ITM Journal of
Sustainable Engineering and Innovation

A Computational Intelligence Framework for Optimal Wheeling Pricing in Competitive Electricity Markets

Abhishek Saxena[1]

Department of Electrical Engineering, SoET, ITM University, Gwalior, Madhya Pradesh, India

Submitted on: 02/10/2026, Accepted and Published on:15/10/2026

Research /Article

ABSTRACT

Fast and accurate wheeling pricing has emerged as an important issue in the recent competitive power market. In this paper Bialek tracing has been employed to track the contribution of each generator and load in the wheeling flow of each transmission line. For computing wheeling prices MVA-mile method among various embedded cost based pricing techniques has been used in which optimized flows have determined by employing metaphor-less Rao-3algorithm based AC-OPF. Embedded cost based pricing techniques are more reliable for wheeling pricing because they are based on actual flows and recover fixed cost of wheeling facilities. In spot power market due to continuously varying load conditions, the flows over transmission lines have also varied drastically. These varied flows are determined by running OPF program for each and every load condition. Consequently the computation of wheeling prices in real time for these varied flows is become quite a time taking process. Instead employing ANN approach for estimating wheeling prices for any loading scenario, instantly and accurately, is quite an attractive efficient alternative. Various radial basis function ANNs (RBFNNs) have been developed for each cluster in which wheeling prices lie in a closed range. All these neural networks trained and tested under parallel computing environment to acquire speed gain. The proposed neural network and parallel computing based intelligent approach for estimating wheeling prices in real time competitive power market has been demonstrated and examined on IEEE 30-bus system and Indian utility 146-bus system. The results are found well within acceptable accuracy limits.

Keywords: Bialek Tracing, Radial Basis Function Neural Network (RBFNN), MVA- mile method, Parallel Computing, Wheeling Pricing, metaphor-Less Rao-3 Algorithm.

1. INTRODUCTION

The recent trend of deregulation in electric supply industry has renewed interest in wheeling pricing. Because in this new electricity commodity business, all market participants (suppliers, buyers and transmission system operators) are keen to know the transmission cost as well as wheeling prices in order to accommodate their existing and future transactions. This also helps in taking right economic decisions while performing and promoting various wheeling services. Assessment of wheeling prices also ensures efficient use of wheeling facilities by various users [1]. Wheeling pricing has become an important issue in terms of use, operation and expansion of the existing wheeling facilities in the current competitive power market environment. Wheeling pricing involves the cost of wheeling facility incurred by various transactions made by generators and loads [2].

Fixed transmission cost includes initial investment in installing transmission lines, operation and maintenance expenses. Besides this wheeling facility cost involves increment in fuel cost incurred due to repeated generation re-scheduling, cost incurred in accommodating different bilateral and multilateral transactions due to various transmission constraints and any other expenses involved to avail specific wheeling facility. In order to recover these charges from wheeling facilities, embedded cost based methods are very effective [3] and hence used in the present work. In the present work MVA-mile method of wheeling pricing has been adopted, which is an extension of original MW-mile method in which only real power flows were to be considered. In MVA-mile method, both real and reactive power flows are to be considered for pricing which seems to be more realistic [4, 5].

*Corresponding Author: Dr. Abhishek Saxena
Email: abhisheksaxena@itmuniversity.ac.in

Cite as ITM J. of Sust. Engg. and Ino. 2026, 01(1)
DOI:

To obtain optimized apparent (real and reactive) power flows, optimal power flow solutions can be obtained either by conventional optimization methods or by evolutionary computing based optimization methods. Since conventional methods are outmoded due to their sluggish response, unwanted complex structure and high dependency on data availability, the evolutionary artificial intelligence based optimization computing techniques are more efficient and popular now a days [6, 7, 8 and 9]. In the present work metaphor-less Rao-3 algorithm based AC-OPF [10] has been implemented to determine optimal power flow solutions over all the transmission lines. Bialek tracing algorithm [11] for both up-stream and down-stream flows, has been employed to establish contribution of each generator and load in all the flows. This enables the flow based wheeling pricing effectively for different users (generators and loads) of the wheeling system.

The assessment of wheeling pricing traditionally for each loading scenario is not feasible for real time power market. Moreover, any noisy or partially corrupted data greatly affects the accuracy. In addition, for every change in load/ generation/ topology, the OPF algorithm has to be run repeatedly each time leading to a time consuming process for wheeling pricing. In this situation ANN based intelligent technique is found most functional as it possesses tolerance for imprecision and uncertainty and having excellent capability of mapping complex and highly nonlinear input output patterns [12 and 13]. Once an appropriate ANN model has been developed, accurate and instant wheeling pricing can be obtained for any unseen load pattern, even for missing or partially corrupted data. In the present work a novel radial basis function neural network (RBFNN) has been approached for estimating wheeling prices for each generator and load. Apparantly with sufficient RBF neurons, the RBFNN can acquire very minimal error during training. During parameter optimization, an appropriate number of RBF neurons in hidden layer can be obtained and they are at par with the input training patterns [14 and 15]. RBFNN is found specifically suitable for function approximation and time series prediction [16].

For large power markets containing a large number of transmission lines it has been found that wheeling prices obtained through MVA-mile wheeling pricing method, lie in a wide range. A single ANN developed for estimating these wheeling prices has not been found satisfactory. To obtain accuracy in the estimation of these wheeling prices a separate RBFNN had to be developed for each cluster in which wheeling prices lie in a closed range. In this work K-means clustering has been employed to obtain different clusters of wheeling prices [17]. More distinctly all these RBFNNs have been developed (trained and tested) under parallel computing environment in which a simultaneous execution of traing and testing has been obtained over multiple processors via shared memory [18 and 19]. This further enhances the speed of the wheeling price prediction.

Though ANN has been found its application in the prediction of spot pricing like locational marginal prices, nodal congestion prices etc. [13and 14]. However it has not been found yet in the literature for estimating wheeling prices.

The present approach of RBFNN based wheeling price estimation under parallel computing environment has been demonstrated and examined on IEEE 30-bus system and Indian utility 146-bus system.

2. METAPHOR LESS RAO-3 OPTIMIZATION ALGORITHM

In the recent years a number of nature inspired, population based meta-heuristic algorithms are immersed based on *metaphor* of some natural phenomena or behavior of animals, fishes, insects etc. In this situation developing simple and effective optimization technique based on metaphor-less algorithm is the superior alternative. In the present work Metaphor-less Rao-3 algorithm for optimization has been approached [20]. In the present paper real and reactive optimal power flows have been determined by employing Metaphor-less Rao-3 algorithm [21]. The proposed approach uses self-adaptive population based learning which avoids local minima by replacing *worst* solution candidate by *best* solution candidate and by comparing updated solution by any randomly picked solution in each iteration.

Let $f(x)$ is the objective function to be minimized or maximized. At any iteration i, suppose that there are 'p' number of design variables and 'q' number of candidate solutions (population size, k = 1, 2, ……. q). Let the best candidate *best* obtains the best value of $f(x)$ i.e. $f(x)_{best}$ in the entire population and the worst candidate *worst* obtains the worst value of $f(x)$ i.e. $f(x)_{worst}$ in the entire population.

If $X_{j,k,i}$ is the value of j^{th} variable for the k^{th} candidate during i^{th} iteration, then this value is modified/ updated as per following equations.

$$X_{j,k,i}^{new} = X_{j,k,i} + rand(1)_{j,i} * (best_{j,i} - worst_{j,i}) \quad (1)$$

where, $best_{j,i}$ is the value of variable j for best candidate and $worst_{j,i}$ is the value of variable j for worst candidate during i^{th} iteration. $X_{j,k,i}^{new}$ is the updated value of $X_{j,k,i}$. $rand(1)_{j,i}$ is the random number for variable j during i^{th} iteration in the range [0,1].

$$X_{j,k,i}^{new} = X_{j,k,i} + rand(1)_{j,i} * (best_{j,i} - worst_{j,i}) + rand(2)_{j,i} * \left(\left| X_{j,k,i} or X_{j,n,i} \right| - \left| X_{j,n,i} or X_{j,k,i} \right| \right) \quad (2)$$

$$X_{j,k,i}^{new} = X_{j,k,i} + rand(1)_{j,i} * (best_{j,i} - \left| worst_{j,i} \right|) + rand(2)_{j,i} * \left(\left| X_{j,k,i} or X_{j,n,i} \right| - (X_{j,n,i} or X_{j,k,i}) \right) \quad (3)$$

In (2), the term $X_{j,k,i}$ or $X_{j,n,i}$ indicates the comparison of candidate solution with any randomly picked candidate solution and the information is exchanged based on their fitness value. If the fitness value of k^{th} candidate solution is better than the fitness value of n^{th} candidate, then the term $X_{j,k,i}$ or $X_{j,n,i}$ will become $X_{j,k,i}$. On the other hand, if the fitness value of n^{th} candidate is better than the k^{th} candidate, then the term $X_{j,k,i}$ or $X_{j,n,i}$ will become $X_{j,n,i}$.

Similarly if the fitness value of n^{th} candidate is better than the k^{th} candidate, then the term $X_{j,n,i}$ or $X_{j,k,i}$ becomes $X_{j,k,i}$. The complete steps are given as follows:

Step 1. Initialize population size, design variables and assign maximum number of iterations as terminating criteria.

Step 2. Evaluate the fitness value to determine finest and inferior solutions i.e. *best* and *worst*.

Step 3. Modify or update the solution as per (1), (2) and (3).

Step 4. Compare the updated fitness value with the previous one. Accept the better solution.

Step 5. Check the termination criteria i.e. maximum number of iterations. If the conditions are met, and then terminate the process and solution corresponds to best fitness value will be the optimized solution. Otherwise go to step 2.

3. Radial Basis Function Based Neural Networks (RBFNN)

The architecture of proposed RBF neural network is shown in Fig. 1. It consists of three layers namely input, hidden and output layer. Hidden layer is made up of RBF neurons. As can be seen from the Fig. 1 each layer is fully connected with the previous layer through neurons. To generate an output, neurons process the inputs through an activation function. In RBFNN, the activation function is Gaussian density function (Ω), which is defined by Centre position and width parameter [22].

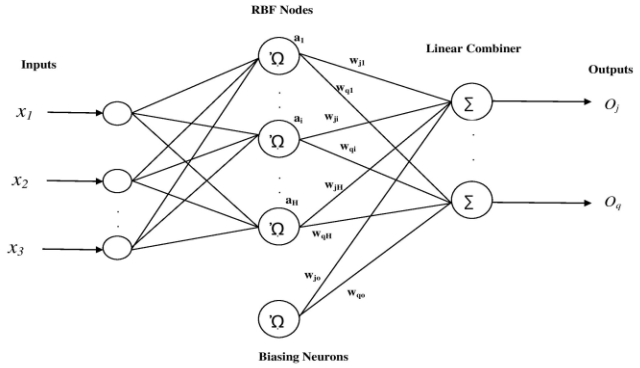


Fig.1: RBFNN Configuration

The RBFNN can be developed in a fraction of time as compared to other design approaches in multilayer feed forward neural networks. For a chosen error goal, different values of RBF spread widths (σ) have been investigated by having several trials to find the acceptable results. An appropriate number of RBF neurons have been determined iteratively during training process untill a specified error goal is achieved [16].

The output of i^{th} RBF unit in the hidden layer is given by:

$$a_i = \exp\left[-\sum_{n=1}^{N}(x_n - \bar{x}_{in})^2 / \sigma_{in}^2\right] \quad (4)$$

Where, N is the number of variables in each input pattern. x_j is the j^{th} variable of input vector for each input pattern. x_{in} is the centre of i^{th} RBF unit for n^{th} input vector. σ_{in} is the width of i^{th} RBF unit for n^{th} input vector.

