REAL POWER OPTIMIZATION WITH LOAD FLOW USING ADAPTIVE HOPFIELD NEURAL NETWORK

Kwang Y. Lee, Fatih M. Nuroglu, and Arthit Sode-Yome
Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802

Abstract: This paper presents real power optimization with load flow using an adaptive Hopfield neural network. In order to speed up the convergence of the Hopfield neural network system, the two adaptive methods, slope adjustment and bias adjustment, were used with adaptive learning rates. Algorithms of economic load dispatch for piecewise quadratic cost functions using the Hopfield neural network have been developed for the two approaches In stead of using the typical B-coefficient method, this paper uses actual load flow to compute the transmission loss accurately. These methods for optimization has been tested in the IEEE 30-bus system to demonstrate its effectiveness. The performance of the proposed approaches is evaluated by comparing the results of the slope adjustment and the bias adjustment methods with those of the conventional Hopfield network, and an additional improvement was demonstrated by the use of momentum in the adaptive learning approaches.

Keywords: Optimal power flow, economic load dispatch, Hopfield neural networks, adaptive Hopfield neural networks

1. INTRODUCTION

In power system, the operation cost at each time needs to be minimized via economic load dispatch (ELD) [1]. Since the introduction by Hopfield [2,3] The Hopfield neural networks have been used in many different applications. An important property of the Hopfield neural network is the decrease in energy by finite amount whenever there is any change in inputs [4]. Thus, the Hopfield neural network can be used for optimization. Tank and Hopfield [5] described how several optimization problem can be rapidly solved by highly interconnected networks of a simple analog processor, which is an implementation of the Hopfield neural network. Park and others [6] presented the economic load dispatch for piecewise quadratic cost functions using the Hopfield neural network. The results of this method compared very well with those of the numerical method in an hierarchical approach [7]. King and others [8] applied the Hopfield neural network in the economic and environmental dispatching of electric power systems.

These applications, however, involved a large number of iterations and often showed oscillations during transients. . Recently, in order to speed up the convergence of the Hopfield neural networks, Lee and others [9] developed adaptive Hopfield neural network methods, slope adjustment and bias adjustment methods, with adaptive learning rates. In these studies, however, the transmission loss was either calculated by the traditional B-coefficient method or ignored all together. This paper incorporates both the adaptive Hopfield network for faster convergence and the load flow for accurate calculation of the transmission loss.

2. ECONOMIC LOAD DISPATCH

The cost function of ELD problem is defined as follows:

$$C = \sum_i C_i(P_i), \tag{1}$$

$$C_i(P_i) = \begin{cases} a_{i1} + b_{i1} P_i + c_{i1} P_i^2, & \text{fuel 1}, \quad \underline{P}_i \le P_i \le P_{i1} \\ a_{i2} + b_{i2} P_i + c_{i2} P_i^2, & \text{fuel 2}, \quad P_{i1} \le P_i \le P_{i2} \\ \vdots & \\ a_{ik} + b_{ik} P_i + c_{ik} P_i^2, & \text{fuel } k, \quad P_{k\text{-}1} \le P_i \le \overline{P}_i \end{cases}, \tag{2}$$

where
$C_i(P_i)$: cost of the i^{th} generator
P_i : the power output of generator i
a_{ik}, b_{ik}, c_{ik} : cost coefficients of the i^{th} generator for fuel type k.

In minimizing the total cost, the constraints of power balance and power limits should be satisfied:

a) Power balance

The total generating power has to be equal to the sum of load demand and transmission-line loss:

$$D + L - \sum_i P_i = 0 , \quad (3)$$

where D is total load and L is transmission loss

The transmission loss is obtained by load flow. Traditionally, the transmission loss was calculated by the B-coefficient method [1] and, for simplification, it was often assumed zero.

b) Maximum and minimum limits of power

The generation power of each generator has some limits and it can be expressed as

$$\underline{P_i} \le P_i \le \overline{P_i} , \quad (4)$$

where $\underline{P}$ and $\overline{P}$ are respectively the minimum and maximum power generation limits.

3. HOPFIELD NETWORKS FOR ELD

The energy function of the continues-time Hopfield network [3] is defined as

$$E = -\frac{1}{2}\sum_i \sum_j T_{ij} V_i V_j - \sum_i I_i V_i + \sum_i q_i V_i , \quad (5)$$

where V_i, I_i, and q_i are respectively the output, external input and threshold bias of neuron i.

