A Radial Basis Function Neural Network based Optimal Placement of Distributed generation sources in Distribution Networks

Swati Gupta, Akash Saxena and Bhanu Pratap Soni

Abstract: Distributed Generation (DG) has been envisaged to play an escalating role in electrical power system in near future. However, the benefits of DGs cannot be explored if they are not optimally placed. The efficiency of the distribution system is increased with the employment of DG units. As the employment of the DG comes with the idea of loss minimization and voltage enhancement of the distribution feeders. This paper presents a Radial Basis Function Neural Network (RBFNN) based strategy to penetrate the effective locations and sizing of the DG in a Radial Distribution Network. Training and learning of the Neural Network (NN) has an ample importance while dealing with the real problems, however, the computation time associated with the process put a question mark on the effectiveness of the approach. RBFNN with conventional learning algorithms does not achieve the desired speed and performance in the training process. To overcome this difficulty a heuristic technique Particle Swarm Optimization (PSO) is employed for efficient learning of the RBFNN. The convergence of RBFNN-PSO is compared with Conventional RBFNN. Efficacy of the proposed approach is tested over the standard IEEE-33 bus system.

Keywords: Distribution Networks, Distributed Generation (DG), Neural Networks, Particle Swarm Optimization (PSO), Radial Basis Function Neural Network (RBFNN).

1. INTRODUCTION

In recent years, the modern power system has emerged as a complex network consisting of different electrical drives, loads and moreover the integration of multiple utility devices in both transmission side and generation side. Escalating population and competitive business environment makes the operating conditions stressed. In early years less emphasis was given to improve the efficiency of a distribution networks as the stability and reliability issues were dominant. Recently DGs have attracted a serious attention of electrical market due to their potential use in some issues like voltage deregulation in power system, increasing power consumption and shortage of transmission capacities. But the benefits of the distributed generator are site specific. Hence optimization and sizing of DG is necessary for maximizing the DG potential benefits.

DGs are referred as embedded generations, CIGRE define the DGs as the power plants having maximum capacity of 100 MW. These DGs are connected in the distribution networks and these connections are neither centrally planned nor dispatched [1]. Classification of DGs is done on the basis of the power injections.

Type I: DG capable of injecting real power only, like photovoltaic, fuel cells etc.

Type II: DG capable of injecting reactive power only to improve the voltage profile fall in type-II DG, e.g. kvar compensator, synchronous compensator, capacitors etc.

*Corresponding Author: Akash Saxena, Department of Electrical Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan, Ram nagaria, Jagatpura, Jaipur, Rajasthan, India,
Email: aakash.saxena@hotmail.com
In past various approaches have been applied to penetrate the DG location and sizing in distribution network. These approaches place DG in the system with the aim of minimizing the power losses. Surprisingly all these approaches are addressed with particular loading conditions, however in real time application these load levels are dynamic and may be the approach gives erroneous results when subjected to a particular operating condition. Various approaches like Ant Colony Optimization, Evolutionary Algorithm (EA) and Particle Swarm Optimization (PSO) [2]-[5]. In[4] a new improved harmony search algorithm to address the placement of DGs in distribution networks. The results are compared with Non dominated Sorting Genetic Algorithm (NSGA) and Particle Swarm Optimization (PSO) .In [7] placement of renewable sources namely wind and solar was done with the help of evolutionary programming. Authors reported two operation strategies, namely “turning off wind turbine generator” and “clipping wind turbine generator output”. These strategies were adopted to restrict the wind power dispatch to a specified fraction of system load for system stability consideration. Further the approaches like Prime Dual Interior Point method [8], mixed integer nonlinear programming [9],[10], evolutionary programming (EP) technique [11], analytical approach [12]–[15], trade-off method [16], Hereford Ranch algorithm [17], linear programming technique [18], genetic algorithm(GA) technique [19], heuristic approaches [20], Classical Second Order method [21], Tabu Search approach [22],and Decision Theory approach [23] have been employed to solve the Dg placement problem in Distribution Networks.

2. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization is a stochastic population based evolutionary algorithm for problem solving. Particle swarm optimization is a stochastic, population-based evolutionary computer algorithm for problem solving. It is a kind of swarm intelligence that is based on social-psychological principles and provides insights into social behavior, as well as contributing to engineering applications. The particle swarm optimization algorithm was first described in 1995 by JAMES KENNEDY and RUSSELL C.EBERHART [24]. In a PSO system, particles fly around in a multidimensional search space. During flight, each particle adjusts its position according to its own experience (This value is called Pbest), and according to the experience of a neighboring particle (This value is called Gbest), makes use of the best position encountered by itself and its neighbor fig.1.
Where
$V_i^k$ is current velocity of particle $i$ at iteration $k$.
$V_i^{k+1}$ is modified velocity of particle $i$.
$X_i^k$ is current position of particle $i$ at iteration $k$.
$X_i^{k+1}$ is modified position of particle $i$.
$V_i^{P\text{best}}$ is velocity based on $P\text{best}$.
$V_i^{G\text{best}}$ is velocity based on $G\text{best}$.

This modification can be represented by concept of velocity. Velocity of each agent can be modified by following equation:

$$V_i^{k+1} = w V_i^k + c_1 \text{ rand } *(p_{\text{best},i} - X_i^k) + c_2 \text{ rand } *(g_{\text{best},i} - X_i^k)$$

Using above equation, a certain velocity, which gradually gets close to $p_{\text{best}}$ and $g_{\text{best}}$ can be calculated. The current position (searching point in the solution space) can be modified by following equation:

$$X_i^{k+1} = X_i^k + V_i^{k+1}$$

This is expected to move the swarm toward the best solutions. Such methods are commonly known as heuristics as they make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc.

3. Radial Basis Function Neural Network

The RBFNN is a feed forward neural network consisting of three layers namely, an input layer which feeds the values to each of the neurons in the hidden layer, a hidden function which consists of neurons with radial basis activation function and an output layer which contains neurons with linear activation function. The learning process for RBF neural networks is composed of initiating centers and widths for RBF units and computing weights for connectors of these units. Based on different applications of RBFNN, in the literature many learning strategies have been applied for changing the parameters of RBFNN during the training process. The conventional learning algorithm applied for real time application cannot satisfy the desired speed and performance in the training process. Hence, the optimum steepest descent learning algorithm is applied to improve the RBFNN training process with fewer epochs so as to make it faster and more accurate. A generic topology of RBFNN with $k$ input and $m$ hidden neurons is shown in Fig.2.

For the training of the RBFNN and considering a $k$ dimensional input vector, $X$, the computed scaler values can be expressed as,

$$Y = f(X) = W_0 + \sum_{i=1}^{m} W_i \Phi(D_i)$$

(1)

Where $W_0$ is the bias, $W_i$s is the weight parameter,$m$ is the number of neurons in the hidden layer and $(D_i)$ is the RBF.

There are many basis functional choices possible for the RBF like spline, multi-quadratic, and Gaussian functions but the most widely used one is the Gaussian function. The
Gaussian RBFNN is found not only suitable in generalizing a global mapping but also in refining local features without altering the already learned mapping. In this study, the Gaussian function is used as the RBF and it is given by

\[
\phi(D_i) = \exp \left( -\frac{D_i^2}{\sigma^2} \right)
\]  

(2)

Here \( \sigma \) is the radius of the cluster represented by the center node (Spread) and usually called width, \( D_i \), is the distance between the input vector and \( X \) and all data centers.

The Euclidean norm is normally used to calculate the distance, \( D_i \) which is given by

\[
D_i = \sqrt{\sum_{j=1}^{k} (X_j - f_{ji})^2}
\]

(3)

Where \( f \) is the cluster center for any of the given neurons in the hidden layer.