The output value of q^{th} output node i.e. O_q is equal to the summation of the weighted output of the hidden RBF units and the biasing term of the output node and is given by:

$$O_q = \sum_{i=1}^{H} w_{qi} a_i + w_{qo} \quad (5)$$

Where, H is the number of RBF hidden units, w_{qi} is the weight between i^{th} RBF unit and the q^{th} output node and w_{qo} is the biasing term between biasing neuron and q^{th} output. w_{qo} represents the strength of biasing neuron. The biasing neuron is connected to output nodes only.

In the present work wheeling pricing estimation has been performed through developing an appropriate RBFNN with enough RBF neurons using MATLAB toolbox [23]. In the proposed RBFNN, all non-zero real and reactive power loads are taken as inputs, while wheeling prices for generators and loads are taken as outputs.

3. Optimal Power Flow (AC-OPF)

For obtaining optimized real and reactive power flows Metaphor-less Rao-3 algorithm based AC-OPF algorithms has been implemented. Mathematical formulation for AC-OPF [24] is given as follows:

Objective function: $Min\ f(x,u)$ (6)

subject to ,

$$g(x,u) = 0 \ and \ h(x,u) \le 0 \quad (7)$$

where, g and h are equality and inequality constraints. x and u denotes dependent and control variables. In the present work dependent variables are real power on slack bus (P_{sl}), voltage on load bus (V_L), reactive power output of generators (Q_G) and apparent flows over transmission lines (S_l). Control variables are real power output of generators (P_G) except slack bus (1 number bus is slack bus), voltage on generator bus (V_G), transformer tap setting (T) and shunt VAR compensation (VAR). In this work total number of transmission lines are taken as N, total number of load buses are taken as N_L, total number of generators buses are taken as N_G, total number of regulating transformers are RT and total number of shunt compensators are SC.

$$x^T = [P_{sl}, V_{L_1} \ldots V_{L_{N_L}}, Q_{G_1} \ldots\ldots Q_{G_{N_G}}, S_{l_1} \ldots\ldots S_{l_N}] \quad (8)$$

$$u^T = [P_{G_2} \ldots\ldots P_{G_{N_G}}, V_{G_1} \ldots\ldots V_{G_{N_G}}, T_1 \ldots\ldots T_{RT}, VAR_1 \ldots\ldots VAR_{SC}] \quad (9)$$

Equality constraints belong to standard power flow equation which is given by (5).

$$\sum P_G = P_L + P_{loss} \quad (10)$$

Inequality constraints are given below

$$V_{G_i}^{\min} \le V_{G_i} \le V_{G_i}^{\max} \quad i = 1,2,\ldots\ldots N_G \quad (11)$$

$$P_{G_i}^{\min} \le P_{G_i} \le P_{G_i}^{\max} \quad i = 1,2,....N_G \tag{12}$$

$$Q_{G_i}^{\min} \le Q_{G_i} \le Q_{G_i}^{\max} \quad i = 1,2,.....N_G \tag{13}$$

Transformer constraints:

$$T_i^{\min} \le T_i \le T_i^{\max} \quad i = 1,.....RT \tag{14}$$

Shunt VAR constraints

$$Q_i^{\min} \le Q_i \le Q_i^{\max} \quad i = 1,.....SC \tag{15}$$

$$V_{Li}^{\min} \le V_{Li} \le V_{Li}^{\max} \quad i = 1,.....N_L \tag{16}$$

$$S_l \le S_l^{\max} \quad ,l = 1,.....N \tag{17}$$

Since controlled variables are self-constrained hence, the objective function can be augmented as follows:

$$f_{avg} = f + \lambda_p \left(P_{sl} - P_{sl}^{\lim}\right)^2 + \lambda_v \sum_{c=1}^{N_L} \left(V_{LC} - V_{LC}^{\lim}\right)^2 + \lambda_Q \sum_{c=1}^{N_G} \left(Q_{GC} - Q_{GC}^{\lim}\right)^2 + \lambda_s \sum_{c=1}^{N_L} \left(S_{N_L} - S_{N_L}^{\max}\right)^2 \tag{18}$$

where $\lambda_p, \lambda_v, \lambda_Q, \lambda_s$ are penalty factors and 'x' is the limit value of dependent variable 'x'. So

(19)

$$x^{\lim} = \begin{cases} x^{\max} & , x \ge x^{\max} \\ x^{\min} & , x < x^{\min} \end{cases}$$

4.1 MVA-mile Method for Wheeling Pricing

In the present work wheeling prices based on real and reactive flows have been computed using MVA-mile method, thus presents realistic economic signals among market participants. In this paper contribution of each generator and load in all the flows over transmission lines is computed using Bialek flow tracing [11]. The wheeling prices for generators and loads are given as follows:

$$WP_G = TTC_G \frac{\sum_{k \in N} C_k L_k S_{G,k}}{\sum_{G \in N_G} \sum_{k \in N} C_k L_k S_{G,k}} \tag{20}$$

Where WP_G is the wheeling price computed for G^{th} generator, TTC_G is the total transmission cost share for generators, C_k is the cost of k^{th} wheeling facility per unit length and per unit flow of line, L_k is the length of the k^{th} transmission line and $S_{G,k}$ is the apparent flow over k^{th} transmission line due to G^{th} generator in MVA. N and N_G are total number of transmission lines and generators.

$$WP_L = TTC_L \frac{\sum_{k \in N} C_k L_k S_{L,k}}{\sum_{L \in N_L} \sum_{k \in N} C_k L_k S_{L,k}} \tag{21}$$

Where WP_L is the wheeling price computed for L^{th} load, TTC_L is the total transmission cost share for loads, and $S_{L,k}$ is the apparent flow over k^{th} transmission line due to L^{th} load in MVA. N_L is total number of loads in the given power system.

4.2 Formation of clusters

For a large power market containing a large number of transmission lines, generators and loads, the wheeling prices allotted for generators and loads are found heterogeneous i.e. lie in a wide range. Therefore a single neural network developing for estimation of these wheeling prices does not perform satisfactorily. On the other hand, if a separate neural network is developed for each generator and each load, there will be a large number of neural networks and a long training time will be required. Hence to overcome this problem clustering technique has been implemented to group all the wheeling prices which lie in a closed range in a cluster. The basic purpose of clustering is to group different system variables into certain number of clusters, such that the variables in a cluster have similar characteristics. In the present work k-means clustering [17] has been employed to make clusters such that wheeling prices which are in a closed range are grouped in a cluster.

5 PARALLEL COMPUTING

In parallel computing, a computationally intensive task has been divided into different sub-tasks and then executed simultaneously by exploiting the processing capacity of multi-processor computer. Thus the execution of time costly task is speed-up by assigning these tasks to different processor of same computer. In the present paper, parallel computing [18 and 19] has been employed to develop a number of RBFNNs simultaneously for estimating wheeling prices allocated to different generators and loads. In parallel computing the assignment of tasks to different processors must be wisely done such that each processor may finish the task in almost same time as far as possible. In this way a significant speed gain can be achieved over sequential computing.

6. RESEARCH METHODOLOGY

A general block diagram shown in Fig. 1 illustrates the research methodology adopted for wheeling price estimation using RBFNN under parallel computing environment. A large number of patterns are generated in wide range of load variation. Metaphor less Rao-3 algorithm based ACOPF has been executed to obtain real and reactive power flows. Then Bialek tracing technique has been implemented to determine the contribution of each generator and

load on these flows. After that, wheeling prices for each generator and load has been computed using MVA-mile method. An unsupervised learning algorithm is used to group almost similar values of wheeling prices into one cluster. For each cluster a separate RBFNN has been developed for fast and accurate estimation of wheeling prices. These RBFNNs are trained and tested under parallel computing environment. Then final results i.e. estimated values of wheeling prices for generators and loads has been compiled.

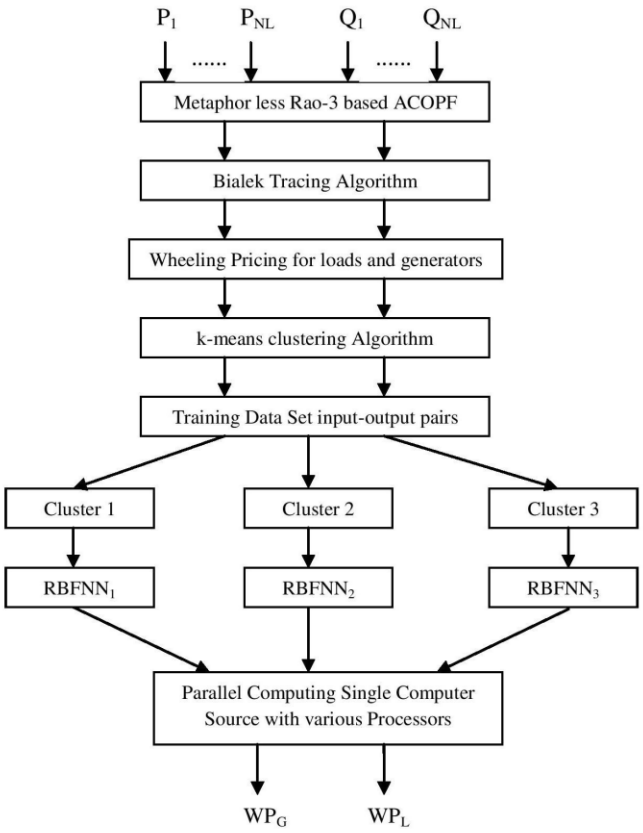


Fig.2: General Block Diagram for Wheeling Price Estimation

7. SOLUTION ALGORITM FLOW CHART

The complete solution algorithm for RBFNN based wheeling pricing estimation using parallel computing has been presented as follows and also depicted in the form of flow chart in Fig. 3.

Step 1. Generate a large number of load patterns by randomly varying the load at each bus in a wide range (± 10% of base case load)

Step 2. For each loading scenario

(a) Execute metaphor-less Rao-3 algorithm based AC-OPF to obtain optimized real and reactive power flows in each wheeling facility of the power market.

(b) Apply Bialek tracing technique to establish the contribution of each generator and load in all the line flows.

(c) Compute wheeling prices on the basis of these optimized flows for each generator and load using MVA-mile method.

Step 3. Present input-output patterns as data set for developing RBFNN. Consider all non-zero real and reactive loads as inputs and wheeling prices for different generators and loads as outputs.

Step 4. Apply k-means clustering to group all the wheeling prices which lie in a closed range.

Step 5. Develop a separate RBFNN for each cluster to estimate the wheeling prices instantly and accurately.

Step 6. Train and test all the RBFNN under parallel computing envionment to obtain speed gain.

Step 7. Stop. Collect final results at the server.

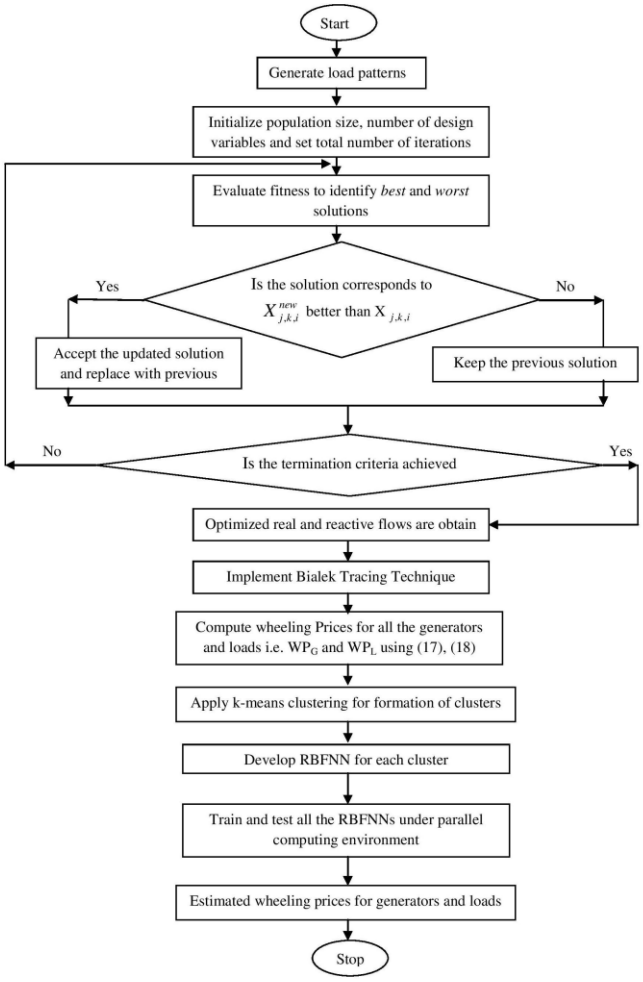


Fig. 3 Solution Algorithm Flow Chart

8. RESULTS AND DISCUSSION

In the present work an attempt has been made to make an accurate prediction of the wheeling prices using radial basis function neural

network (RBFNN). The present approach of wheeling price estimation has been demonstrated on IEEE-30 bus system and Indian utility 146-bus system which clearly establishes the effectiveness of the proposed approach. All the programming modules have been developed in MATLAB 2020a version working platform using Core *i*5 with 2.2 GHz and 8 GB RAM and 64-bit operating system with 4 processing cores.

