover, Z. X. Liang and J. D. "A zoom feature for a dynamic programming solution to economic dispatch including transmission losses." *IEEE Trans. on Power Systems* vol. 7, no. 2 (May 1992): 544-550.

[9] H. T. Yang, P. C. Yang and C. L. Huang. "Evolutionary Programming Based Economic Dispatch For Units With Non-smooth Fuel Cost Functions." *IEEE Transactions on Power Systems* vol. 11, no. 1 (1996): 112-118.

[10] J. B. Park, K. S. Lee, J. R. Shin and K. Y. Lee. "A particle swarm optimization for economic dispatch with non smooth cost functions." *IEEE Trans. on Power Systems* vol. 8, no. 3 (August 1993): 1325-1332.

[11] J. Yuryevich, K. P. Wong. "Evolutionary Programming Based Optimal Power Flow Algorithm." *IEEE Transaction on power systems* vol. 14, no. 4 (November 1999): 1245 - 1250 .

[12] Ke Meng, Zhao Yang Dong, Dian Hui Wang, Kit Po Wong. "A Self-Adaptive RBF Neural Network Classifier for Transformer Fault Analysis." *IEEE Trans. Power Systems* vol. 25, no. 3 (August 2010): 1350-1360.

[13] Lakshmi Devi A., Vamsi Krishna O. "Combined economic and emission dispatch using Evolutionary algorithms-a case study." *ARPN Journal of Engineering and Applied Sciences* 3, no. 6 (December 2008): 28-35.

[14] Lowery, P. G. "Generation unit commitment by dynamic programming." *IEEE Trans. Power App.Syst.,* vol. PAS-102 (1983): 1218–1225.

[15] Mary, N. Thenmozhi and D. "Economic emission load dispatch using hybrid Genetic Algorithm." *Chiang Mai, Thailand.* 2004. 476-479.

[16] Matuda.S., Akimoto.Y.,. "The representatio of large numbers in neural networks and its applications to economic load dispatching of electric power." *ICNN.* 1989. 587-592.

[17] Powell.M.J.D. "Restart Procedures for the conjugate gradient method." *Mathematical Programming* vol. 12 (1977): 241-254.

[18] Rayapudi, S. Rao. "An Intelligent Water Drop Algorithm for Solving Economic Load Dispatch Problem." *International Journal of Electrical and Electronics Engineering* vol. 5, no. 1 (2011): 43-49.

[19] Roa-Sepulveda C.A., Herrera M., Pavez-Lazo B., Knight U.G., Coonick A.H. "Economic dispatch using fuzzy decision trees." *Electric Power Systems Research* vol.66, no. 2 (August 2003): 115-122.

[20] Rollet, R., G. B. Benie, W. Li, and S. Wang. "Image classification algorithm based on the RBF neural network and K-means." *International Journal of Remote Sensing* vol. 19, no. 15 (1998): 3003-3009.

[21] Sailaja Kumari M., Sydulu M. "A Fast Computational Genetic Algorithm for Economic Load Dispatch." *International Journal of Recent Trends in Engineering* vol. 1, no. 1 (May 2009): 349-356.

[22] Singh. G., Srivastava.S.C.,Kalra.P.K.,Vinod Kumar.D.M. "Fast approach to artificial neural network training and its application to economic load dispatch." *Electrical Machines and Power Systems*, 1995: 13-24.

[23] Sivanandam S. N., Sumathi S., Deepa S.N. *Introduction to Neural networks using MATLAB 6.0.* New Delhi: Tata McGraw-Hill Publishing Company Limited, 2006.

[24] T. Jayabarathi, G. Sadasivam and V. Ramachandran. "Evolutionary programming based economic dispatch of generators with prohibited operating zones." *Electric Power Systems Research* vol. 52, no. 3 (1999): 261-266.

[25] W.-M. Lin, F.-S. Cheng, and M.-T. Tsay. "An improved tabu search for economic dispatch with multiple minima." *IEEE Trans. Power Syst* vol. 17 (February 2002): 108 - 112.

[26] Wilson, J. A. Muckstadt and R. C. "An application of mixed-integer programming duality to scheduling thermal generating systems." *IEEE Trans. Power App. Syst* vol. PAS-87, no. 12 (1968): 1968-1978.

[27] Wong, K. P. Wong and Y. W. "Genetic and genetic/simulated-annealing approaches to economic dispatch." *Proc. Inst. Elect. Eng. Gen. Trans. Distrib.* vol. 141 (September 1994): 507-513.

[28] Wood J., Wollenberg B. F. *Power generation operation and control.* John Wiley & Sons, 1984.

[29] Y. Labbi, D. Ben Attous. "A hybrid GA–PS method to solve the economic load dispatch problem" *Journal of Theoretical and Applied Information Technology* vol.15, no. 1 (2010): 61-68.

[30] Yoshimura, A. I. Cohen and M. "A branch-and-bound algorithm for unit commitment." *IEEE Trans. Power App. Syst.* vol. PAS-102, no. 2 (1983): 444–451.