The dynamics of the neurons is defined by

$$\frac{dU_i}{dt} = \sum_j T_{ij} V_j + I_i , \quad (6)$$

where U_i is the total input to neuron i and the sigmoidal function can be defined as

$$V_i = g_i(l\, U_i) = g_i(\frac{U_i}{U_0}) = \frac{1}{1 + \exp\left(-\frac{U_i + q_i}{U_0}\right)} . \quad (7)$$

Stability is known to be guaranteed since the energy function is bounded and its increment is found to be nonpositive. The time evolution of the system is a motion in state-space that seeks out minima in E and comes to a stop at such points.

In order to solve the ELD problem, the following energy function is defined by augmenting the objective function (1) with the constraint (2):

$$E = \frac{1}{2} A(D + L - \sum_i P_i)^2 + \frac{1}{2} B \sum_i (a_i + b_i P_i + c_i P_i^2) ,(8)$$

where a_{ik}, b_{ik}, c_{ik} are the cost coefficients as discrete functions of P_i defined in (1).

By comparing (8) with (5) whose threshold is assumed to be zero, the weight parameters and external input of neuron i in the network [6] are given by

$$\begin{aligned} T_{ii} &= -A - Bc_i , \\ T_{ij} &= -A , \\ I_i &= A(D + L) - \frac{Bb_i}{2} . \end{aligned} \quad (9)$$

From (6), the differential synchronous transition mode used in computation of the Hopfield neural network is

$$U_i(k) - U_i(k-1) = \sum_j T_{ij} V_j(k) + I_i . \quad (10)$$

The sigmoidal function (7) can be modified [6] to meet the power limit constraint as follows:

$$V_i(k+1) = (\overline{P}_i - \underline{P}_i) \frac{1}{1 + \exp\left[-\frac{U_i(k) + q_i}{U_0}\right]} + \underline{P_i} \quad (11)$$

4. ADAPTIVE HOPFIELD NETWORKS

The traditional approach in solving the economic load dispatch (ELD) problem using the Hopfield neural network requires a large number of iterations and often oscillates during the transient [6] and [9]. In order to speed up convergence, two adaptive adjustment methods are developed in this paper: slope adjustment and bias adjustment methods [9].

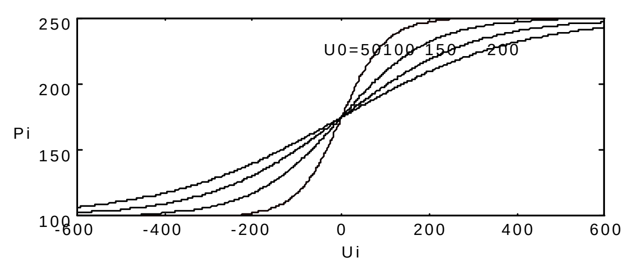

Figure 1. Sigmoidal threshold function with different values of the gain parameter.

4.1 Slope Adjustment Method

In transient state, the neuron input oscillates around the threshold value, zero in Fig. 1. Some neuron inputs oscillate away from the threshold value. If the gain parameter is set too high, the oscillation will occur at the saturation region. If the slope in this region is too low, the neurons can not go to the stable state and will cause instability.

Since energy is to be minimized and its convergence depends on the gain parameter U_0, the gradient-descent method can be applied to adjust the gain parameter as

$$U_0(k+1) = U_0(k) - \eta_s \frac{\partial E}{\partial U_0}, \tag{12}$$

where η_s is a learning rate.

From (8) and (11), the gradient of energy with respect to the gain parameter can be computed as

$$\frac{\partial E}{\partial U_0} = \sum_{i=1}^{n} \frac{\partial E}{\partial P_i} \frac{\partial P_i}{\partial U_0}. \tag{13}$$

The update rule of (12) needs a suitable choice of the learning rate η_s. For speed and convergence, a method to compute adaptive learning rates is developed following the procedure in Ku and Lee [10]. It can be shown [9,10] that the optimal learning rate is

$$\eta_s^* = \frac{1}{g_{s,\max}^2}, \tag{14}$$

where $g_{s,\max} := \max\|g_s(k)\|,\ g_s(k) = \partial E(k)/\partial U_0$.

