The Journal of CPRI,
Vol. 11, No. 2, June 2015 pp. 277-286

Multi-layered feed-forward back propagation neural network approach for solving short-term thermal unit commitment

Pavan Kumar V* and Kulkarni P S**

This paper presents an approach for solving the short-term thermal unit commitment (UC) problem using a multi-layered Feed-forward Back propagation Neural Network (FF-BPNN). The main focus of the paper is on finding the schedule of committed thermal units within a short computational timesuch that the total operating cost is minimized. The proposed method is implemented and tested on a 3-unit and 10-unit systems for a scheduling period of 4-hours and 24-hours respectively in MATLAB™ software using the Neural Network toolbox. Comparison of simulation results of the proposed method with the results of previous published methods shows that the proposed FF-BPNN method provides better solution with less computational time.

Keywords: *Artificial Neural Networks (ANN), Dynamic Programming (DP), Multi-layered Feed-forward Back propagation Neural Network (FF-BPNN), Lagrangian Relaxation (LR), Priority List (PL), Unit Commitment (UC)*

NOMENCLATURE

A_i, B_i, C_i : fuel cost coefficients of unit 'i'

CSU_i : cold start-up cost of unit 'i' ($)

D_i, E_i : start-up cost coefficients of unit 'i'

F : total fuel cost($)

$F_i(P_{i,t})$: fuel cost of unit 'i' at hour 't' ($)

HSU_i : hot start-up cost of unit 'i' ($)

i : index of thermal units

$U_{i,t}$: status of units

N : number of units

PD_t : power demand at hour 't' (MW)

$P_{i,t}$: output power of unit 'i' at hour 't' (MW)

$P_{i,min}$: minimum output power limit of unit'i' (MW)

$P_{i,max}$: maximum output power limit of unit'i'(MW)

R_t : maximum reserve capacity at hour 't' (MW)

SU_i : start-up cost of unit 'i'

SD_i : shut-down cost of unit 'i'

SO_i : initial cold start-up cost of unit 'i'($)

t : time period (sec)

T : number of hours

S_{ini} : initial start-up hours of unit 'i'(hrs)

$T_{i,on}$: continuouson-time duration of unit 'i' (hrs)

$T_{i,off}$: continuousoff-time duration of unit 'i' (hrs)

$T_{i,up}$: minimum up time of unit 'i' (hrs)

*PG Student, Department of Electrical Engineering, VNIT, Nagpur, India. E-mail: pavankumar.vummadi.99@gmail.com
**Associate Professor, Department of Electrical Engineering, VNIT, Nagpur, India. E-mail: pskulkarni@eee.vnit.ac.in

$T_{i,down}$: minimum down time of unit 'i' (hrs)

$T_{i,cold}$: cold start hours of unit 'i' (hrs)

M : number of layers in the network

s : index of neurons in a layer of the network

n : number of neurons in a layer of the network

w : weights of neurons in a layer of the network

b : biases of neurons in a layer of the network

e : errors of a layer in thenetwork

a : actual output of the network

t_0 : target output of the network

p : input pattern to the network

tr : transpose of a matrix

k : iteration number

α : learning rate

f : transfer function

$\dot{f}$: derivative of the transfer function

1.0 INTRODUCTION

In any power system the load on the system is dynamic in nature. During the daytime and early evenings it is higher as the industrial loads are high and during the late evenings and early mornings it is lower as most of the population is asleep [1],[8]. Hence the load on the system varies from time to time and this load must be met with the available resources economically. When sufficient generation is kept online throughout the day to meet the peak demand, there is a possibility that some of the units may operate at their minimum generation limits during the off-peak periods. This may become uneconomical due to unnecessary running of units when load demand is less. Now the problem for any power system operator during this time is to determine which units are to be taken offline and for how long.

Unit Commitment (UC) problem in a power system refers to determining the ON/OFF states of the generating units that minimize the total operating cost for a given time horizon. The UC problem solution is a complex optimization problem. Exact solution for the UC problem can be obtained by completely enumerating all the feasible combinations of the generating units which is generally a huge number. Mathematically the UC problem is formulated as non-linear, large scale, mixed integer combinatorial optimization problem with several constraints [3],[4].