---

**Fig. 2 : A Generic architecture of RBFNN**
Fig. 3: Flowchart for the proposed methodology
In a RBNN, the estimated output vector \( Y \) can be expressed
\[
Y = [ y_i ] = W \phi^T
\]  
\( i = 1, 2 \ldots n \)

Therefore the error vector, \( E \) and its respective sum squared error, \( J \), which should be minimized through the learning process are defined as
\[
E = Y_d - Y = Y_d - W \phi^T
\]  \( J = \frac{1}{2} EE^T \)

It should be noted that in the conventional steepest descent algorithm, new weights are computed using the gradient of \( J \) in the \( W \) as,
\[
OJ = \frac{\partial J}{\partial W} = \frac{\partial \left( \frac{1}{2} EE^T \right)}{\partial W} = (Y_d - Y) \frac{\partial Y}{\partial W} = E \frac{\partial (W \phi^T)}{\partial W} = E \phi
\]  \( \Delta W = OJ = E \phi \)  \( W_{new} = W_{old} + \lambda \Delta W \)

Where the coefficient \( \lambda \) is called learning rate which remains constant throughout the learning process.

Eq.(7) shows that the optimum direction of the delta weight vector in the sense of first order estimation, does not still specify the optimum length of \( J \) vector and the optimum learning rate(OLR).to achieve the OLR, the sum squared error of the new weights should be obtained using Eqs(4)-(8) as follows:
\[
J(W) + \lambda \Delta W = \frac{1}{2} \left( Y_d - (W + \lambda \Delta W \phi^T) \right) \left( Y_d - (W + \lambda \Delta W \phi^T) \right)^T
\]
\[
= \frac{1}{2} \left( E - \lambda \Delta W \phi^T \right) \left( E - \lambda \Delta W \phi^T \right)^T
\]
\[
= \frac{1}{2} EE^T - \lambda E \phi \Delta W^T + \frac{1}{2} \lambda^2 \Delta W \phi \phi^T \Delta W^T
\]
\[
= A + B \lambda + C \lambda^2
\]  \( A = \frac{1}{2} EE^T, B = -E \phi \Delta W^T \) and \( C = \frac{1}{2} \Delta W \phi \phi^T \Delta W^T \) are scaler constants.

Thus, \( J(W + \lambda \Delta W) \) is a quadratic function of \( \lambda \) with constant coefficients A, B and
C. Therefore $J(\lambda)$ defines a quadratic function of $\Phi$ with positive coefficients of the second order term. $J(\lambda)$ can be minimized by taking its derivative as,

$$\frac{\partial J}{\partial \lambda} = \frac{\partial (A + B\lambda + C\lambda^2)}{\partial \lambda} = B + 2\lambda C = 0$$

(10)

Hence,

$$\lambda_{\text{min}} = -\frac{B}{2C} = \frac{(E\Phi)(E\Phi)^T}{(E\Phi\Phi^T)(E\Phi\Phi^T)^T}$$

(11)

This learning rate minimizes the $J(\lambda)$, and so OLR can be expressed as,

$$\hat{\lambda}_{\text{opt}} = \frac{(E\Phi)(E\Phi)^T}{(E\Phi\Phi^T)(E\Phi\Phi^T)^T} \geq 0$$

(12)

Using the above equation, the optimum delta weight vector can be determined as,

$$\Delta W_{\text{opt}} = \hat{\lambda}_{\text{opt}} \Delta W = \frac{(E\Phi)(E\Phi)^T E\Phi}{(E\Phi\Phi^T)(E\Phi\Phi^T)^T}$$

(13)

Hence,

$$\Delta W_{\text{opt}} = \hat{\lambda}_{\text{opt}} \Delta W = \frac{(E\Phi)(E\Phi)^T E\Phi}{(E\Phi\Phi^T)(E\Phi\Phi^T)^T}$$

(14)

For which the initial value for $W$ is set with a random value.

4. Numerical Results

In this paper PSO serves two purposes at first PSO calculates the size and optimal location of DG and at second the training of RBFNN is carried out by the PSO. The convergence characteristics of RBFNN (Conventional and PSO) are compared in fig. 6. Further Fig. 4 & 5 shows the value of losses and sizing of the DGs at each bus for a given operating condition, due to space limitations we are showing the results of Base case only. The proposed method is implemented using MATLAB 2013 and run on a Pentium IV CPU, 2.69 GHz, and 1.84 GB RAM computer [26].

