Opposition based red wolf algorithm for solving optimal reactive power problem

Lenin Kanagasabai
Department of Electrical and Electronic Engineering, Prasad V. Potluri Siddhartha Institute of Technology, India

Article Info
Article history:
Received Feb 12, 2021
Revised Apr 2, 2021
Accepted Apr 24, 2021

Keywords:
Optimal reactive power
Transmission loss
Red wolf optimization
Opposition based learning

ABSTRACT
This paper projects opposition based red wolf optimization (ORWO) algorithm for solving optimal reactive power problem. Each Red wolf has a flag vector in the algorithm, and length is equivalent to the whole sum of numbers which features in the dataset of the wolf optimization (WO). In this projected algorithm Red wolf optimization algorithm has been intermingled with opposition based learning (OBL). By this amalgamate procedure the convergence speed of the projected algorithm will be increased. To discover an improved candidate solution, the concurrent consideration of a probable and its corresponding opposite are estimated which is closer to the global optimum than an arbitrary candidate solution. Proposed algorithm has been tested in standard IEEE 14,300 bus test system and simulation results show the proposed algorithm reduced the real power loss considerably.

This is an open access article under the CC BY-SA license.

1. INTRODUCTION

In this work the key objective is Actual power loss reduction. Optimal reactive power problem has been solved by a variety of methods [1]-[6]. However many technical hitches are found while solving problem due to an assortment of constraints. Evolutionary techniques [7]-[16] are applied to solve the reactive power problem, but the key problem is some algorithms stuck in local optimal solution & failed to balance the exploration & exploitation during the search of global solution. This paper projects opposition based red wolf optimization (ORWO) algorithm for solving optimal reactive power problem. Each red wolf has a flag vector, in the algorithm, and length is equivalent to the whole sum of numbers which features in the dataset of the wolf optimization (WO) [17]-[19]. In this projected algorithm red wolf optimization algorithm has been intermingled with opposition based learning (OBL) [20]. By this convergence speed will be increased. To discover an improved candidate solution, the concurrent consideration of a probable and its corresponding opposite are estimated which is closer to the global optimum than an arbitrary candidate solution. Proposed algorithm has been tested in standard IEEE 14,300 bus test system and simulation results show the proposed algorithm reduced the real power loss considerably.

2. PROBLEM FORMULATION

Objective of the problem is to reduce the true power loss:

\[
F = P_L = \sum_{k \in \text{Nbr}} g_k \left( V_i^2 + V_j^2 - 2V_iV_j \cos \theta_{ij} \right)
\]  

Journal homepage: http://ijaas.iaescore.com
Voltage deviation given as (2).
\[ F = P_L + \omega_c \times \text{Voltage Deviation} \]  
(2)

Voltage deviation given by (3).
\[ \text{Voltage Deviation} = \sum_{i=1}^{Np} |V_i - 1| \]  
(3)

Constraint (Equality)
\[ P_G = P_D + P_L \]  
(4)

Constraints (Inequality)
\[ p_{\text{gslack}}^{\text{min}} \leq p_{\text{gslack}} \leq p_{\text{gslack}}^{\text{max}} \]  
(5)
\[ Q_{\text{gi}}^{\text{min}} \leq Q_{\text{gi}} \leq Q_{\text{gi}}^{\text{max}}, i \in N_g \]  
(6)
\[ V_i^{\text{min}} \leq V_i \leq V_i^{\text{max}}, i \in N \]  
(7)
\[ T_i^{\text{min}} \leq T_i \leq T_i^{\text{max}}, i \in N_T \]  
(8)
\[ Q_c^{\text{min}} \leq Q_c \leq Q_c^{\text{max}}, i \in N_C \]  
(9)

3. OPPOSITION BASED RED WOLF OPTIMIZATION

Red wolf optimization mimics the communal management and hunt deeds of red wolves in nature. There are three fittest candidate solutions assumed as \( \alpha, \beta \) and \( \gamma \) to lead the population toward promising regions of the exploration space in each iteration of red wolf optimization. \( \varphi \) is named for the rest of red wolves and it will assist \( \alpha, \beta \) and \( \gamma \) to encircle, hunt, and attack prey, that is, to find enhanced solutions. In order to scientifically replicate the encompassing behavior of red wolves, equation (10) is proposed.