The UC schedule is obtained by including many factors such as

i. Operating costs/constraints of the units.

ii. Generation and spinning reserve constraints.

iii. Start-up and shut-down constraints of the plant.

Proper scheduling of the units ensures better usage of the available capacity.

Many methods have been proposed to solve the UC problem. Earlier, classical methods such as Dynamic Programming (DP), Priority List (PL), Lagrangian Relaxation (LR) were used to solve the unit commitment problem. Among these, PL method is the simplest method but the solution obtained is rough and sub-optimal. DP method has been far more a flexible method to solve the UC problem but the computational time required is more for finding the optimal solution due to the "curse of dimensionality". LR method provides a faster solution to solve the short-term UC problem, but it fails in dealing with the solution quality and feasibility problems as the number of units is increased [5].

During the past few years Neural Networks (NN) have been used to find the solution for the UC problem. Artificial Neural Networks (ANNs) have the advantage of parallel processing due to which the computational time is considerably reduced. They also have the advantages of rapid convergence and good solution quality. For these reasons, ANNs are being adapted now-a-days very frequently to solve any problem.

The neural network computing has opened up a new route for the optimization of generation scheduling. With proper and sufficient training, the information regarding the optimal operation of the system can be stored in the network and the output can be obtained in a much shorter time whenever it is required.

In this paper, a multi-layered feed-forward back propagation neural network (FF-BPNN) is considered to determine the variables corresponding to the operating level of generating units in a power system and total production cost. Load demand profile of a utility is given as input to the neural network and the generation capacity of the units is the output from the neural network. This method is tested on a 3- and 10- unit systems using MATLAB™ software and the results obtained from this method are compared with those obtained from the classical conventional methods such as PL, DP, LR and other neural network methods such as DPHNN (DP based fast computation Hopfield Neural Network) in terms of total operating cost and computational time.

2.0 PROBLEM FORMULATION

The UC schedule should be able to minimize the total operational cost to meet the predicted power demand and satisfy all constraints. The objective function and various constraints of the UC problem are explained in the following sections.

2.1 Objective Function

The objective function of UC problem is to minimize the sum of fuel cost,start-up cost and shut-down cost of all individual units subjected to various constraints for a given period of time [6]. It can be mathematically described as

$$Min\ F = \sum_{i=1}^{N}\sum_{t=1}^{T}\left[F_i(P_{i,t})U_{i,t} + SU_i\{U_{i,t}(1-U_{i,t-1})\} + SD_i\{U_{i,t}(1-U_{i,t+1})\}\right] \quad ....(1)$$

The fuel cost is the major component of the total operating cost and is given in quadratic form as

$$F_i(P_{i,t}) = A_i + B_iP_{i,t} + C_iP_{i,t}^2 \quad ....(2)$$

Start-up cost depends upon the off-time of the unit. It varies from a maximum value when the unit is started from a cold state and to a minimum value if the unit has been shut down recently. Hence it can be expressed as an exponential function of the off-time of a generating unit.

$$SU_i = SO_i\left[1 - e^{\left(-\frac{T_{i,off}}{D_i}\right)}\right] + E_i \quad ....(3)$$

Time dependent start-up cost is simplified as follows

$$SU_i = \begin{cases} HSU_i;\ T_{i,down} + T_{i,cold} \geq T_{i,off}^t \\ CSU_i;\ T_{i,down} + T_{i,cold} < T_{i,off}^t \end{cases} \quad ....(4)$$

2.2 Constraints

The UC problem has many constraints depending on the nature of the power system under consideration. Constraints that are considered in this work are divided into the following categories.