For showing the effectiveness of proposed approach the DG size and locations are calculated by RBFNN and RBFNN-PSO, the results for the same under four operating conditions has been shown in the table I. RBFNN is trained through 110 datasets and target for the simulations are achieved through PSO algorithm. Out of these datasets 80% are used for training and rest is used for testing purpose. A convergence characteristic is shown in fig.6 Mean Square Error (MSE) is decreased with RBFNN-PSO.
Fig. 4: Size of DG at each bus of 33 bus system for base case

Fig. 5: Active power loss at each bus of 33 bus system for base case

Fig. 6: Comparison of Convergence between two methods
Table: Calculation of DG sizing and Power loss

<table>
<thead>
<tr>
<th>Bus no</th>
<th>Conventional RBFNN size</th>
<th>loss</th>
<th>PSO-RBFNN size</th>
<th>loss</th>
<th>Actual size</th>
<th>loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4.17861</td>
<td>0.203243</td>
<td>4.09476</td>
<td>0.19577</td>
<td>4.193082</td>
<td>0.205534</td>
</tr>
<tr>
<td>3</td>
<td>3.76876</td>
<td>0.178767</td>
<td>3.678971</td>
<td>0.15676</td>
<td>3.688829</td>
<td>0.164051</td>
</tr>
<tr>
<td>4</td>
<td>3.23452</td>
<td>0.176121</td>
<td>3.209813</td>
<td>0.14238</td>
<td>3.213348</td>
<td>0.151148</td>
</tr>
<tr>
<td>5</td>
<td>2.84672</td>
<td>0.130789</td>
<td>2.7486</td>
<td>0.136521</td>
<td>2.940986</td>
<td>0.138808</td>
</tr>
<tr>
<td>6</td>
<td>2.57861</td>
<td>0.109327</td>
<td>2.53812</td>
<td>0.108994</td>
<td>2.618548</td>
<td>0.113442</td>
</tr>
<tr>
<td>7</td>
<td>2.462574</td>
<td>0.109844</td>
<td>2.45743</td>
<td>0.109665</td>
<td>2.482574</td>
<td>0.114444</td>
</tr>
<tr>
<td>8</td>
<td>1.793295</td>
<td>0.119252</td>
<td>1.80787</td>
<td>0.118725</td>
<td>1.810295</td>
<td>0.120725</td>
</tr>
<tr>
<td>9</td>
<td>1.569087</td>
<td>0.119487</td>
<td>1.55998</td>
<td>0.119847</td>
<td>1.580588</td>
<td>0.124069</td>
</tr>
<tr>
<td>10</td>
<td>1.448975</td>
<td>0.119457</td>
<td>1.40489</td>
<td>0.109637</td>
<td>1.414734</td>
<td>0.126453</td>
</tr>
<tr>
<td>11</td>
<td>1.365416</td>
<td>0.119436</td>
<td>1.34767</td>
<td>0.119730</td>
<td>1.388376</td>
<td>0.126934</td>
</tr>
<tr>
<td>12</td>
<td>1.35967</td>
<td>0.119805</td>
<td>1.34008</td>
<td>0.118572</td>
<td>1.340396</td>
<td>0.128015</td>
</tr>
<tr>
<td>13</td>
<td>1.00942</td>
<td>0.127591</td>
<td>1.00901</td>
<td>0.127592</td>
<td>1.009708</td>
<td>0.131759</td>
</tr>
<tr>
<td>14</td>
<td>1.02687</td>
<td>0.128722</td>
<td>1.026019</td>
<td>0.128242</td>
<td>1.027199</td>
<td>0.132426</td>
</tr>
<tr>
<td>15</td>
<td>1.099135</td>
<td>0.13275</td>
<td>1.099594</td>
<td>0.134644</td>
<td>1.013569</td>
<td>0.134679</td>
</tr>
<tr>
<td>16</td>
<td>0.943602</td>
<td>0.136542</td>
<td>0.937872</td>
<td>0.129898</td>
<td>0.973602</td>
<td>0.13796</td>
</tr>
<tr>
<td>17</td>
<td>0.864398</td>
<td>0.139281</td>
<td>0.84371</td>
<td>0.13912</td>
<td>0.877957</td>
<td>0.143721</td>
</tr>
<tr>
<td>18</td>
<td>0.829365</td>
<td>0.143940</td>
<td>0.81787</td>