4.2 Bias Adjustment Method

There is a limitation in the slope adjustment method in that slopes are small near the saturation region of the sigmoidal function, Fig 1. If every input can use the same maximum possible slope, convergence will be much faster. This can be achieved by changing the bias to shift the input near the center of the sigmoidal function. The bias can be changed following the similar gradient-descent method used in the slope adjustment method:

$$\theta_i(k+1) = \theta_i(k) - \eta_b \frac{\partial E}{\partial \theta_i}, \tag{15}$$

where η_b is a learning rate.

The bias can be applied to every neuron as in (7), therefore, from (8) and (11), a derivative of energy with respect to a bias can be individually computed as

$$\frac{\partial E}{\partial \theta_i} = \frac{\partial E}{\partial P_i} \frac{\partial P_i}{\partial \theta_i}. \tag{16}$$

The adaptive learning rate is also developed following the similar procedure [10]. It can be shown that the optimal learning rate is

$$\eta_b^* = -\frac{1}{g_b(k)}, \tag{17}$$

where

$$g_b(k) = \sum \sum T_{ij} \frac{\partial V_i}{\partial \theta} \frac{\partial V_j}{\partial \theta}. \tag{18}$$

Again, any other learning rate larger than η_b^* does not guarantee a faster convergence.

4.3 Momentum

The speed of convergence can be accelerated by adding momentum in the update processes. The momentum can be applied when updating the input in (10), the gain parameter in (12), and the bias in (15):

$$U_i(k) - U_i(k-1) = \sum_j T_{ij} V_j(k) + \alpha_u \Delta U_i(k-1), \tag{19}$$

$$U_0(k) = U_0(k-1) - \eta_s \frac{\partial E}{\partial U_0} + \alpha_s \Delta U_0(k-1), \tag{20}$$

$$\theta(k) = \theta(k-1) - \eta_b \frac{\partial E}{\partial \theta} + \alpha_b \Delta\theta(k-1), \tag{21}$$

where α's are momentum factors.

5. SIMULATION RESULTS

The results of slope adjustment and bias adjustment methods are compared at different load demands. Then the results of the slope and bias adjustment methods with adaptive learning rates and momentum applied are compared with each other. Three load levels are tested: 220 MW, 283.4 MW and 380 MW. The 220 MW and 380 MW load levels were used to force one or more units to hit lower and upper limits. The IEEE 30-bus, 6 generator system is used for loadflow. The line data is given in [13]. All the graphs are based on the nominal load, 283.4 MW, to demonstrate the convergence. A Compaq 90 MHz Pentium computer was used for simulation.

For real power optimization with loadflow, first the load flow is run to obtain initial generator power and the system loss. Second, the adaptive Hopfield networks update the slope and bias parameters and generator power. Then, with the new generator power, load flow is run to obtain the new system loss and generator power. Then, if the system converges, the program will stop, otherwise the process will be repeated. This procedure is shown in Fig. 2.

TABLE I
COST COEFFICIENTS FOR PIECEWISE QUADRATIC COST FUNCTIONS

U	Min	F1	P1	F2	P2	F3	Max	F	a	b	c
1	50	1	100	2	190	3	200	1	0.000	1.900	.00355
								2	0.000	2.000	.00376
								3	0.000	2.200	.00415
2	20	1	35	2	50	3	80	1	0.000	1.700	.01700
								2	0.000	1.750	.01750
								3	0.000	2.050	.02350
3	10	1	25	2			35	1	0.000	1.000	.06250
								2	0.000	1.200	.0825
4	10	1	20	2			30	1	0.000	3.250	.00834
								2	0.000	3.650	.01234
5	15	1	30	2			50	1	0.000	3.000	.02500
								2	0.000	3.300	.03500
6	12	1	25	2			40	1	0.000	3.000	.02500
								2	0.000	3.300	.03500

Slope adjustment method with load flow was run for fixed and adaptive learning rates at three load demands. The results are shown in Table II for comparison. For 283.4 MW, the generation cost was plotted for fixed learning rate in Fig 3. In Fig 3, the effectiveness of the adaptive Hopfield neural networks is clearly shown.

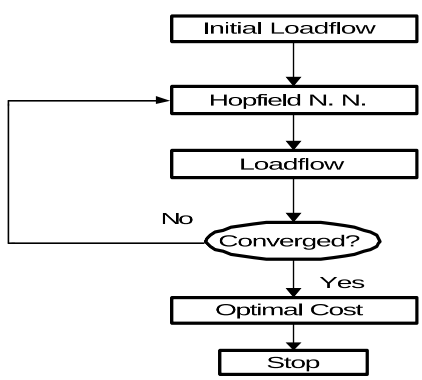

Figure 2 : Flowchart for the optimization with using load flow.