2.2.1 System Constraints

2.2.1.1 Power balance equation

The sum of the output powers of all the generating units which are online is equal to the forecasted power demand

$$\sum_{i=1}^{N} P_{i,t}U_{i,t} = PD_t \quad ....(5)$$

2.2.1.2 Spinning Reserve constraints

Spinning reserve is the total available generation capacity from all the units spinning on the system minus the load demand present on the system.

$$\sum_{i=1}^{N} P_{i,max}U_{i,t} \geq PD_t + R_t \quad ....(6)$$

2.2.2 Unit constraints

2.2.2.1 Limits on the generating units

The maximum and minimum output limits on the generating units

$$U_{i,t}P_{i,min} \leq P_{i,t} \leq U_{i,t}P_{i,max} \quad ....(7)$$

2.2.2.2 Minimum up time

A particular unit must be ON for certain no. of hours before it can be shut down

$$T_{i,on} \geq T_{i,up} \quad ....(8)$$

2.2.2.3 Minimum down time

A particular unit must be OFF for certain no. of hours before it can be brought online

$$T_{i,off} \geq T_{i,down} \quad ....(9)$$

Minimum up and minimum down time constraints can be incorporated in the UC problem as

$$U_{i,t} = \begin{cases} 1\ if T_{i,on} < T_{i,up} \\ 0\ if T_{i,off} < T_{i,down} \\ 0\ or\ 1\ otherwise \end{cases} \quad ....(10)$$

2.2.2.4 Must run units

Due to the economic and reliability considerations, these must run units are included in the UC problem.

3.0 ARTIFICIAL NEURAL NETWORK

ANN follows the functionality of the human brain. There are innumerable connections among billions of neurons in the human brain and they are so extensive that the whole network can learn and adapt to various complicated problems. ANN executes several processes in a parallel manner for obtaining outputs instead of carrying out commands and performing calculations in a serial order. An ANN learns from experience and their characteristics make them suitable for solving practical problems quickly and accurately.

Now-a-days ANN is finding applications in several power system operation and control problems[7]. ANN's application for power system optimization problems has become an active research area in recent years. Furthermore its application to UC is posing an interesting engineering application for estimating ANN parameters based on typical load curves and their corresponding UC schedules. Initially the current load curve pattern is compared with the information available in the load database and the most economical schedule for the generating units is selected.

The ANN used in this study is Feed-forward multilayer network with Error Back-Propagation training algorithm (BPA) with supervised learning. Back-propagation indicates that the ANN learns by examples and repetition. In supervised learning, both the inputs and outputs are provided to the network and the network then processes the inputs and compares its resulting/ actual outputs with the desired/target outputs. The errors are then calculated, causing the system to adjust the weights which control the network in an iterative process until the errors reach a minimum tolerance value [2].

The architecture of ANN consists of several layers. The layers are input, hidden and the output layers. The total number of layers depend on the problem requirement. Input layer receives the data and sends the output signals to the hidden layers. Hidden layer gets the inputs from input layers and sends its output to the output layer. The output layer sends its output to the outside upon receiving the signal from the hidden layers [9]. A transfer function is used to represent the interior mechanism of a neuron. We next discuss the Back propagation algorithm in solving the Unit Commitment.

3.1 Back propagation algorithm

3.1.1 Overview

In order to train single layer networks, we can correlate weights and biases with error and devise

a new strategy to change them so as to reduce error. But for training multilayer networks, this strategy can be used only for the output layer (since output layer weights and biases are directly involved in the final output and hence error). In such a case, correlation between weights and biases of hidden and output layers is not possible to find the error. To rectify this problem a new algorithm has been devised by Werbos and Rumelhart named as Back Propagation Algorithm (BPA).

In Back Propagation neural network (BPNN), the outputs of each node in a layer are connected to inputs of all the nodes in the subsequent layer. Data flows through the network in only one direction i.e., from input to output and the errors are propagated backward through the network from last layer to the first.

3.1.2 PA general algorithm

Step 1 : Decide the no. of hidden layers, no. of neurons in the input, hidden and the output layers.

Step 2 : Obtain the input pattern-target pairs.

Step 3 : Initialize weights and biases to small random values and set error threshold for stopping iterations. Set maximum no. of iterations in case if goal is not reached.