<td>0.13939</td>
<td>0.831601</td>
<td>0.146943</td>
</tr>
<tr>
<td>19</td>
<td>1.0156425</td>
<td>0.209945</td>
<td>1.014676</td>
<td>0.209425</td>
<td>1.016266</td>
<td>0.210014</td>
</tr>
<tr>
<td>20</td>
<td>0.477945</td>
<td>0.209191</td>
<td>0.465379</td>
<td>0.210387</td>
<td>0.485379</td>
<td>0.211395</td>
</tr>
<tr>
<td>21</td>
<td>0.404389</td>
<td>0.209572</td>
<td>0.402769</td>
<td>0.2090115</td>
<td>0.403369</td>
<td>0.211508</td>
</tr>
<tr>
<td>22</td>
<td>0.337812</td>
<td>0.2107875</td>
<td>0.326521</td>
<td>0.2109612</td>
<td>0.343381</td>
<td>0.211845</td>
</tr>
<tr>
<td>23</td>
<td>2.423789</td>
<td>0.169360</td>
<td>2.490954</td>
<td>0.1693577</td>
<td>2.501834</td>
<td>0.173602</td>
</tr>
<tr>
<td>24</td>
<td>1.721787</td>
<td>0.176505</td>
<td>1.7281321</td>
<td>0.1717705</td>
<td>1.732086</td>
<td>0.177705</td>
</tr>
<tr>
<td>25</td>
<td>1.078786</td>
<td>0.181217</td>
<td>1.077711</td>
<td>0.1791322</td>
<td>1.07996</td>
<td>0.182927</td>
</tr>
<tr>
<td>26</td>
<td>2.456676</td>
<td>0.114914</td>
<td>2.461321</td>
<td>0.114910</td>
<td>2.477206</td>
<td>0.1154</td>
</tr>
<tr>
<td>27</td>
<td>2.278908</td>
<td>0.116591</td>
<td>2.276812</td>
<td>0.1152617</td>
<td>2.309008</td>
<td>0.117869</td>
</tr>
<tr>
<td>28</td>
<td>1.865451</td>
<td>0.12187</td>
<td>1.850091</td>
<td>0.1201211</td>
<td>1.875148</td>
<td>0.123658</td>
</tr>
<tr>
<td>29</td>
<td>1.645429</td>
<td>0.124327</td>
<td>1.6312421</td>
<td>0.1232321</td>
<td>1.66622</td>
<td>0.125988</td>
</tr>
<tr>
<td>30</td>
<td>1.545421</td>
<td>0.1265445</td>
<td>1.5312176</td>
<td>0.125411</td>
<td>1.559048</td>
<td>0.127926</td>
</tr>
<tr>
<td>31</td>
<td>1.345132</td>
<td>0.1329987</td>
<td>1.321212</td>
<td>0.1321132</td>
<td>1.36753</td>
<td>0.134132</td>
</tr>
<tr>
<td>32</td>
<td>1.310998</td>
<td>0.1357551</td>
<td>1.310787</td>
<td>0.1343236</td>
<td>1.311423</td>
<td>0.136555</td>
</tr>
<tr>
<td>33</td>
<td>1.039898</td>
<td>0.140121</td>
<td>1.038788</td>
<td>0.140031</td>
<td>1.040618</td>
<td>0.140064</td>
</tr>
</tbody>
</table>
Fig. 7: Comparative analysis of DG size at each bus with different algorithms

Fig. 8: Comparative analysis of losses at each bus with different approaches

Fig. 7 and 8 shows the comparative analysis of losses and DG size is calculated with RBFNN and RBFNN-PSO it is observed that the values obtained from RBFNN-PSO is much less than the RBFNN conventional and it is more close with the actual values.

5. Conclusion

Penetration of DG in distribution network is a classical problem in this practical and complex power system. This paper proposes a new technique based on RBFNN. The Distribution Network, IEEE-33 bus test system is chosen to exhibit the proposed approach. Convergence characteristic depicting the efficiency of both methods are shown and it is observed that MSE is achieved by RBFNN-PSO. Better convergence and Tabulated results
shows that results obtained from RBFNN-PSO achieved 10e-5 MSE. Effective placements of DGs are carried out by the proposed approach.

References