Bias adjustment method with loadflow was run for fixed and adaptive learning rates at three load demands. The results are shown in Table II for comparison. For 283.4 MW, the generation cost was plotted for fixed ,in Fig 3., and adaptive, in Fig 4, learning rates.

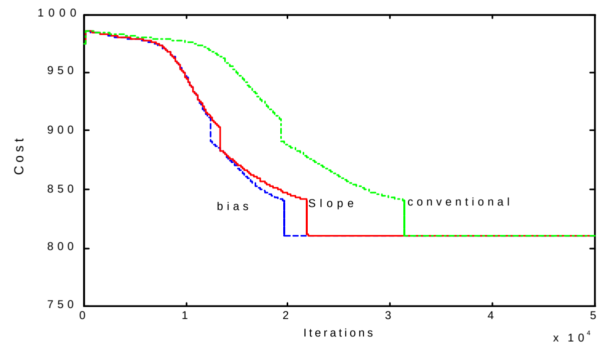

Fig.3: Cost of the using Hopfield network (dashed-dotted line), slope adjustment method with fixed learning rate (solid line) and bias adjustment method with fixed learning rate (dashed line).

TABLE II
RESULTS FOR CONVENTIONAL HOPFIELD NETWORK (A), THE SLOPE ADJUSTMENT METHOD WITH FIXED LEARNING RATE h=1.0 (B) AND THE BIAS ADJUSTMENT METHOD WITH FIXED LEARNING RATE h=1.0 (C)

	220 M			283.4 MW			380 MW		
U	A	B	C	A	B	C	A	B	C
1	134.80	135.11	133.93	180.05	179.12	180.86	200	200	200
2	37.602	37.299	37.352	49.246	49.05	49.19	78.821	78.937	78.935
3	16.495	16.340	16.492	19.853	19.75	19.83	25.837	25.802	25.822
4	10.704	10.774	11.009	15.961	16.31	15.12	29.210	29.886	29.890
5	15.369	15.451	16.027	16.916	17.51	17.18	31.826	31.976	31.758
6	12.083	12.093	12.165	12.953	13.13	12.86	29.995	29.784	29.993
Tot. P.	227.05	227.06	226.97	294.99	294.88	295.45	396.39	396.385	396.39
Cost	589.63	589.73	590.06	810.06	810.08	810.24	1391	1390.89	1390.9
Lost	7.053	7.062	6.974	11.582	11.62	11.645	16.389	16.385	16.398
Iter.	48000	31000	25000	50000	30000	30000	49000	31700	31500
U0	100	100	100	100	100	100	100	100	100
Theta	0	0	50	0	0	50	0	0	50
Time	NA	3m 45s	3m 4s	NA	3m 37s	3m 37s	NA	3m 51s	3m 53s
h	0	1	1	0	1.0	1.0	0	1	1

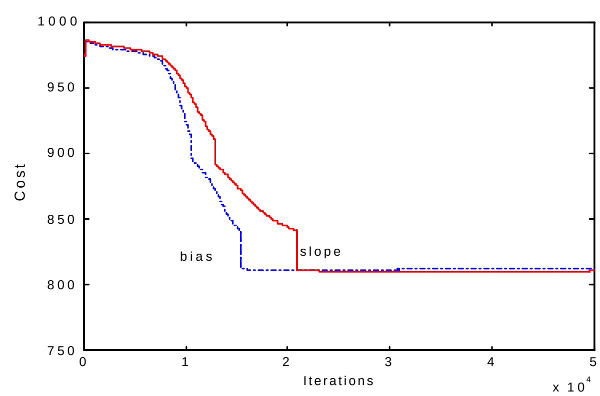

Figure 4: Cost of the using slope adjustment method with load flow for adaptive learning rate (solid line) and slope adjustment method with load flow for adaptive learning rate (dashed-dotted line).

Momentum is applied to the input of each method and gain parameter for slope adjustment method to speed up the convergence of the system. When the momentum is applied to the system, the number of iterations is significantly decreased. In Fig. 5 , the momentum factor of 0.9 and 0.97 is applied to input and gain parameter of slope adjustment methods, and the number of iterations is reduced to about 15 percent of that without momentum of slope adjustment method. The system bus information (after convergence) is shown in Table IV.