Step 4 : Apply first input pattern at the input layer and propagate it forward through the network to get the output at each output neuron.

a^0 = (neurons in first layer receive external inputs) ....(11)

$$a^{m+1} = f^{m+1}(w^{m+1}a^m + b^{m+1}) \quad ....(12)$$

$$a = a^M \quad ....(13)$$

Outputs of neurons in last layer are the network outputs.

Step 5 : Back propagate the error through the network

$$e^M = -2\dot{f}^M(n^M)(t_o - a) \quad ....(14)$$

$$e^m = \dot{f}^m(n^m)(w^{m+1})^{tr}e^{m+1}$$

$$m = M-1, \ldots\ldots, 2, 1 \quad ....(15)$$

Where

$$\dot{f}^m(n^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & 0 \\ 0 & \dot{f}^m(n_2^m) & 0 \\ 0 & 0 & \dot{f}^m(n_s^m) \end{bmatrix}$$

Step 6 : Update weights and biases.

$$w^m(k+1) = w^m(k) - \alpha e^m(a^m)^{tr} \quad ....(16)$$

$$b^m(k+1) = b^m(k) - \alpha e^m \quad ....(17)$$

Step 7 : Repeat from step 4 to 6 for remaining input pattern-target pairs.

Step 8 : Continue to iterate until the difference between the network response and the target function reach an acceptable level.

3.2 Implementation of FF-BPNN using Neural Network Toolbox for solving UC problem

The solution of UC problem using FF-BPNNcan be implemented in MATLAB™ software through Neural Network Toolbox to reduce the computing time. For training the neural network, total load supplied is given as input and generation schedules of the thermal units are given as outputs to the network. The ON-OFF status of the generating units is obtained from the output of the testing phase of the neural network. This output can be used to calculate the running cost of each unit at the end of each hour. The running cost and start-up costs at the end of each hour for the finally obtained status of a unit are summed together to give the operating cost of that particular unit at each hour. The overall operating cost is calculated by summing up the operating costs of all the units.

Procedure for solving UC problem using MATLAB™'s Neural Network Toolbox is as follows.

Step 1 : The input-target pattern pairs are to be kept ready using the "New variable" tab that is available in the MATLAB™ toolstrip.

Step 2 : Type "nntool" in command window, neural network/Data manager (nntool) editor appears.

Step 3 : Import the input data and target data which is to be used for training the neural network using the "Import" tab available in the neural network editor.

Step 4 : Create a new network by selecting the "New" tab in the neural network editor. A pop-up screen appears showing different types of networks, training functions, no. of layers, no. of neurons and transfer functions. Select the required ones and import data as per the requirement of the problem.

Step 5 : Now, open the created network using "Open" tab. Upon selecting, a new window appears with a pictorial view of the created network. This network can be trained by adjusting the training parameters.

Step 6 : The above Step 4 and Step 5 should be repeated several times by selecting different no. of layers and neurons until the error reduce to an acceptable level.

Step 7 : Now, the trained neural network has to be simulated to obtain the required outputs.

4.0 SIMULATION RESULTS

The proposed FF-BPNN method for UC has been implemented in MATLAB™ R2012b (8.0 version) and executed on Intel core i_3-2nd generation (2.2 GHz) PC with 4 GB RAM. This method has been tested on a 3- and 10-unit systems to solve UC problem. Results obtained are compared with those obtained from PL, DP, LR and other neural networks such as DPHNN in terms of total operating cost and computational time.

The simulation results are shown below.

4.1 3-unit system

In this case, a 3-unit system is considered. The fuel cost data and power demands of 3-unit system are obtained from [1], [10] and are presented in Table 1 and Table 2 respectively.