8.1 IEEE 30-bus System

The IEEE-30 bus system [25] comprises of 41 transmission tracks, 24 load buses and 6 generator buses. Generating units are at bus number 1, 2, 5, 8, 11 and 13. The lower voltage limit for all the buses (generator and load bus) is same i.e. 0.95 pu. However the upper voltage limit for generator buses is 1.1 pu and for all remaining buses (including slack bus) it is 1.05 pu.

Out of 24 load buses, only 21 buses are there which are having non-zero loads. For obtaining continuously varying loading scenario, ±10% perturbation in real and reactive loads has been implemented at all the 21 load buses with non-zero loads. The total annual cost of transmission for this system is 8851.84k$. For the allocation of wheeling prices across generators and loads the total share of generators in this fixed embedded cost is taken as 30% (2655 k$) while share of loads is taken as 70% (6196.3 k$).

To simulate the realistic power market operation and for including maximum possible loading conditions a number of loading patterns are required, to develop a neural network. Hence as many as 501 load patterns have been generated. These generated 501 loading scenarios (input-output patterns) have been divided into two parts and used for training as well as for testing of RBFNN. For training 78% of the patterns were utilized, while for testing remaining 22% unseen loading scenarios were used. Various trails were taken to obtain an optimum structure of the RBFNN. Non zero apparent loads are taken as inputs for the RBFNN, while outputs are the wheeling prices for generators and loads.

8.1.1 *Wheeling Pricing for Generators*

There are total 6 generators in this power system. Employing k-means clustering, 03 desirable clusters have been obtained which are given in Table 1. As can be seen from this table that generator 1 is caught in cluster 1, generators 3, 4 and 6 in cluster 2 and generators 2 and 5 are grouped in cluster 3.

Table 1 : Cluster formation for generators in IEEE-30 bus system

Generators	Minimum WP_G	Maximum WP_G	Cluster
Gen1	2029.5800	2080.9030	1
Gen 3	2.3927	2.8869	2
Gen 4	3.8424	4.2474	
Gen 6	12.3575	14.8160	
Gen 2	330.1468	365.9250	3
Gen 5	190.2849	276.3295	

After having several trials, the optimal structure of RBFNN for cluster 1 was obtained as 21-05-1. Table 2 shows the testing performance of only 10 patterns due to limited space while Figure 4 shows the target and RBFNN output for all the 100 unseen testing patterns. Overlapping of these outputs clarly shows the superior performance of the developed RBFNN. In the fifth column of Table 2 the percentage absolute error has been computed as [(*Target output* – *RBFNN output*)/ *Target output*] x100. Mean absolute percentage error (MAPE) is found as 0.0847 in all the 100 testing patterns. Maximum absolute percentage error for cluster 1 is found as 0.4574. The elapsed time for training of the developed RBFNN is found as 0.302690 seconds.

Table 2 : Wheeling Price prediction for Generators in IEEE-30 bus system (Cluster-1)

TP	Target output	RBFNN output	Error	%Abs. Error
1	2060.499	2063.326	-2.82654	0.13718
2	2074.638	2075.824	-1.18622	0.05718
3	2076.977	2074.403	2.5737	0.12392
4	2075.963	2072.842	3.12022	0.1503
5	2073.804	2075.878	-2.07462	0.10004
6	2059.285	2060.802	-1.5177	0.0737
7	2077.226	2077.37	-0.14354	0.00691
8	2077.065	2076.242	0.82268	0.03961
9	2073.275	2075.094	-1.81866	0.08772
10	2077.175	2073.636	3.53877	0.17036

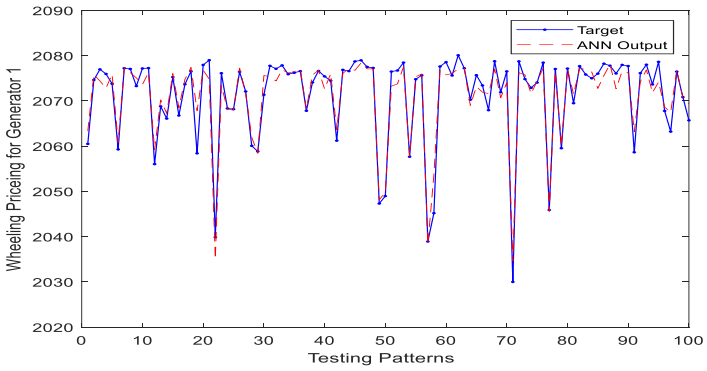


Fig.4 Testing performance of RBFNN in IEEE-30 bus system (Cluster-1)

Cluster 2 is made up of 3 generators namely generator 3, 4 and 6. By having several trials an optimum structure of RBFNN is obtained as 21-24-3. The testing performance of the developed RBFNN for cluster 2 is shown in Table 3 for only 10 tesing

patterns due to restricted space. However testing performance for all the 100 testing patterns have been shown in Fig. 5.

In the fifth column of Table 3, the percentage absolute error has been computed as [(*Target output – RBFNN output*)/ *Target output*] x100. Mean absolute percentage error (MAPE) is found as 0.26320% in all the 100 testing patterns. Maximum absolute percentage error for cluster 2 is found as 1.1976, 0.2702 and 1.3738. The elapsed time for training of this developed RBFNN is also found as 0.302690 seconds due to parallel computing.

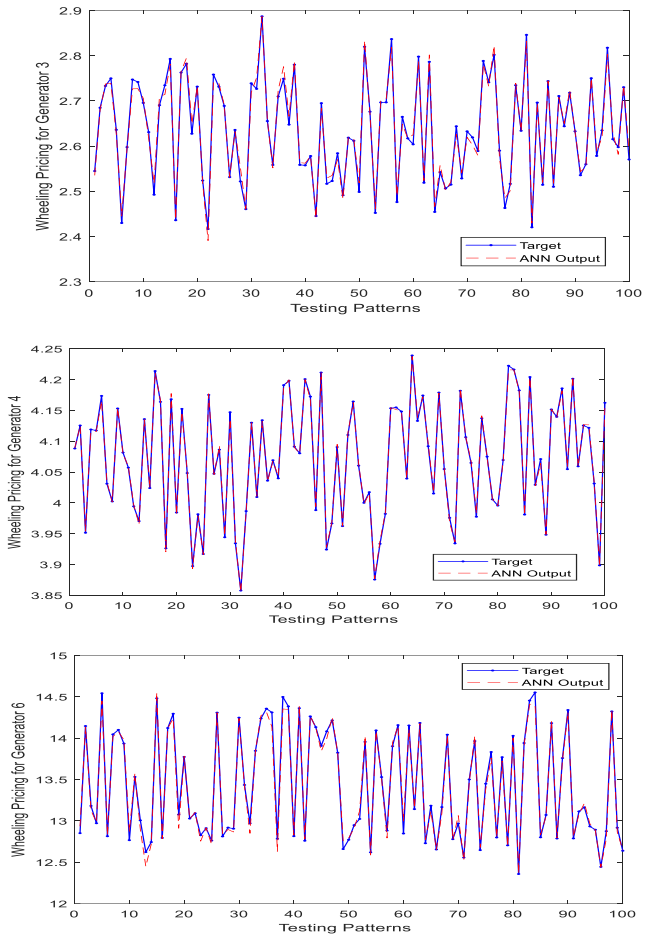


Fig. 5 Testing performance of ANN in IEEE-30 bus system (Cluster-2)

Table-3 : Wheeling Price prediction for Generators in IEEE-30 bus system (Cluster-2)

TP	Target	ANN output	Error	%Abs. Error
1	2.5447	2.53514	0.00956	0.37581
	4.0881	4.08916	-0.00106	0.02597
	12.8513	12.93493	-0.08363	0.65072
2	2.6846	2.67998	0.00462	0.17199
	4.1255	4.12475	0.00075	0.01829
	14.1465	14.13598	0.01052	0.07439
3	2.733	2.73748	-0.00448	0.16377
	3.9514	3.95407	-0.00267	0.0676
	13.1808	13.12628	0.05452	0.41363
4	2.7497	2.73854	0.01116	0.40585
	4.119	4.11642	0.00258	0.06268
	12.9712	12.98253	-0.01133	0.08734
5	2.6359	2.62742	0.00848	0.32155
	4.1174	4.11558	0.00182	0.04425
	14.5426	14.44993	0.09267	0.63724
6	2.4298	2.44502	-0.01522	0.62633
	4.1735	4.16532	0.00818	0.19594
	12.8143	12.82971	-0.01541	0.12027
7	2.5976	2.59572	0.00188	0.07229
	4.031	4.03396	-0.00296	0.07344
	14.0455	14.01909	0.02641	0.18802
8	2.7471	2.72654	0.02056	0.74835
	4.0022	4.00146	0.00074	0.01853
	14.0998	14.07801	0.02179	0.15456
9	2.7417	2.72683	0.01487	0.54225
	4.1531	4.15392	-0.00082	0.01966
	13.9346	13.97905	-0.04445	0.31901
10	2.6955	2.7057	-0.0102	0.37834
	4.0815	4.07959	0.00191	0.04674
	12.7665	12.8702	-0.1037	0.8123

Cluster 3 consists of generator 2 and 5. For this cluster the optimum structure of developed RBFNN is found as 21-18-2. Table 4 shows the testing performance of 10 testing patterns due to restricted space. However Figure 6 shows the testing performance of RBFNN for entire 100 testing patterns.

In the fifth column of Table 4, the percentage absolute error has been computed as [(*Target output – RBFNN output*)/ *Target output*] x100. Mean absolute percentage error (MAPE) is found as 0.0257703% in all the 100 testing patterns. Maximum absolute percentage error for cluster 3 is found as 0.6503 and 1.7218. A summary of all the errors have been given in Table 5.