\[ \vec{G} = |\vec{F}, \vec{Y}_\varphi(t) - \vec{Y}(t)|, \vec{Y}(t + 1) = \vec{Y}_\varphi(t) - \vec{H}, \vec{G} \]  
(10)

Where \( t \) indicates the current iteration, \( \vec{H} = 2\vec{b}_1 - \vec{b}_1 - \vec{b}_2 \), \( \vec{F} = 2 \vec{r}_2 \), \( \vec{Y}_\varphi \) the position vector of the prey, \( \vec{Y} \) is the position vector of a red wolf, \( \vec{b}_1 \) is linearly decreased from 2.0 to 0, and \( \vec{r}_1 \) and \( \vec{r}_2 \) are arbitrary vectors in [0, 1].

In order to mathematically simulate the hunting behavior of red wolves, equations (11), (12), (13) are proposed,

\[ \vec{G}_\alpha = |\vec{F}, \vec{Y}_\alpha - \vec{Y}| \]  
(11)
\[ \vec{G}_\beta = |\vec{F}, \vec{Y}_\beta - \vec{Y}| \]  
(12)
\[ \vec{G}_\gamma = |\vec{F}, \vec{Y}_\gamma - \vec{Y}| \]  
(13)

\[ \vec{Y}(t + 1) = \frac{\vec{Y}_\alpha + \vec{Y}_\beta + \vec{Y}_\gamma}{3} \]  
(14)

The position of a red wolf was updated by (13) & (14) is used to discrete the position.

\[ flag_{i,j} = \begin{cases} 1 & |Y_{i,j} | > 0.50 \\ 0 & \text{otherwise} \end{cases} \]  
(14)
Where $i$, indicates the $j$th position of the $i$th Red wolf, $f_{lag_{i,j}}$ is features of the wolf.

Opposition based learning (OBL) is one of the influential optimization tools to boost the convergence speed of different optimization techniques. The thriving implementation of the OBL engages evaluation of opposite population and existing population in the similar generation to discover the superior candidate solution of a given reactive power problem. The conception of opposite number requirements is to be defined to explain OBL.

Let $N$ ($N \in [x,y]$) be a real number and the $N^o$ (opposite number) can be defined as (15).

$$N^o = x + y - N$$

In the exploration space it has been extended as (16).

$$N^o_i = x_i + y_i - N_i$$

Where $(N_1, N_2, \ldots N_d)$ is a point in the dimensional search space, $N_i \in [x_i, y_i], i \rightarrow \{1,2,3,\ldots d\}$

In all oppositional based optimization, the conception of OBL is used in the initialization procedure and as well as in each iteration using the generation jumping rate, $J_{r}$.

Step 1: The positions of Red wolves (search agents) are initialized arbitrarily in the exploration space. Population size (number of wolves) and number of iterations are fixed.

Step 2: Compute fitness value of each search agent which represents the distance of wolf from the prey.

Step 3: Initialize opposite points and use them to generate opposite population and compute the fitness of each individual populations.

Step 4: Perform sorting among existing population (pop) and opposite population (opop), based on their fitness values.

Step 5: Select $n_{P}$ number of fittest solutions from the combination of the existing and the equivalent opposite population.

Step 6: Based on the fitness values, solutions are identified which represent the alpha, beta and delta category wolves, respectively.

Step 7: Modify the positions of the Red wolves

Step 8: Update the fitness value using the modified position of the Red wolves.

Step 9: Using jumping rate, the opposite population are generated from the current population.

Step 10: Select $n_{P}$ number of fittest solution from the combination of the current and the opposite population.

Step 11: Repeat step 6 to step 10 until the maximum number of iteration is reached.

Step 12: Output the best solution.

Commence

**Initialize the parameters**

Initialize $b$, $H$, and $F$; beginning positions of Red wolves has been stimulated.