TABLE 1
FUEL COST DATA OF 3-UNIT SYSTEM

U	A_i ($/h)	B_i ($/MWh)	C_i ($/MW²h)	$P_{i,min}$ (MW)	$P_{i,max}$ (MW)
1	500	10	0.0020	100	600
2	300	08	0.0025	100	400
3	100	06	0.0050	50	200

TABLE 2
POWER DEMAND FOR 3-UNIT SYSTEM OVER 4-HOURS PERIOD

Hour	Power demand (MW)
1	170
2	520
3	1100
4	330

Here for the 3 –unit system, two hidden layers with 28 neurons in each hidden layer are considered in the design of the back propagation neural network. The input to the neural network contains the total load supplied and the three thermal generator outputs are considered as outputs of neural network. Thus there is one input node and three output nodes. 'TRAINLM' training function and 'LEARNGDM' learning functions along with tan-sigmoid transfer function are used to train the neural network.

The following training parameters (optional) are considered during training:

1. Maximum number of epochs to train=1000
2. Epochs between updating display=25
3. Mean-squared error goal=0.03
4. Learning rate = 0.01

Learning stops when either the maximum number of epochs has occurred or the network's mean-squared error has dropped below the error goal. Using these parameters, training of the designed network requires 856 epochs.

Figure 1 shows MSE vs Epoch for the trained FF-BPNN created for 3-unit system.

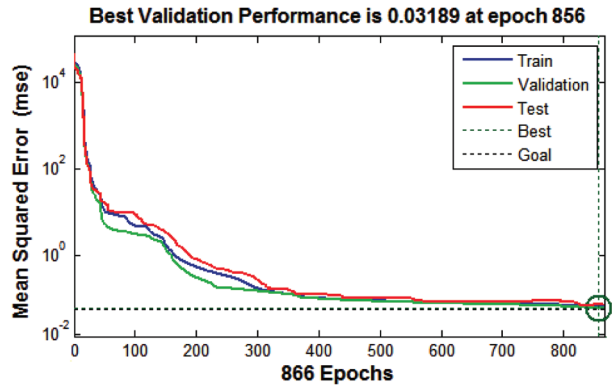

FIG. 1 MSE VS EPOCH FOR TRAINED FF-BPNN CREATED FOR THE 3 – UNIT SYSTEM

The UC schedule of 3-unit system for proposed FF-BPNN method is displayed in Table 3.

4.2 10-unit system

In this case, 10-unit system is considered. The power demands and the fuel cost data of 10-unit system are obtained from [10], [11] and are presented in Table 4 and Table 5 respectively.

TABLE 3
UC SCHEDULE OF 3-UNIT SYSTEM FOR PROPOSED FF-BPNN METHOD

Hour	Units			Fuel cost ($)
	1	2	3	
1	0	0	1	1265
2	0	1	1	4615
3	1	1	1	11400
4	0	1	1	2881
Total Fuel Cost ($)				20161

** 1 – UNIT COMMITTED
** 0 – UNIT NOT COMMITTED

TABLE 4
POWER DEMANDS FOR 10-UNIT SYSTEM OVER 24-HOURS PERIOD

Hour	PD (MW)	Hour	PD (MW)	Hour	PD (MW)
1	700	9	1300	17	1000
2	750	10	1400	18	1100
3	850	11	1450	19	1200
4	950	12	1500	20	1400
5	1000	13	1400	21	1300
6	1100	14	1300	22	1100
7	1150	15	1200	23	900
8	1200	16	1050	24	800

TABLE 5
FUEL COST DATA OF 10-UNIT SYSTEM

U	A_i ($/h)	B_i ($/MWh)	C_i ($/MW²h)	$P_{i,min}$ (MW)	$P_{i,max}$ (MW)	CSU_i ($/h)	HSU_i ($/h)	$T_{i,up}$ (h)	$T_{i,down}$ (h)	$T_{i,cold}$ (h)	S_{ini} (h)
1	1000	16.19	0.00048	150	455	9000	4500	8	8	5	8
2	970	17.26	0.00031	150	455	10000	5000	8	8	5	8
3	700	16.60	0.00200	20	130	1100	550	5	5	4	-5
4	680	16.50	0.00211	20	130	1120	560	5	5	4	-5
5	450	19.70	0.00398	25	162	1800	900	6	6	4	-6
6	370	22.26	0.00712	20	80	340	170	3	3	2	-3
7	480	27.74	0.00079	25	85	520	260	3	3	2	-3
8	660	25.92	0.00413	10	55	60	30	1	1	0	-1
9	665	27.27	0.00222	10	55	60	30	1	1	0	-1
10	670	27.79	0.00173	10	55	60	30	1	1	0	1