Table-4 : Wheeling Price prediction for Generators in IEEE-30 bus system (Cluster-3)

TP	Target	ANN output	Error	%Abs. Error
1	360.4243	360.4664	-0.04209	0.01168
	215.1436	215.6731	-0.52951	0.24612
2	363.6351	363.3066	0.3285	0.09034
	196.3219	196.2921	0.02982	0.01519
3	363.2951	363.4313	-0.13621	0.03749
	195.4143	194.8778	0.53648	0.27453
4	362.5051	362.7772	-0.27207	0.07505
	197.2437	198.0433	-0.79959	0.40538

5	363.815	363.9876	-0.17255	0.04743
	196.6369	195.8526	0.78432	0.39887
6	359.2349	358.1034	1.13155	0.31499
	217.6143	220.3002	-2.68593	1.23426
7	362.9344	363.3285	-0.39407	0.10858
	194.7168	193.4927	1.22408	0.62865
8	364.4147	364.3246	0.09014	0.02474
	193.2224	191.72	1.50243	0.77756
9	362.6886	363.1815	-0.49292	0.13591
	198.7581	197.6085	1.14962	0.5784
10	362.8096	363.011	-0.20144	0.05552
	196.0231	194.1773	1.84584	0.94164

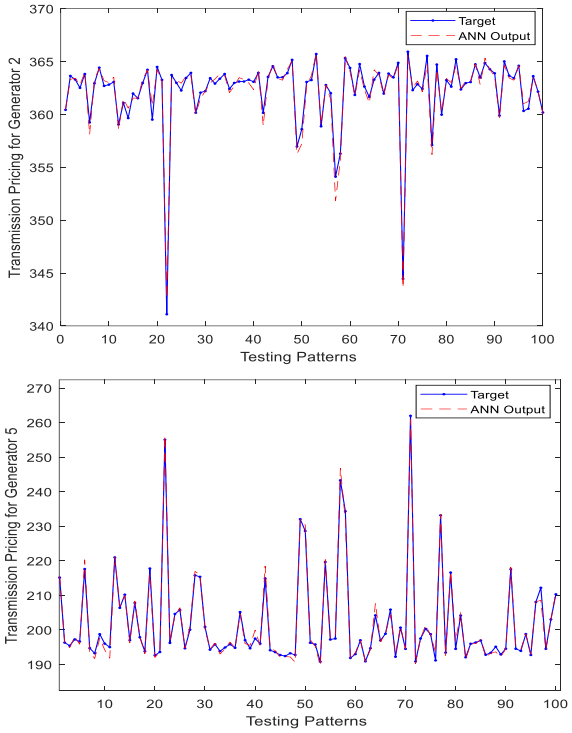


Fig. 6 Testing performance of ANN in IEEE-30 bus system (Cluster-3)

Table-5 : Summary of Wheeling price prediction for Generators

MAPE	Generators for estimating wheeling price	Maximum % Error
0.0847	WP_{g1}	0.4574
0.26320	WP_{g3}	1.1976
	WP_{g4}	0.2702
	WP_{g6}	1.3738
0.0257703	WP_{g2}	0.6503
	WP_{g5}	1.7218

8.1.2 *Wheeling Pricing for Loads*

In IEEE 30-bus system, there are 21 non zero loads. By applying k-means clustering, total 7 clusters have been formed to acquire precision in estimating wheeling prices. The details of load clusters are shown in Table 6 in which maximum and minimum value of wheeling pricing across load is shown for all the clusters.

Table 7: Cluster formation for loads in IEEE-30 bus system

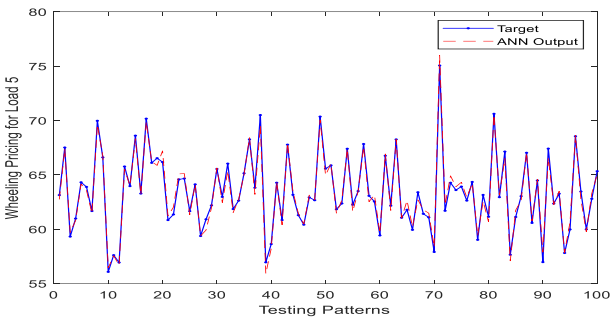
Cluster	Loads	Minimum WP_L	Maximum WP_L	No. of Loads
4	Load 1	86.2330	105.9420	5
	Load 5	54.9831	75.0552	
	Load 7	56.7352	69.6743	
	Load 15	117.5005	158.6143	
	Load 19	40.6804	74.2029	
5	Load 10	496.6136	576.3927	1
6	Load 8	828.1529	954.9922	2
	Load 17	799.6099	890.8661	
7	Load 18	2.5551e+03	2.8686e+03	1
8	Load 3	199.1852	234.8157	3
	Load 14	243.4587	300.0052	
	Load 16	219.6578	295.5389	
9	Load 2	7.9566	9.4750	7
	Load 9	11.0151	13.2156	
	Load 11	0.6609	0.8734	
	Load 12	8.5682	10.3800	
	Load 14	6.4819	7.6414	
	Load 20	1.3846	1.7437	
	Load 21	2.7808	3.7862	
10	Load 6	0.0647	0.0868	2
	Load 13	0.1456	0.2007	

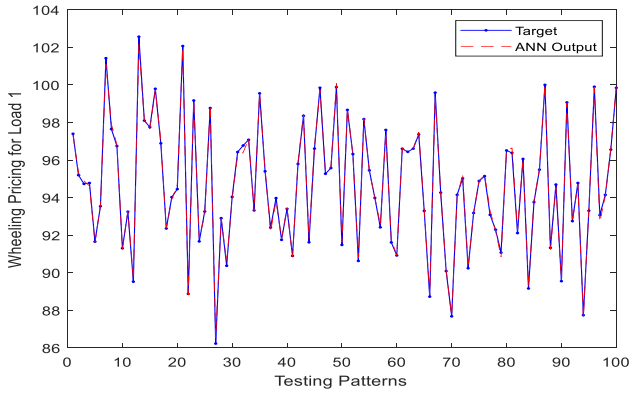
After having several trials, the optimal structure of RBFNN for cluster 4 was obtained as 21-13-5. Table 8 shows the testing performance of only 10 patterns due to limited space while Figure 7 shows the target and RBFNN output for all the 100 unseen testing patterns. Overlapping of these outputs clarly shows the superior performance of the developed RBFNN. In the fifth column of Table 2 the percentage absolute error has been computed as [(*Target output – RBFNN output*)/ *Target output*] x100. Mean absolute percentage error (MAPE) is found as 0.27678722 in all the 100 testing patterns. Maximum absolute percentage error for cluster 4 is found as 0.3684, 1.5245, 0.5574, 0.5833 and 2.1097. The elapsed time for training of the developed RBFNN is found as 0.302690 seconds.

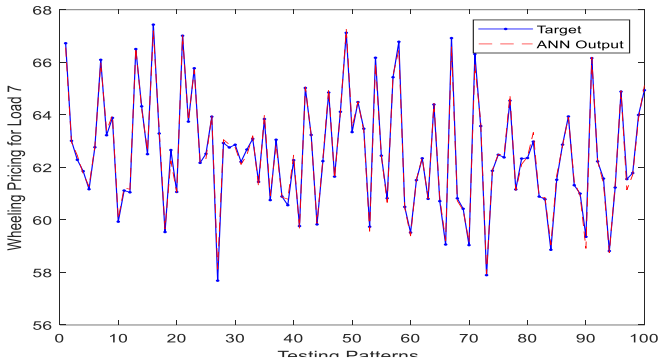
Table 8: Wheeling Price prediction for loads in IEEE-30 bus system (Cluster-4)

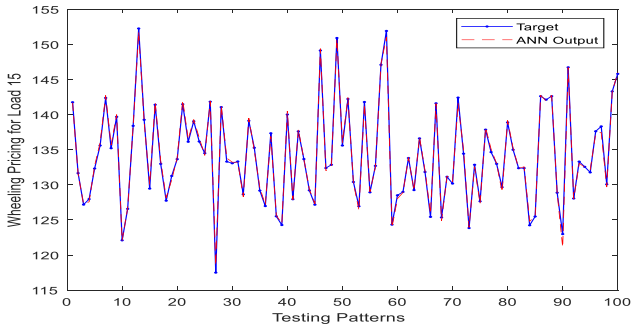
TP	Load	Target	ANN output	Error	% Abs. Error

	unit				
1	Load 1	97.3975	97.41715	-0.01965	0.02017
	Load 5	63.1311	63.35419	-0.22309	0.35337
	Load 7	66.7269	66.61158	0.11532	0.17282
	Load 15	141.7752	141.6811	0.09408	0.06636
	Load 19	60.1611	59.88105	0.28005	0.46551
2	Load 1	95.1963	95.31335	-0.11705	0.12295
	Load 5	67.5094	67.71864	-0.20924	0.30994
	Load 7	63.0117	63.0675	-0.0558	0.08855
	Load 15	131.6742	131.8802	-0.20602	0.15646
	Load 19	52.7794	52.46224	0.31716	0.60091
3	Load 1	94.7325	94.90091	-0.16841	0.17777
	Load 5	59.3125	59.04034	0.27216	0.45886
	Load 7	62.2885	62.37831	-0.08981	0.14419
	Load 15	127.1874	127.371	-0.18363	0.14438
	Load 19	65.5475	65.32955	0.21795	0.3325
4	Load 1	94.7809	94.81192	-0.03102	0.03273
	Load 5	60.9723	61.31825	-0.34595	0.56739
	Load 7	61.8565	61.89685	-0.04035	0.06523
	Load 15	127.9781	127.7797	0.19842	0.15504
	Load 19	50.6288	50.70876	-0.07996	0.15794
5	Load 1	91.656	91.51953	0.13647	0.1489
	Load 5	64.3088	64.04123	0.26757	0.41607
	Load 7	61.1691	61.17463	-0.00553	0.00905
	Load 15	132.3305	132.5911	-0.26055	0.19689
	Load 19	61.0408	61.57866	-0.53786	0.88115
6	Load 1	93.5406	93.4631	0.0775	0.08285
	Load 5	63.883	63.94308	-0.06008	0.09405
	Load 7	62.7669	62.64866	0.11824	0.18838
	Load 15	135.6104	135.694	-0.08359	0.06164
	Load 19	58.468	58.66214	-0.19414	0.33204
7	Load 1	101.4214	101.3095	0.11188	0.11031
	Load 5	61.6651	61.97001	-0.30491	0.49446
	Load 7	66.0963	66.11143	-0.01513	0.02289
	Load 15	142.3908	142.5997	-0.20893	0.14673
	Load 19	55.8901	56.20489	-0.31479	0.56323
8	Load 1	97.65	97.56238	0.08762	0.08973
	Load 5	69.9671	69.32873	0.63837	0.91238
	Load 7	63.224	63.37892	-0.15492	0.24503
	Load 15	135.2215	135.1489	0.07259	0.05368
	Load 19	58.3844	58.91883	-0.53443	0.91537
9	Load 1	96.7454	96.81826	-0.07286	0.07531
	Load 5	66.5995	66.7518	-0.1523	0.22868
	Load 7	63.8858	63.81031	0.07549	0.11817
	Load 15	139.7	139.6909	0.00907	0.00649
	Load 19	63.5943	63.72054	-0.12624	0.19851
10	Load 1	91.3039	91.50364	-0.19974	0.21877
	Load 5	56.0679	56.24598	-0.17808	0.31761
	Load 7	59.933	59.93982	-0.00682	0.01139
	Load 15	122.1012	121.8171	0.28414	0.23271
	Load 19	55.8616	55.26195	0.59965	1.07346









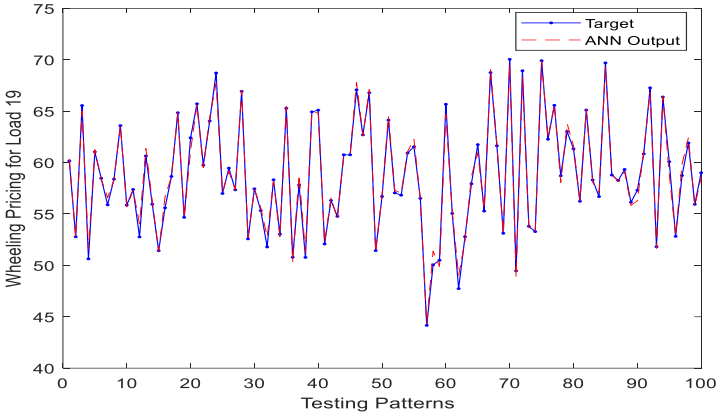


Fig. 7 Testing performance of ANN in IEEE-30 bus system (Cluster-4)

After having several trials, the optimal structure of RBFNN for cluster 1 was obtained as 21-06-1. Table 9 shows the testing performance of only 10 patterns due to limited space while Figure 8 shows the target and RBFNN output for all the 100 unseen testing patterns. Overlapping of these outputs clarly shows the superior performance of the developed RBFNN. In the fifth column of Table 2 the percentage absolute error has been computed as [(*Target output – RBFNN output*)/ *Target output*] x100. Mean absolute percentage error (MAPE) is found as 0.1079 in all the 100 testing patterns. Maximum absolute percentage error for cluster 1 is found as 0.9759. The elapsed time for training of the developed RBFNN is found as 0.302690 seconds.