As seen in Table III, for 220 MW load, Units 4,5,and 6 have 10.606, 15.357, and 12.074 MW, respectively; which are all very close to lower limits. Also for 380 MW load, Units 1,2, and 4 have 199.999, 79.024, and 29.809 MW, respectively, which are close to 200, 80, and 30 MW upper limits , respectively.

TABLE III
RESULTS FOR ADAPTIVE SLOPE ADJUSTMENT METHOD WITH MOMENTUM 0.9 APPLIED TO INPUT AND GAIN PARAMETER WITH LOAD FLOW AND ADAPTIVE BIAS ADJUSTMENT METHOD WITH MOMENTUM 0.9 APPLIED TO INPUT WITH LOAD FLOW.

	220 MW		283.4 MW		380 MW	
U	Slope	Bias	Slope	Bias	Slope	Bias
1	135.466	133.977	182.984	180.173	199.999	199.986
2	37.178	37.474	48.691	49.188	79.024	79.448
3	16.405	16.518	19.61	19.897	25.76	25.687
4	10.606	10.971	15.385	15.334	29.809	29.435
5	15.357	15.891	15.951	17.443	32.16	32.507
6	12.074	12.15	12.629	12.939	29.628	29.322
Total Power	227.086	226.981	295.252	294.974	396.324	389.385
Cost	589.647	589.965	810.278	810.203	1390.997	391.16
Lost	7.0861	9.981	11.852	11.574	16.324	9.385
Iterations	3200	2700	4000	3000	3000	2500
U0	100	100	100	100	100	100
Theta	0	50	0	50	0	50
Time	25s	21s	30s	23s	25s	20s
n	1.00E-04	1	1.00E-04	1	1.00E-04	1

In Fig. 6, momentum 0.9 is applied to input of bias adjustment method with adaptive learning rate using load flow. Reduction of the iteration number for the bias adjustment method is similar to the case for slope adjustment method.

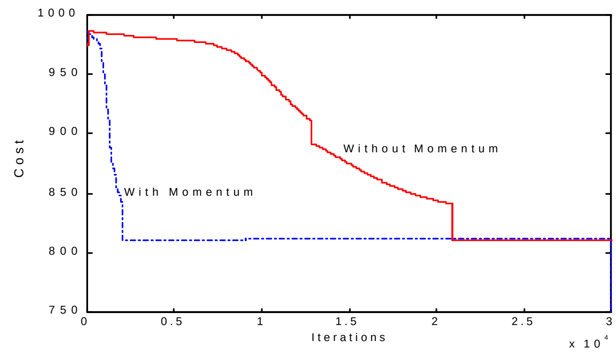

Figure 5: Cost of the using slope adjustment method with load flow for adaptive learning rate without momentum (solid line) and with momentum 0.9 applied at input and momentum 0.97 at gain parameter (dashed-dotted line).

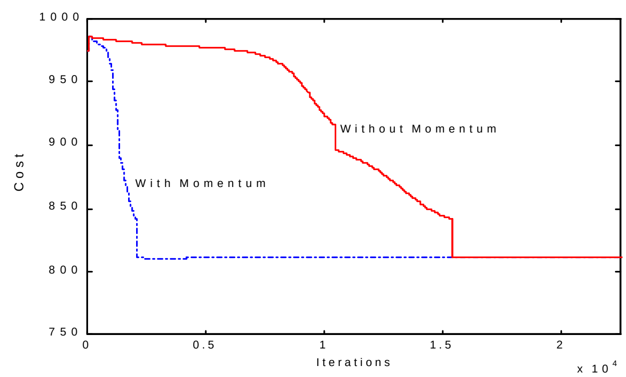

Figure 6: Cost of the using bias adjustment method with load flow for adaptive learning rate without momentum (solid line) and with momentum 0.9 applied at input (dashed-dotted line).