TABLE 6

UC SCHEDULE OF 10-UNIT SYSTEM FOR PROPOSED FF-BPNN METHOD

Hour	Units										Fuel cost ($)	Start-up Cost ($)
	1	2	3	4	5	6	7	8	9	10		
1	1	1	0	0	0	0	0	0	0	0	13668	0
2	1	1	0	0	0	0	0	0	0	0	14552	0
3	1	1	0	0	1	0	0	0	0	0	16819	900
4	1	1	0	0	1	0	0	0	0	0	18598	0
5	1	1	0	1	1	0	0	0	0	0	20020	560
6	1	1	1	1	1	0	0	0	0	0	22397	1100
7	1	1	1	1	1	0	0	0	0	0	23254	0
8	1	1	1	1	1	0	0	0	0	0	24145	0
9	1	1	1	1	1	1	1	0	0	0	27261	860
10	1	1	1	1	1	1	1	1	0	0	30057	60
11	1	1	1	1	1	1	1	1	1	0	31905	60
12	1	1	1	1	1	1	1	1	1	1	33890	60
13	1	1	1	1	1	1	1	1	0	0	30057	0
14	1	1	1	1	1	1	1	0	0	0	27264	0
15	1	1	1	1	1	0	0	0	0	0	24145	0
16	1	1	1	1	1	0	0	0	0	0	21513	0
17	1	1	1	1	1	0	0	0	0	0	20641	0
18	1	1	1	1	1	0	0	0	0	0	22397	0
19	1	1	1	1	1	0	0	0	0	0	24145	0
20	1	1	1	1	1	1	1	1	0	0	30057	490
21	1	1	1	1	1	1	1	0	0	0	27264	0
22	1	1	0	0	1	1	1	0	0	0	22736	0
23	1	1	0	0	1	0	0	0	0	0	17679	0
24	1	1	0	0	0	0	0	0	0	0	15423	0
Total ($)											559887	4090
Total Operating Cost ($)											563977	

** 1 – UNIT COMMITTED
** 0 – UNIT NOT COMMITTED

TABLE 7

COMPARISON OF RESULTS

	3-unit		10-unit	
Method	**Production Cost ($)**	**Computational time (sec)**	**Production Cost ($)**	**Computational time (sec)**
PL	20162.75	63	565664	197
LR [10]	20170	75	565825	235
DP	20168.16	84	565971	260
DPHNN [5]	-	-	588750	5.3
Proposed method (FF-BPNN)	20161	0.46	563977	1.68

Here for the 10 – unit system, two hidden layers with 73 neurons in each hidden layer are considered in the design of the back propagation neural network. The inputs and outputs of this neural network are same as that of 3-unit system except the total number output nodes. There will be one input node and ten output nodes because there are ten thermal units. 'TRAINSCG' training and 'LEARNGDM' learning functions with tan-sigmoid transfer function are used to train the neural network.

The following training parameters (optional) are considered during training:

1. Maximum number of epochs to train=1000
2. Epochs between updating display = 25
3. Mean-squared error goal = 0.025
4. Learning rate = 0.01

Using these training parameters, the training of this designed network requires 935 epochs.

Figure 2 shows MSE vs Epoch for the trained FF-BPNN created for the 10-unit system.