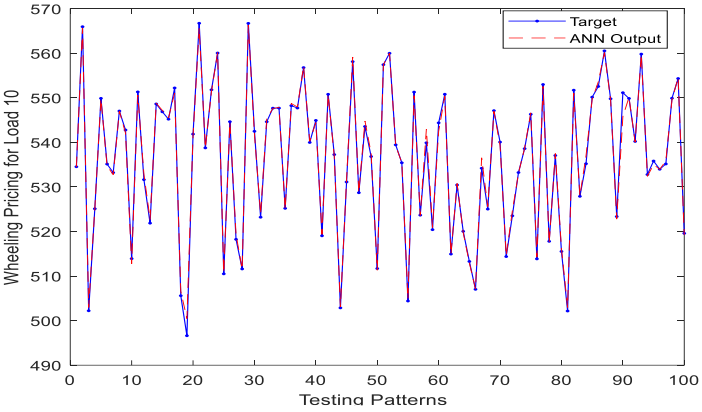


Fig. 8 Testing performance of ANN in IEEE-30 bus system (Cluster-5)

Table 9: Wheeling Price prediction for loads in IEEE-30 bus system (Cluster-5)

TP	Load unit	Target	ANN output	Error	% Abs. Error
1	Load10	534.5354	534.4787	0.05668	0.0106
2	Load10	565.9721	565.6724	0.29972	0.05296
3	Load1	502.214	502.9685	-	0.15005

	0	9		0.75358	
4	Load10	525.1114	525.1031	0.00832	0.00158
5	Load10	549.887	549.7689	0.11813	0.02148
6	Load10	535.1343	534.7164	0.41792	0.0781
7	Load10	533.2243	532.761	0.4633	0.08689
8	Load10	547.0462	547.0206	0.02564	0.00469
9	Load10	542.7897	542.1032	0.68651	0.12648
10	Load10	513.9104	512.7632	1.14723	0.22324

After having several trials, the optimal structure of RBFNN for cluster 6 was obtained as 21-16-2. Table 10 shows the testing performance of only 10 patterns due to limited space while Figure 9 shows the target and RBFNN output for all the 100 unseen testing patterns. Overlapping of these outputs clarly shows the superior performance of the developed RBFNN. In the fifth column of Table 10 the percentage absolute error has been computed as [(*Target output – RBFNN output*)/ *Target output*] x100. Mean absolute percentage error (MAPE) is found as 0.2317 in all the 100 testing patterns. Maximum absolute percentage error for cluster 6 is found as 1.6708 and 0.5346. The elapsed time for training of the developed RBFNN is found as 0.302690 seconds.

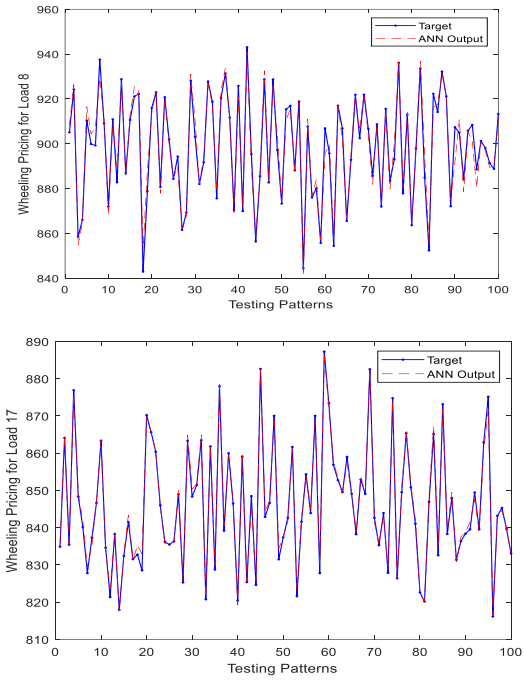


Fig. 9 Testing performance of ANN in IEEE-30 bus system (Cluster-6)

Table 10: Wheeling Price prediction for loads in IEEE-30 bus system (Cluster-6)

TP	Load unit	Target	ANN output	Error	% Abs. Error
1	Load 8	905.0864	908.3151	-3.22873	0.35673
	Load 17	834.8993	836.0191	-1.11978	0.13412
2	Load 8	924.2664	926.4069	-2.14053	0.23159
	Load 17	864.1005	864.7056	-0.60513	0.07003
3	Load 8	858.5029	854.7518	3.75109	0.43693
	Load 17	835.4308	834.6363	0.79453	0.0951
4	Load 8	866.0712	864.9901	1.08114	0.12483
	Load 17	876.914	875.9036	1.01045	0.11523
5	Load 8	910.3861	916.5783	-6.19216	0.68017
	Load 17	848.361	849.2545	-0.89345	0.10532
6	Load 8	899.8849	904.333	-4.44807	0.49429
	Load 17	840.1935	841.8232	-1.62974	0.19397
7	Load 8	899.2019	907.5365	-8.33461	0.92689
	Load 17	827.8247	829.2495	-1.4248	0.17211
8	Load 8	937.5637	929.1352	8.4285	0.89898
	Load 17	837.3153	835.1869	2.12844	0.2542
9	Load 8	909.0777	910.9553	-1.8776	0.20654
	Load 17	846.6879	847.367	-0.67907	0.0802
10	Load 8	871.9805	868.3061	3.67445	0.42139
	Load 17	863.3833	863.4959	-0.11262	0.01304

After having several trials, the optimal structure of RBFNN for cluster 7 was obtained as 21-09-1. Table 11 shows the testing performance of only 10 patterns due to limited space while Figure 10 shows the target and RBFNN output for all the 100 unseen testing patterns. Overlapping of these outputs clarly shows the superior performance of the developed RBFNN. In the fifth column of Table 11 the percentage absolute error has been computed as [(*Target output – RBFNN output*)/ *Target output*] x100. Mean absolute percentage error (MAPE) is found as 0.1093

in all the 100 testing patterns. Maximum absolute percentage error for cluster 7 is found as 0.8658. The elapsed time for training of the developed RBFNN is found as 0.302690 seconds.

Table 11: Wheeling Price prediction for loads in IEEE-30 bus system (Cluster-7)

TP	Load unit	Target	ANN output	Error	% Abs. Error
1	Load 18	2687.263	2686.868	0.39445	0.01468
2	Load 18	2628.69	2626.749	1.94156	0.07386
3	Load 18	2831.243	2828.492	2.75111	0.09717
4	Load 18	2767.434	2766.515	0.91935	0.03322
5	Load 18	2690.577	2686.553	4.02427	0.14957
6	Load 18	2733.885	2736.392	-2.50737	0.09171
7	Load 18	2703.194	2697.214	5.98041	0.22123
8	Load 18	2636.52	2642.798	-6.27841	0.23813
9	Load 18	2645.863	2641.183	4.67969	0.17687
10	Load 18	2829.604	2833.344	-3.74033	0.13219

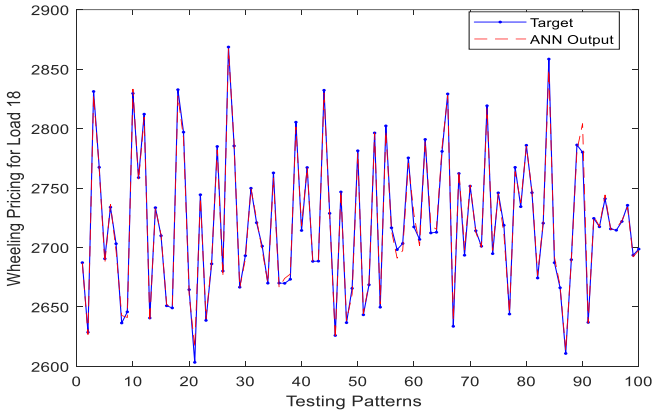


Fig. 10 Testing performance of ANN in IEEE-30 bus system (Cluster-7)

After having several trials, the optimal structure of RBFNN for cluster 8 was obtained as 21-06-3. Table 12 shows the testing performance of only 10 patterns due to limited space while Figure 11 shows the target and RBFNN output for all the 100 unseen testing patterns. Overlapping of these outputs clarly shows the superior performance of the developed RBFNN. In the fifth column of Table 12 the percentage absolute error has been computed as [(*Target output – RBFNN output*)/ *Target output*] x100. Mean absolute percentage error (MAPE) is found as 0.1734 in all the 100 testing patterns. Maximum absolute percentage error for cluster 8 is found as 0.3557, 1.5788 and 1.6594. The elapsed time for training of the developed RBFNN is found as 0.302690 seconds.

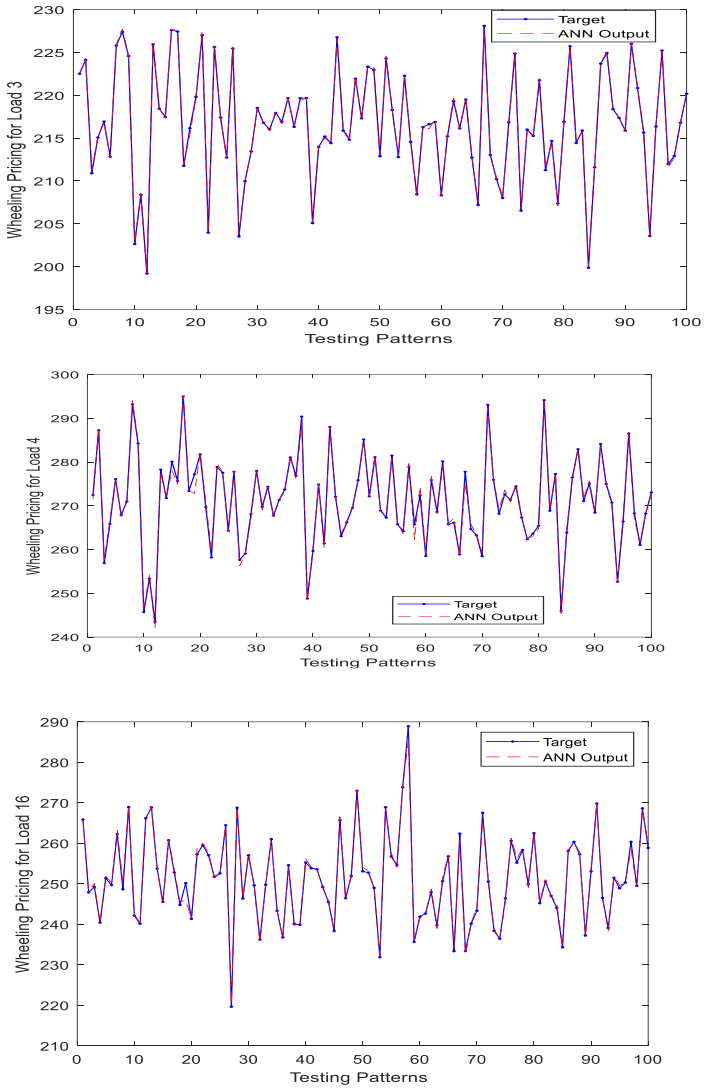


Fig. 11 Testing performance of ANN in IEEE-30 bus system (Cluster-8)

Table 12: Wheeling Price prediction for loads in IEEE-30 bus system (Cluster-8)

TP	Load unit	Target	ANN output	Error	% Abs. Error
1	Load 3	222.5147	222.5147	0.00004	0.00002
	Load 4	272.4497	271.6368	0.81295	0.29839
	Load 16	265.8845	265.6295	0.25499	0.0959
2	Load	224.140	224.2819	-	0.06299