TABLE IV
BUS INFORMATION OF 283.4 MW LOAD WITH ADAPTIVE SLOPE ADJUSTMENT METHOD WITH MOMENTUM

Bus #	V (p.u)	P (MW)	Q(MVAr)	Bus #	V (p.u)	P (MW)	Q(MVAr)
1	1	182.984	-41.067	16	0.934	-3.495	-2.414
2	1	27	24.362	17	0.919	-8.999	-5.899
3	1	-10.381	28.257	18	0.912	-3.199	-0.965
4	1	15.387	28.366	19	0.906	-9.499	-3.482
5	1	-78.241	39.272	20	0.909	-2.198	-0.59
6	1	12.63	30.189	21	0.907	-17.496	-11.741
7	0.984	-22.795	-12.687	22	0.908	0.007	0.705
8	0.989	-2.399	-0.825	23	0.908	-3.197	-1.798
9	0.942	0	2.139	24	0.892	-8.693	-7.841
10	0.922	-5.792	1.698	25	0.893	0.01	0.485
11	0.987	-7.603	-1.413	26	0.87	-3.494	-2.842
12	0.958	-11.176	-2.238	27	0.904	0.004	2.26
13	0.99	0.007	-0.586	28	0.984	0.005	-2.409
14	0.938	-6.193	-0.331	29	0.882	-2.399	-0.952
15	0.927	-8.174	-5.318	30	0.87	-10.599	-1.296

For both figures (Fig 5. and Fig. 6), 283.4 MW load was used. The results of the cases for both methods are shown in Table 4.

6. CONCLUSIONS

In this paper, the slope and bias adjustment are applied to the real power optimization with loadflow. For all three load levels, slope and bias adjustment methods gave very good responses. Especially with momentum applied, the iteration numbers are very low and the computation time is around 20-25 seconds. Around lower and upper limits of generator, the system converged very smoothly.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under grants INT-9605028 and ECS-9705105.

REFERENCES

1. A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, Second Edition, John Wiley & Sons, New York, NY (1996)
2. J. J. Hopfield, 'Neural networks and physical systems with emergent collective computational abilities', *Proceedings of National Academy of Science, USA.*, Vol. 79, pp. 2554-2558, April 1982.
3. J. J. Hopfield, 'Neurons with graded response have collective computational properties like those of two-state neurons', *Proceedings of National Academy of Science, USA.*, Vol. 81, pp. 3088-3092, May 1984.
4. J. A. Freeman and D. M. Skapura, *Neural Networks Algorithms, Applications, and Programming Techniques*, Addison-Wesley Pub. Co., Inc., July 1992.
5. W. D. Tank and J. J. Hopfield, 'Simple neural optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit', *IEEE Transactions on Systems, Mans, and Cybernetics*, Vol. CAS-33, No. 5, pp. 553-541, January/February 1992.
6. H. Park, Y. S. Kim, I. K. Eom, and K. Y. Lee, 'Economic load dispatch for piecewise quadratic cost function using hopfield neural networks', *IEEE Transactions on Power system Apparatus and Systems*, Vol. 8, No. 3, pp. 1030-1038, August 1993.
7. C. E. Lin, and G. L. Viviani, 'Hierachical economic dispatch for piecewise quadratic cost functions', *IEEE Transactions on Power Apparatus and Systems*., Vol. PAS-10, No. 6., June 1984.
8. T. D. King, M. E. El-Hawary, and F. El-Hawary, "Optimal environmental dispatching of electric power system via an improved hopfield neural network model", *IEEE Transactions on Power Systems*, Vol. 1, No. 3, pp. 1559-1565, August 1995.
9. K. Y. Lee, A. Sode-Yome, and J. H. Park, 'Adaptive Hopfield Neural Networks for Economic Load Dispatch', IEEE Trans. on Power Systems, Vol 13 No 2 (May 1998) pp 519-526
10. C. C. Ku and K. Y. Lee, 'Diagonal recurrent neural networks for dynamic systems control," *IEEE, Transactions on Neural Networks'*, Vol. 6, No. 1, pp. 144-156, January 1995.
11. M. A. El-Sharkawi and D. Niebur, *Application of Artificial Neural Networks to Power Systems*, *IEEE Power System Society*, 96 TP 112-0, IEEE, 1996.
12. J. J. Hopfield and W. D. Tank, 'Neural computation of decisions in optimization problems', *Biological Cybernetics*, Vol. 52, pp 141-152, 1985.
13. K. Y. Lee, Y. M. Park and J. L. Ortiz, 'United Approach to Optimal Real and Reactive Power Dispatch', *IEEE Trans. on Power Appratus & Systems*, PAS-104 1985 pp 1147-1153