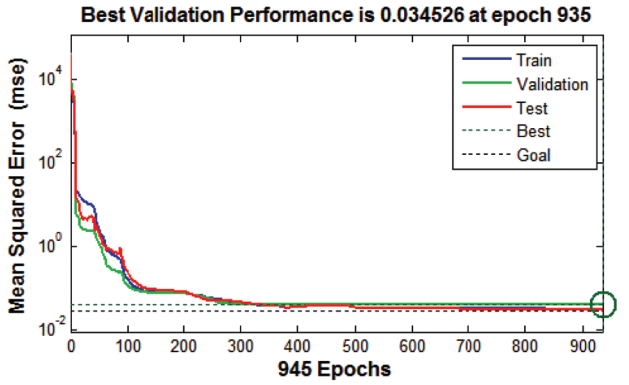

FIG. 2 MSE vs EPOCH FOR TRAINED FF-BPNN CREATED FOR THE 10 – UNIT SYSTEM

The UC schedule of 10-unit system for proposed FF-BPNN method is displayed in Table 6.

From Table 7 it is clear that the proposed approach provides a better solution to reduce the operating cost and solves the unit commitment problem effectively with less computational time compared to other available methods.

5.0 CONCLUSIONS

Feed-Forward back propagation neural network (FF-BPNN) for solving Unit Commitment (UC) problem is presented in this paper. This method solves the UC problem without any huge computational burden and gives us a lower total operating cost within less computational time. However, it needs more samples of data to train the neural network for obtaining the results with least error. This method can be implemented further to a larger system having more number of generating units as well as different types of generating units in order to solve the UC problem effectively and efficiently. The main drawback of the proposed method is that neural network program is system dependent, which means that an ANN with its trained network for one system is not readily transferrable to another system with a different configuration. The training pattern developed for one system may not be suitable for another system.

Future work aims ateliminating its drawback and reducing the training time for the proposed network.

ACKNOWLEDGEMENT

The authors are thankful to the authorities of Visvesvaraya National Institute of Technology (VNIT), Nagpur for the facilities provided to carry out this research work.

REFERENCES

[1] A J Wood and B F Wollenberg, "Power Generation, Operation and Control", John Wiley & Sons, New York, pp. 138-155, 1996.

[2] K Rajasekaran and G A Vijayalakshmi Pai, "Neural Networks, Fuzzy logic and Genetic algorithms: Synthesis and applications", Prentice-Hall of India Private Limited, New Delhi, pp. 34, 2006.

[3] C C Asir Rajan and M R Mohan, "An evolutionary programming-based tabu search method for solving unit commitment problem", IEEE Trans. Power Syst., Vol. 19, No. 1, pp. 577-585, 2004.

[4] C Y Chung, H Yu and K P Wong, "An advanced quantum-inspired evolutionary algorithm for unit commitment", IEEE Trans. Power Syst., Vol. 26, No. 2, pp. 847-854, 2011.

[5] S Senthil Kumar and V Palanisamy, "A dynamic programming based fast computation Hopfield neural network for unit commitment and economic dispatch", Electrical Power System Research, Elsevier, Vol. 77, pp. 917-925, 2007.

[6] T Saksornchai, W J Lee, K Methaprayoon, J R Liao and R J Ross, "Improve the unit commitment scheduling by using the neural-network-based short-term load forecasting", IEEE Trans. Ind. Appl., Vol. 41, No. 1, pp. 169-179, 2005.

[7] P S Kulkarni, A G Kothari and D P Kothari, "Combined economic and emission dispatch using improved back-propagation neural network", Electric Machines and Power Systems, Taylor & Francis, Vol. 28, pp. 31-44, 2000.

[8] B Saravanan, S Das, S Sikri and D P Kothari, "A solution to the unit commitment problem – A review", Front. Energy, Springer, Vol. 7, No. 2, pp. 223-236, 2013.

[9] Z Ouyang, and S M Shahidehpour, "A multi-stage intelligent system for unit commitment", IEEE Trans. Power Syst., Vol. 7, No. 2, pp. 639-646, 1992.

[10] C P Cheng, C W Liu and C C Liu, "Unit commitment by lagrangian relaxation and genetic algorithms", IEEE Trans. Power Syst., Vol. 15, No. 2, pp. 707-714, 2000.

[11] M Carrion and J M Arroyo, "A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem", IEEE Trans. Power Syst., Vol. 21, No. 3, pp. 1371-1378, 2006.