	3	7		0.14118	
	Load 4	287.2671	287.7477	-0.48063	0.16731
	Load 16	247.8932	248.1445	-0.25134	0.10139
3	Load 3	210.8981	211.1644	-0.26628	0.12626
	Load 4	256.9002	257.6621	-0.76193	0.29659
	Load 16	249.2333	250.0515	-0.81815	0.32827
4	Load 3	215.0707	215.1757	-0.105	0.04882
	Load 4	265.8582	266.8068	-0.94857	0.3568
	Load 16	240.4355	240.2206	0.21493	0.08939
5	Load 3	216.9313	216.8269	0.10443	0.04814
	Load 4	276.1049	275.9128	0.1921	0.06958
	Load 16	251.3874	251.8444	-0.45697	0.18178
6	Load 3	212.8329	212.7731	0.05984	0.02812
	Load 4	267.8874	267.5693	0.31814	0.11876
	Load 16	249.6989	250.2546	-0.55565	0.22253
7	Load 3	225.803	225.934	-0.13099	0.05801
	Load 4	270.9744	271.2442	-0.26984	0.09958
	Load 16	262.2987	263.2267	-0.92803	0.35381
8	Load 3	227.4087	227.747	-0.33831	0.14877
	Load 4	293.1941	294.0226	-0.8285	0.28258
	Load 16	248.6463	249.9775	-1.33121	0.53538
9	Load 3	224.5741	224.5737	0.00045	0.0002
	Load 4	284.2265	284.6816	-0.45512	0.16013
	Load 16	268.9299	269.3301	-0.40016	0.1488
10	Load 3	202.6425	202.736	-0.09352	0.04615
	Load 4	245.6922	246.0944	-0.40217	0.16369
	Load 16	242.1368	242.3852	-0.24842	0.10259

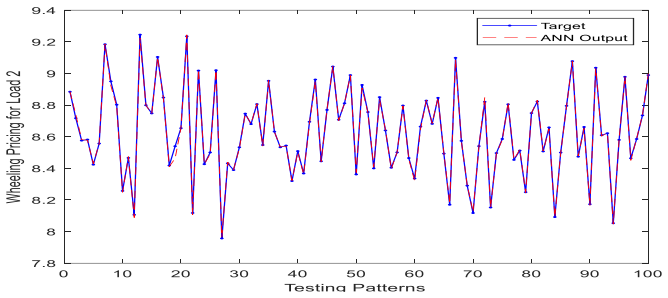
After having several trials, the optimal structure of RBFNN for cluster 9 was obtained as 21-22-7. Table 13 shows the testing performance of only 10 patterns due to limited space while Figure 12 shows the target and RBFNN output for all the 100 unseen testing patterns. Overlapping of these outputs clarly shows the superior performance of the developed RBFNN. In the fifth column of Table 13 the percentage absolute error has been computed as [(*Target output – RBFNN output*)/ *Target output*] x100. Mean absolute percentage error (MAPE) is found as 0.1421584 in all the 100 testing patterns. Maximum absolute percentage error for cluster 9 is found as 0.9006, 0.4731, 1.7118, 1.2530, 1.1632, 1.3736 and 1.1139. The elapsed time for training of the developed RBFNN is found as 0.302690 seconds.

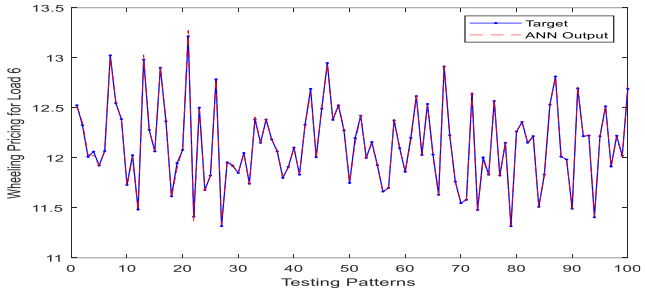
Table 13: Wheeling Price prediction for loads in IEEE-30 bus system (Cluster-9)

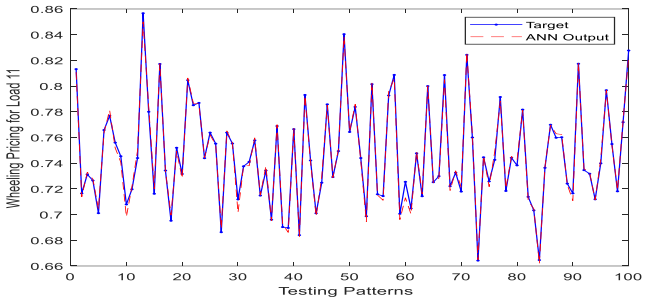
TP	**Load unit**	**Target**	**ANN output**	**Error**	**% Abs. Error**
1	Load 2	8.8841	8.88453	-0.00043	0.00484
	Load 9	12.5262	12.51913	0.00707	0.05644
	Load 11	0.8132	0.81085	0.00235	0.28894
	Load 12	9.5413	9.54437	-0.00307	0.03221
	Load 14	7.249	7.254	-0.005	0.06893
	Load 20	1.6461	1.64523	0.00087	0.05309
	Load 21	3.546	3.54967	-0.00367	0.10349
2	Load 2	8.7169	8.73871	-0.02181	0.25022
	Load 9	12.3269	12.35378	-0.02688	0.21807
	Load 11	0.7165	0.71371	0.00279	0.38967
	Load 12	9.6158	9.65952	-0.04372	0.45463
	Load 14	7.1512	7.17794	-0.02674	0.37386
	Load 20	1.651	1.65518	-0.00418	0.25294
	Load 21	3.3482	3.35513	-0.00693	0.20698
3	Load 2	8.5755	8.58102	-0.00552	0.06441
	Load 9	12.0104	12.03173	-0.02133	0.1776
	Load	0.7313	0.73301	-	0.23336

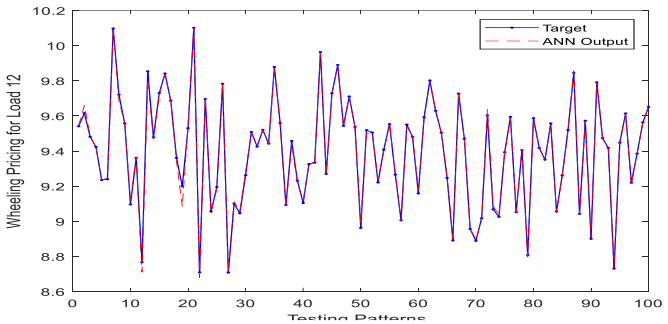
	11			0.00171	
	Load 12	9.4813	9.48574	-0.00444	0.04678
	Load 14	7.0246	7.0266	-0.002	0.02842
	Load 20	1.5922	1.59375	-0.00155	0.09711
	Load 21	3.1272	3.12317	0.00403	0.12888
4	Load 2	8.5817	8.57759	0.00411	0.04786
	Load 9	12.0623	12.02504	0.03726	0.30891
	Load 11	0.7265	0.72763	-0.00113	0.15602
	Load 12	9.424	9.42476	-0.00076	0.00804
	Load 14	7.0128	7.00568	0.00712	0.10154
	Load 20	1.6332	1.63555	-0.00235	0.14382
	Load 21	3.4793	3.48602	-0.00672	0.19316
5	Load 2	8.4232	8.42894	-0.00574	0.0681
	Load 9	11.9234	11.92428	-0.00088	0.00734
	Load 11	0.7011	0.70417	-0.00307	0.43787
	Load 12	9.2358	9.23662	-0.00082	0.00889
	Load 14	6.8925	6.89886	-0.00636	0.09221
	Load 20	1.5397	1.53995	-0.00025	0.01632
	Load 21	3.1916	3.19024	0.00136	0.04254
6	Load 2	8.5567	8.55676	-0.00006	0.00075
	Load 9	12.0687	12.07093	-0.00223	0.01844
	Load 11	0.7658	0.76392	0.00188	0.24562
	Load 12	9.2398	9.24662	-0.00682	0.07378
	Load 14	7.0498	7.05435	-0.00455	0.0645
	Load 20	1.5564	1.55669	-0.00029	0.01856
	Load 21	3.0036	2.99965	0.00395	0.13143
7	Load 2	9.1842	9.18576	-0.00156	0.01704
	Load 9	13.025	13.00071	0.02429	0.18646
	Load 11	0.7769	0.78124	-0.00434	0.55851
	Load 12	10.0979	10.09652	0.00138	0.01368
	Load 14	7.5516	7.55476	-0.00316	0.04189
	Load 20	1.6975	1.69837	-0.00087	0.05108
	Load 21	3.6943	3.68746	0.00684	0.18513
8	Load 2	8.9498	8.92309	0.02671	0.29843
	Load 9	12.5462	12.5654	-0.0192	0.15301
	Load 11	0.756	0.75184	0.00416	0.55067
	Load 12	9.721	9.68666	0.03434	0.35321
	Load 14	7.1833	7.15995	0.02335	0.32508
	Load 20	1.6071	1.60282	0.00428	0.26601
	Load 21	3.1469	3.13351	0.01339	0.42565
9	Load 2	8.8021	8.79131	0.01079	0.12261
	Load 9	12.3881	12.37874	0.00936	0.07558
	Load 11	0.7454	0.74081	0.00459	0.6158
	Load 12	9.5578	9.55887	-0.00107	0.01117
	Load 14	7.1504	7.14726	0.00314	0.04396
	Load 20	1.5765	1.57202	0.00448	0.28439
	Load 21	3.1542	3.1459	0.0083	0.2631
10	Load 2	8.257	8.24852	0.00848	0.10267
	Load 9	11.7296	11.71747	0.01213	0.10342
	Load 11	0.7079	0.69832	0.00958	1.3535
	Load 12	9.0964	9.12093	-0.02453	0.26963
	Load 14	6.714	6.71697	-0.00297	0.04424
	Load 20	1.5658	1.56743	-0.00163	0.10425
	Load	3.3716	3.38259	-	0.3261

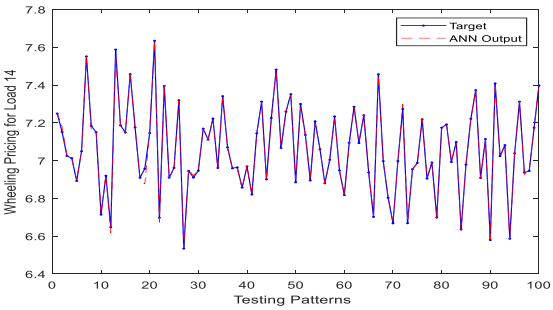
	21			0.01099	

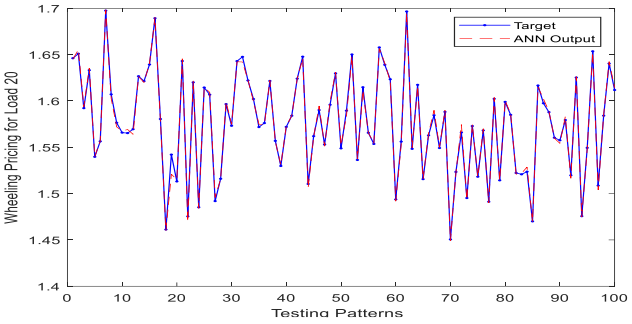












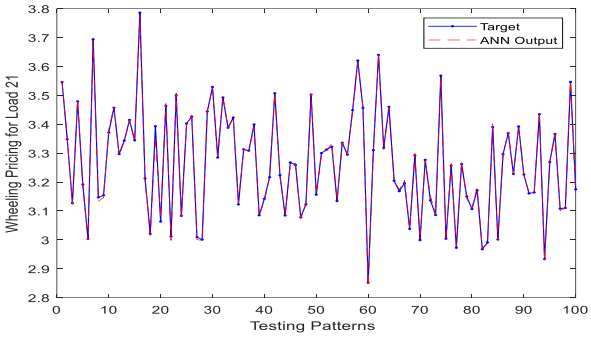


Fig. 12 Testing performance of ANN in IEEE-30 bus system (Cluster-9)

After having several trials, the optimal structure of RBFNN for cluster 10 was obtained as 21-06-2. Table 14 shows the testing performance of only 10 patterns due to limited space while Figure 13 shows the target and RBFNN output for all the 100 unseen testing patterns. Overlapping of these outputs clarly shows the superior performance of the developed RBFNN. In the fifth column of Table 14 the percentage absolute error has been computed as [(*Target output – RBFNN output*)/ *Target output*] x100. Mean absolute percentage error (MAPE) is found as 0.2644 in all the 100 testing patterns. Maximum absolute percentage error for cluster 10 is found as 1.2820 and 1.4526. The elapsed time for training of the developed RBFNN is found as 0.302690 seconds.