**Generate opposite population; for j=; population size for i=1; number of control variables**

$$N^o_i = x_i + y_i - N_i$$

**Classify the existing population and the opposite population from best to worst**

Depending upon the population size $n_{P}$, choose $n_{P}$ number of fittest solutions from the existing and the opposite population

**Modernize the control variables**

By utilizing the jumping rate, opposite population are generated from the existing population:

End if

End for

**Work out the maximum fitness of Red wolves as follows**,

Primary maximum fitness of the Red wolf is designated as “$\alpha$”

Second maximum fitness of the Red wolf is designated as “$\beta$”

Third maximum fitness of the Red wolf is designated as “$\gamma$”

While $k < \text{maximum iteration}$

For $i=1$; population size

For $j=1$; number of variables

If $\text{jumping rate} > \text{random}$

$$\text{opop}(i,j) = \min(j) + \max(j) - \text{pop}(i,j)$$

*Opposition based red wolf algorithm for solving optimal reactive power problem (Lenin Kanagasabai)*
Else
    \[ opop(i,j) = pop(i,j) \]
End if
End for

Intermittently modify the values of \( b, \overrightarrow{H} \) and \( \overrightarrow{F} \);

Fitness of Red wolves has been calculated
The assessment of red wolves "\( \alpha \), \( \beta \)" and "\( \gamma \)" has to be revised
\[ k = k + 1 \]
End while
Revise the value of "\( \alpha \)" as the optimal characteristic division;
End

4. SIMULATION RESULTS

At first in standard IEEE 14 bus system the validity of the proposed opposition based red wolf optimization (ORWO) algorithm has been tested & comparison results are presented in Table 1. Figure 1 Provide the details of Comparison of real power loss.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>1.06</td>
<td>1.05</td>
<td>1.02</td>
</tr>
<tr>
<td>V2</td>
<td>1.03</td>
<td>1.05</td>
<td>1.01</td>
</tr>
<tr>
<td>V3</td>
<td>0.98</td>
<td>1.03</td>
<td>1.00</td>
</tr>
<tr>
<td>V6</td>
<td>1.05</td>
<td>1.05</td>
<td>1.00</td>
</tr>
<tr>
<td>V8</td>
<td>1.00</td>
<td>1.04</td>
<td>0.90</td>
</tr>
<tr>
<td>Q9</td>
<td>0.139</td>
<td>0.132</td>
<td>0.100</td>
</tr>
<tr>
<td>T56</td>
<td>0.979</td>
<td>0.960</td>
<td>0.900</td>
</tr>
<tr>
<td>T47</td>
<td>0.950</td>
<td>0.950</td>
<td>0.900</td>
</tr>
<tr>
<td>T49</td>
<td>1.014</td>
<td>1.007</td>
<td>1.000</td>
</tr>
<tr>
<td>Ploss (MW)</td>
<td>5.92892</td>
<td>5.50031</td>
<td>4.54826</td>
</tr>
</tbody>
</table>

Figure 1. Comparison of real power loss

Then IEEE 300 bus system [22] is used as test system to validate the performance of the Opposition based Red Wolf Optimization (ORWO) algorithm. Table 2 shows the comparison of real power loss obtained after optimization. Figure 2 gives the comparison of real power values. Real power loss has been considerably reduced when compared to the other standard reported algorithms.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PLOSS (MW)</td>
<td>646.2998</td>
<td>650.6027</td>
<td>635.8942</td>
<td>614.2846</td>
</tr>
</tbody>
</table>

Table 2. Comparison of real power loss
5. CONCLUSION

Opposition based red wolf optimization (ORWO) algorithm has successfully solved the reactive power optimization problem. In this projected algorithm red wolf optimization algorithm has been intermingled with opposition based learning (OBL). By this amalgamate procedure the convergence speed of the projected algorithm has been increased. Proposed algorithm has been tested in standard IEEE 14,300 bus test system and simulation results show the proposed opposition based red wolf optimization (ORWO) algorithm reduced the real power loss considerably.

REFERENCES