Table 14: Wheeling Price prediction for loads in IEEE-30 bus system (Cluster-10)

TP	Load unit	Target	ANN output	Error	% Abs. Error
1	Load 6	0.0761	0.07617	-0.00007	0.09459
	Load 13	0.1797	0.17958	0.00012	0.06931
2	Load 6	0.0743	0.07415	0.00015	0.20666
	Load 13	0.1843	0.18494	-0.00064	0.34688
3	Load 6	0.0803	0.08063	-0.00033	0.41597
	Load	0.1721	0.17242	-	0.1849

	13			0.00032	
4	Load 6	0.0775	0.07771	-0.00021	0.26535
	Load 13	0.1773	0.1776	-0.0003	0.17147
5	Load 6	0.0675	0.06731	0.00019	0.27804
	Load 13	0.1715	0.17174	-0.00024	0.14036
6	Load 6	0.0795	0.07948	0.00002	0.03133
	Load 13	0.1806	0.18021	0.00039	0.21574
7	Load 6	0.0795	0.07983	-0.00033	0.41683
	Load 13	0.1943	0.19357	0.00073	0.37765
8	Load 6	0.0754	0.07443	0.00097	1.28197
	Load 13	0.1597	0.16168	-0.00198	1.24162
9	Load 6	0.0733	0.07323	0.00007	0.08947
	Load 13	0.165	0.16519	-0.00019	0.11289
10	Load 6	0.0753	0.07496	0.00034	0.45479
	Load 13	0.152	0.15403	-0.00203	1.33472

Table 15: Pa bus system

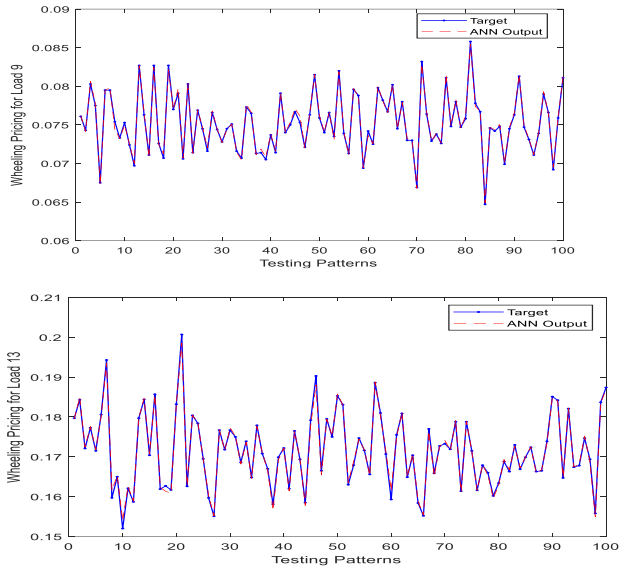


Fig. 13 Testing performance of ANN in IEEE-30 bus system (Cluster-10)

Parallel computing has been employed in this paper to further reduce the developing time for RBFNN. For total 10 clusters (3 for generators and 7 for loads) ten separate RBFNNs have been developed in parallel computing environment. In parallel computing, due to the tasks assigned to all the workers, each processor completes its task at the same time. The computational work was carried out using MATLAB 2020.a version on a personal computer having a 1.8 GHz Intel Processor, 8GB RAM, Core *i*5, and 64-bit operating system with 4 processing cores.

Table 15 shows the elapsed time required by each worker while ten RBFNNs were developed simultaneously by 4 workers using parallel computing. However, the maximum elapsed time in this case i.e. parallel computing was 10.6956 seconds. However if all the RBFNNs were developed sequentially then total elepsed time by each worker will be 27.0675 seconds. Then the speed gain has been calculated as follows:

Speed gain = Sequential time / parallel computing time

A significant speed gain of 2.5307 has been obtained under parallel computing environment..

Parallel computing details for Wheeling pricing in IEEE 30-

Worker	Clusters	Structure of RBFNN	Elapsed Time (s)	Speed Gain
1	Cluster-1	21-05-1	6.5915	2.5307
	Cluster-2	21-24-3		
	Cluster-3	21-18-2		
2	Cluster-4	21-13-5	5.5087	
	Cluster-5	21-06-1		
3	Cluster-6	21-16-2	4.2717	
	Cluster-7	21-09-1		
	Cluster-8	21-06-3		
	Cluster-10	21-06-2		
4	Cluster-9	21-22-7	10.6956	

8.2 Indian Utility 146-bus System

The proposed approch of RBFNN and parallel computing based wheeling price estimation has also been demonstrated on Indian Utility 146-bus system [26]. The Indian utility 146-bus system comprises of 24 generator buses, 146 load buses and 171 transmission lines. Out of 24 generators, only 7 non-zero generators and out of 146 loads only 44 non-zero loads have been considered for computing wheeling prices. In this test system, minimum and maximum voltage limit at all the buses irrespective of generator and load is taken as 0.9 pu and 1.1 pu respectively.

The total transmission cost (annualized) of the system is given as 50213.516 Rs. in lakh. As earlier, the share of generators and loads in this fixed embedded cost is taken as 30% (15064.0548 Rs. in lakh) and 70% (35149.4612 Rs. in lakh) respectively.

Using k-means clustering technique, total 3 clusters have been formed for generators and 08 clusters for loads, which are given in Table 16 and Table 18 respectively. Due to limited space, the results (Target output, RBFNN output and percentage absolute error) for only two clusters, one of generator (cluster number 1) and one of load (cluster number 6) have been presented in the Table 17 and 19 respectively.

Table 16 Cluster Formation for Generators in Indian utility 146-bus system

Gen. Unit	Cluster
Gen1, Gen 3 and Gen 5	1
Gen 6 and Gen 11	2
Gen 12 and Gen 21	3

After having several trials, the optimal structure of RBFNN for cluster 1 was obtained as 07-14-3. Table 17 shows the testing performance of only 10 patterns due to limited space. In the fifth column of Table 17 the percentage absolute error has been computed as [(*Target output – RBFNN output*)/ *Target output*] x100. Mean absolute percentage error (MAPE) is found as 0.0989in all the 100 testing patterns. Maximum absolute percentage error for cluster 10 is found as 0.7634, 1.0328 and 0.60876. The elapsed time for training of the developed RBFNN is found as 0.608740 seconds.

Table-17 : Wheeling Price prediction for Generators in Indian utility 146-bus system (Cluster-1)

TP	Target	ANN output	Error	%Abs. Error
1	190.5346	190.4518	0.08281	0.04346
	136.1883	136.4401	-0.25183	0.18491
	161.4715	161.4301	0.04143	0.02566
2	181.8629	181.7657	0.09717	0.05343
	121.4069	121.6598	-0.25289	0.2083
	176.9336	177.0523	-0.11868	0.06708
3	178.5163	178.5656	-0.04926	0.02759
	124.8911	125.1519	-0.26075	0.20878
	173.3973	173.3322	0.06514	0.03757
4	193.2379	193.3486	-0.11069	0.05728
	123.3425	123.5447	-0.20224	0.16397
	146.7873	146.8736	-0.08629	0.05879
5	186.5898	186.6287	-0.03887	0.02083
	122.037	122.1879	-0.15085	0.12361
	180.8535	180.801	0.05255	0.02905
6	181.8405	181.9231	-0.08256	0.0454
	123.6651	123.6512	0.01395	0.01128
	163.8977	163.8035	0.09424	0.0575
7	179.6072	179.5365	0.07067	0.03935
	126.6688	127.0116	-0.34277	0.2706
	165.9594	165.9459	0.01348	0.00813
8	202.3325	202.3973	-0.06476	0.03201
	136.8718	137.0022	-0.13037	0.09525
	146.287	146.4487	-0.16174	0.11056
9	184.8707	184.5955	0.27525	0.14889
	131.1383	130.8476	0.29069	0.22167
	170.9474	170.8828	0.06456	0.03777
10	183.5902	183.8085	-0.21832	0.11892
	124.7977	124.6494	0.14828	0.11882
	173.4639	173.1471	0.3168	0.18263

By employing k-means clustering technique total 08 clusters have been formed for all the non-zero load buses for the instant and accurate prediction of wheeling prices. These load clusters have been given in Table 18.

Table 18 Cluster Formation for loads in Indian utility 146-bus system

S.No.	Cluster	Load Buses	S.No.	Cluster	Load Buses
01	04	1, 58, 3, 48, 6, 4 and12	05	08	46, 4, 8, 51, 18, 53 and 2
02	05	13, 60, 16, 17, 65, 19, 64 and 81	06	09	58, 15, 63, 22 and 37
03	06	72, 30 and 52	07	10	66, 40, 27, 73, 77and 31
04	07	33, 36, 69, 38, 39, 71 and 45	08	11	24 and 88

Due to limited space the results for only cluster 6 for loads have been presented. After having several trials, the optimal structure of RBFNN for cluster 6 was obtained as 44-27-3. Table 19 shows the testing performance of only 10 patterns due to limited space. In the fifth column of Table 19 the percentage absolute error has been computed as [(*Target output – RBFNN output*)/ *Target output*] x100. Mean absolute percentage error (MAPE) is found as 0.146298 in all the 100 testing patterns. Maximum absolute percentage error for cluster 10 is found as 1.20953, 1.71057 and 0.74182. The elapsed time for training of the developed RBFNN is found as 1.59640 seconds.

Table-19 : Wheeling Price prediction for loads in Indian utility 146-bus system (Cluster-6)

TP	Target	ANN output	Error	%Absolute

				Error
01	31.7653	31.63421	0.13109	0.0041
	29.0428	29.04482	-0.00202	0.00696
	30.8253	30.81704	0.00826	0.0268
02	27.6387	27.63023	0.00847	0.03065
	32.3382	32.3276	0.0106	0.03279
	68.6072	68.61313	-0.00593	0.00865
03	26.9922	26.99517	-0.00297	0.01101
	31.9393	31.9366	0.0027	0.00846
	71.4769	71.48119	-0.00429	0.006
04	29.2717	29.27833	-0.00663	0.02265
	33.3557	33.34515	0.01055	0.03164
	76.3746	76.39794	-0.02334	0.03056
05	27.5738	27.57187	0.00193	0.007
	30.7048	30.6948	0.01	0.03258
	70.6702	70.6997	-0.0295	0.04174
06	29.1438	29.13763	0.00617	0.02117
	31.8356	31.82873	0.00687	0.02158
	73.6958	73.70417	-0.00837	0.01136
07	28.931	28.93223	-0.00123	0.00424
	32.8793	32.86729	0.01201	0.03652
	69.7584	69.76649	-0.00809	0.0116
08	28.9167	28.911	0.0057	0.01971
	36.1781	36.17106	0.00704	0.01946
	76.5066	76.49478	0.01182	0.01545
09	29.309	29.30208	0.00692	0.02362
	34.4665	34.40089	0.06561	0.19036
	75.3039	75.25005	0.05385	0.07151
10	28.9698	28.96839	0.00141	0.00486
	33.584	33.58107	0.00293	0.00873
	73.1159	73.1103	0.0056	0.00766

9. CONCLUSION

This paper presents a novel approach of estimating wheeling prices among system users employing artificial neural network under parallel computing environment. It has been found that the use of parallel computing have established a significant amount of speed gain. Embedded cost based wheeling price methods have been found a considerable recognition in recovering fixed transmission cost. This paper has presented an MVA-mile method of wheeling pricing, which is a type of embedded cost based methods. In the literature wheeling prices over transmission lines have been found, however in this paper wheeling prices across each generator and load have been computed using Bialek tracing technique. Moreover metaphor-less Rao-3 optimization algorithm has been implemented for determining optimal flows over all the transmission lines of the given power system. This algorithm was found as a superior optimization technique over all the available Meta heuristic nature inspired optimization methods. For considering continuously varying loading scenario of the power market, ANN based approach has been developed for estimating wheeling prices, which is found quite efficient and successful. Since for large power markets the wheeling prices are found quite heterogeneous, so to estimate those far different wheeling prices k-means clustering technique has been employed to group all the wheeling prices which lie in a closed range into one cluster. For instant and accurate estimation of wheeling prices, separate RBFNNs have been developed for each cluster. All these RBFNNs have been trained and tested under parallel computing environment. The proposed RBFNN based approach of estimating wheeling prices has been demonstrated on IEEE 30-bus system and Indian Utility 146-bus system. In future this work can be carried out for a bigger power market with bilateral and multilateral transactions.

Acknowledgement

The authors sincerely acknowledge the Management ITM University, Gwalior, for providing facilities to carry out research work.

Appendix A1

A1. Bialek Tracing Algorithm

The only requirement of the Bialek method is that it should satisfy Kirchhoff's current law completely. For this, this method is equally applicable for real and reactive power flows. In Bialek tracing, the allocation to each user is done through topology analysis. The Bialek tracing method is divided into two algorithms 'upstream' and 'downstream' based on the proposed sharing principle, which performs tracing for each user to calculate wheeling pricing. The upstream algorithm displays how much of the total current flowing to a node on a particular generator bus is present [7]. The total apparent power flow S_i through the node given all the currents can be expressed as follows:

$$S_i = \sum_{j \in v_i^{(u)}} \left| S_{ij} \right| + S_{Gi} \qquad for\ i = 1,2,3,...n \tag{22}$$

Where $v_i^{(u)}$ is the set of buses supplying directly to bus i. S_{ij} shows the apparent power flow from node i to line $i-j$ and S_{Gi} is the generation at node i. Along with this 'n' is the total number of buses.

Since each outflow down the line from a node is dependent only on the voltage gradient and the impedance of the line, according to

the principle of proportional sharing, each current that leads to the node has the same ratio of current to the total nodal flow S_i. The line flow may be written in terms of nodal flow at bus j as $|S_{ij}| = x_{ji} S_j$ Thus

$$S_i = \sum_{j \in V_i^{(u)}} x_{ji} S_j + S_{Gi} \quad (23)$$

$$S_{Gi} = S_i - \sum_{j \in V_i^{(u)}} x_{ji} S_j \quad (24)$$

or $A_u S = S_G$ (25)

Where A_u is the upstream distribution matrix, S is the total vector nodal flow and S_G is the vector of bus generation. Here A_u is

$$[A_U]_{ij} = \begin{cases} 1 & for\ i = j \\ -x_{ji} & for\ j \in v_i^{(u)} \\ 0 & otherwise \end{cases} \quad (26)$$

if A_u^{-1} exist then $S = A_u^{-1} S_G$ (27)

The i^{th} element of S will be

$$S_i = \sum_{m=1}^{n} [A_u^{-1}]_{im} S_{Gm} \qquad for\ i = 1,2,3....n \quad (28)$$

It is worth noting that the 'm^{th}' generator's share in the 'i^{th}' bus power is $S_i = \sum_{m=1}^{n} [A_u^{-1}]_{im} S_{GM}$. Here, the bus power at bus i, $S_i$is also equal to the sum of power flows in the lines leaving the bus i and the load demand S_{Di}. By using Proportional Sharing Principle

$$|(S_{ij})_{ik}| = \frac{|(S_{ij})_{ik}|}{S_i} . S_i = \frac{|(S_{ij})_{ik}|}{S_i} . \sum_{m=1}^{n} [A_u^{-1}]_{im} S_{Gm} \quad (29)$$

or

$$|(S_{ij})_{ik}| = \sum_{n} D_{ik,m}^{G} S_{Gm} \qquad for\ all\ k \in V_i^{(D)} \quad (30)$$

Where $D_{ik,m}^{G} = |(S_{ij})_{ik}| [A_u^{-1}]_{im} / S_i$ and $v_i^{(D)}$ is the set of buses, which are directly supplied from bus i and $D_{ik,m}^{G}$ is topological distribution factor, which is the portion of the generation due to m^{th}generator that flows in the line ik.

The downstream algorithm calculates the share of the individual load in the individual line flow.

when apparent power of bus S_i is expressed as the sum of outflows

$$S_i = \sum_{l \in V_i^{(d)}} |(S_{ij})_{ik}| + S_{Di} = \sum_{l \in V_i^{(d)}} x_{li} S_l + S_{Di} \qquad for\ i = 1,2,3,...n \quad (31)$$

where $V_i^{(d)}$ is the group of buses which directly supplied bus i and $x_{li} = |(S_{ij})_{li}| / S_l$. (32)

equation (23) may be written as

$$S_i - \sum_{l \in V_i^{(d)}} x_{li} S_l = S_{Di} \quad or \quad A_D S = S_D \quad (33)$$

where A_D is the downstream distribution matrix S_D is the vector of bus load demand. the il^{th}element of the matrix A_D is

$$[A_D]_{il} = \begin{cases} 1 & for\ i = j \\ -x_{li} & = -|(S_{ij})_{il}| / S_l \ \ for\ l \in v_i^{(d)} \\ 0 & otherwise \end{cases} \quad (34)$$

If A_D^{-1} exists then $P = A_D^{-1} S_D$, and its i^{th} element is equal to $S_i = \sum_{k=1}^{n} [A_D^{-1}]_{ik} S_{Dk} \qquad for\ i = 1,2,3,....n$ (35)

With that, the inflow to node i from line i-j can be calculated using the proportional sharing principle as

$$|(S_{ij})| = \frac{|(S_{ij})|}{S_i} . S_i = \frac{|(S_{ij})|}{S_i} . \sum_{k=1}^{n} [A_D^{-1}]_{im} S_{Dk} \quad (36)$$

or

$$|(S_{ij})_{ik}| = \sum_{k=1}^{n} D_{ij,k}^{L} S_{Dk} \qquad for\ all\ k \in V_i^{(D)} \quad (37)$$

Where $D_{ij,k}^{L} = |(S_{ij})_{ik}| [A_D^{-1}]_{im} / S_{Dk}$ is the is topological load distribution factor, which gives the portion of the k^{th} load demand that flows in the line ik.

COMPETING INTEREST

The authors declare that they have no competing interests.

REFERENCES

[1] W. J. Lee, C. H. Lin, and L. D. Swift, “Wheeling charge under a deregulated environment,” *IEEE Transactions on Industry Applications*, vol. 37, 2001, no. 1, pp. 178-183.

[2] Gang Duan, Zhao Yang Dong, Wei Bai, Xin Feng Wang, "Power flow based monetary flow method for electricity transmission and wheeling pricing," *Electric Power Systems Research*, vol. 2, 2005, pp. 293-305.

[3] H. H. Happ, “Cost of wheeling methodologies,” *IEEE Transaction on Power Systems*, vol. 9, issue 1, 1994, pp. 147-156.

[4] C. T. Su and J. H. Liaw, “Power wheeling pricing using power tracing and MVA-kM method,” in *Proc. Power Tech., IEEE Porto*, 2001.

[5] N. Kumar, Y. R. V. Reddy, D. Das and N. P. Padhy, "Allocation of transmission charge by using MVA-mile approaches for restructured Indian power utility," *IEEE Power and Energy Society General Meeting*, 2011, pp. 1-6.

[6] Bishwajit Dey, Saurav Raj and Sheila Mahapatra, "A Variegated GWO Algorithm Implementation in emerging power

systems optimization problems", *Research Square*, 2023, pp. 1-45. DOI:https:// doi.org/10.21203/rs.3.rs-1291323/v1

[7] Saurav Raj, Sheila Mahapatra, Rohit Babu and Sumit Verma,"Hybrid intelligence strategy for techno-economic reactive power dispatch approach to ensure system security," Chaos, Solitons & Fractals, 2023, vol. 170, DOI: https://doi.org/ 10.1016/j.chaos.2023.113363

[8] Swetha Shekarappa G., Sheila Mahapatra and Saurav Raj, "Management of Var sources for the reactive power planning problem by oppositional Harris Hawk optimizer,". *Journal of Electrical Systems and Information Technology*, vol. 10, article number 45, 2023, pp. 1-18. DOI: https://doi.org/10.1186/s43067-023-00111-3.

[9] Manjulata Badi and Sheila Mahapatra, " Optimal reactive power management through a hybrid BOA-GWO-PSO algorithm for alleviating congestion," *International Journal of System Assurance Engineering and Management*, *Springer*, 2023, vol. 14, issue 4, pp.1437-1456.

[10] S. Shyamkant Halve, R. Arya and A. Koshti, "Metaphor-less RAO-3 and sine cosine algorithm for optimal sizing of distributed generations of multiple types in radial distribution system," *Electrica*, 2023, vol. 23, Issue 2, pp. 177-191.

[11] J. Bialek, "Allocation of transmission supplementary charge to real and reactive loads," *IEEE Transactions on Power Systems*, 1998, vol. 13, issue 3, pp. 749-754.

[12] Seema N. Pandey, Shashikala Tapaswi, Laxmi Srivastava, "Integrated evolutionary neural network approach with distributed computing for congestion management," *Applied Soft Computing*,2010, vol. 10, issue 1, pp. 251-260.

[13] Seema N. Pandey, Shashikala Tapaswi and Laxmi Srivastava, "Nodal Congestion Price Estimation in spot Power market using artificial neural Network," *IET Generation, Transmission and Distribution*, vol. 2, issue 2, March 2008, pp. 280-290.

[14] S. N. Pandey, S. Tapaswi and L. Srivastava, "Locational Marginal Price Projection using a Novel RBFNN Approach in Spot Power Market," *2008 Joint International Conference on Power System Technology and IEEE Power India Conference*, 2008, pp. 1-7.

[15] Demuth H. and Beale M, Neural Network Toolbox User's Guide, (The mathworks, Inc., Natick, MA), 2002, 7-7.

[16] T. S. Bi, Y. X. Ni, C. M. Shen, F. F. Wu and Q. X. Yang, "A novel radial basis function neural network for fault section estimation in transmission network," *International Conference on Advances in Power SystemControl, Operation and Management, APSCOM-00*, 2000, vol. 1, pp. 259-263.

[17] Mohiuddin Ahmed, Raihan Seraj Syed Mohammed Shamsul Islam, "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation," *Electronics* , 2020, vol. 9, 1295.

[18] Sarika Varshney, Laxmi Srivastava and Manjaree Pandit, "ANN based integrated security assessment of power system using parallel computing," *International Journal of Electrical Power and Energy Systems*, 2012, vol. 9, pp. 42-59.

[19] ParallelComputing Toolbox, Available: <www.mathworks.com/products/parallel computing.

[20] Ravipudi Venkata Rao, "Rao Algorithms: Three Metaphor-less simple algorithms for solving optimization Problems," *International Journal of Industrial Engineering Computations*, 2020, vol. 11, pp. 107-130

[21] Shuvam Sahay, Ramanaiah Upputuri and Niranjan Kumar, "Optimal power flow-based approach for grid dispatch problems through Rao algorithms," *Journal of Engineering Research*, 2023, vol. 11, pp. 1-5.

[22] A. Y Abdelaziz, "Static Security Assessment using Radial Basis Function Neural Networks," *The 10th International Middle East Power System Conference*, Dec. 13-15 2005, Egypt, pp. 535-540.

[23] Demuth H., and Beale M., *Neural Network Toolbox User's Guide*, (The mathworks, Inc., Natick, MA), 2002, pp. 7-7.

[24] M. A. Abido and J. M. Bakhashwain, "Optimal VAR dispatch using a multiobjective evolutionary algorithm," *Electrical Power and Energy Systems*, 2005, vol. 27, pp. 13-20.

[25] 30-Bus System (IEEE Test Case). 2015 [Online] Available: https://www.al-roomi.org/power-flow/30-bus-system.

[26] 146-Bus System (Indian Utility Test Case). 2015 [Online] Available: https://www.al-roomi.org/power-flow/146-bus-system.