Research of Distribution Network Reactive Power Optimization Based on Improved Cloud Particle Swarm Optimization BP Neural Network

Hong-Sheng Su and Pei-Jiong Zhang

Abstracts—In view of cloud particle swarm optimization is easily trapped in local minimum and existed slowly convergence in the distribution network reactive power optimization, the cloud particle swarm optimization is improved based on cloud digital features (Ex, En, He), local search and global search which are based on the solution space transform are combined, the random particle is crossed by cloud generator of X condition, the cross operation of particle according to cloud genetic, variation operation of the particle according to the normal cloud particle, apparent improvement are got in time and storage by using the algorithm, in order to enhance convergence accelerate, train BP neural network by improved cloud particle swarm, the simulation experiment are done in IEEE30-bus system and YUMEN power grid in northwest china, simulation results show that a better global solution can be attained by using the improved algorithm, convergence speed and accuracy are accelerated and improved.

Index Terms—Distribution network, reactive power optimization, cloud particle swarm algorithm, cloud model, BP neural network.

I. INTRODUCTION

Reactive power optimization is a constraint, large-scale and nonlinear combinatorial optimization problem and is a reactive regulatory method which meets all the constraint condition and obtains one or more optimization performance index of the system through the optimization of some control variables, when the structure parameters of the power system and the load are given [1]-[3]. Conventional optimization methods are mainly composed of classical algorithm and artificial intelligence methods.

Bundled with discrete variables to place and multiple maximum, conventional algorithm for reactive power optimization is hard to get good results because of great calculating error, curse of dimensionality, etc [4]-[6]. Some artificial intelligence methods such as simulated annealing, genetic algorithm, immune algorithm, ant colony algorithm, particle swarm optimization and neural network have been used to solve reactive power optimization problem [7]-[9]. These algorithms based on warm intelligent optimization technologies have global searching ability and process discrete and multi-objective optimization problems. However, simple algorithm can only be lead to a local minimum and slow convergence.

Dynamic adaptive differential evolution algorithm proposed in [10] and improved particle swarm optimization algorithm proposed in [11], which reactive power optimization by different methods of distribution network, but the selection of weight and convergence precision were not accuracy. Conventional adaptive particle swarm optimization use the strategies which can't reflect the process of actual optimization search of linear inertia weight reduction. Cloud particle swarm based on cloud digital features (Ex, En, He) has stability and randomness [12], [13]. Cloud generator of X condition can adjust nonlinear dynamically inertia weight of particles [14], [15], but it is trapped in a local minimum and less convergence precision in reactive power optimization. In this paper, BP neural network is trained by CPSO improved algorithm. The results indicate that the convergence speed and accuracy of reactive power optimization for distribution network were well solved with CPSO.

II. REACTIVE POWER OPTIMIZATION MODEL OF DISTRIBUTION NETWORK

A. Objective Function

Under the condition of guaranteeing system reactive power balance, generator bus voltages, on-load regulating transformer’s transformation ratio, compensation capacitor as control method, in order to reduce the system network loss, improve the quality of power for the target. The paper is from aspect of economic, which is active network loss minimum for target, objective function [16]:

\[
\min F = \sum_{i=1}^{L} P_{sw} \alpha \sum_{i=1}^{M} \left( \frac{\Delta U}{U_{sw} - U_{sw}} \right)^2 + \lambda_2 \sum_{i=1}^{N} \left( \frac{\Delta Q_{Gi}}{Q_{Gi, max} - Q_{Gi, min}} \right)^2
\]

\[
U_i = \begin{cases} 
U_{min} - U_i, & U_i < U_{min} \\
0, & U_{min} \leq U_i \leq U_{max} \\
U_i - U_{max}, & U_{max} < U_i 
\end{cases}
\]

\[
\Delta Q_{Gi} = \begin{cases} 
Q_{Gi, min} - Q_{Gi}, & Q_{Gi} < Q_{Gi, min} \\
0, & Q_{Gi, min} \leq Q_{Gi} \leq Q_{Gi, max} \\
Q_{Gi, max} - Q_{Gi}, & Q_{Gi} > Q_{Gi, max} 
\end{cases}
\]

Manuscript received October 15, 2012; revised November 19, 2012.
Hong-Sheng Su was with Lanzhou Jiaotong University in 2001. (e-mail: shsen@163.com)
Pei-Jiong Zhang was with Physical and electronic engineering institute of Long Dong College. In the postgraduate period, (e-mail:364182509@qq.com)
B. Variable Constraints

\[
\begin{align*}
Q_{\text{min}} \leq Q_i \leq Q_{\text{max}} & \quad i = 1, 2, \ldots, n_g \\
U_{\text{min}} \leq U_i \leq U_{\text{max}} & \quad i = 1, 2, \ldots, n_c \\
Q_{\text{min}} \leq Q_i \leq Q_{\text{max}} & \quad i = 1, 2, \ldots, n_c \\
T_{\text{min}} \leq T_i \leq T_{\text{max}} & \quad i = 1, 2, \ldots, n_c \\
U_{\text{min}} \leq U_i \leq U_{\text{max}} & \quad i = 1, 2, \ldots, n_c
\end{align*}
\]

(3)

C. Power Constraints

\[
\begin{align*}
P_G - P_L = & \sum_{j=1}^{m} U_j (G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}) = 0 \\
Q_G + Q_L = & \sum_{j=1}^{m} U_j (G_{ij} \sin \delta_{ij} + B_{ij} \cos \delta_{ij}) = 0
\end{align*}
\]

(4)

where,

\begin{itemize}
  \item \( N, M, L \): Generator nodes total number
  \item \( P_{\text{loss}} \): active loss of the power system
  \item \( U_o, U_{\text{max}}, U_{\text{min}} \): Node voltage, Voltage limit
  \item \( Q_{\text{G}}, Q_{\text{L}} \): Generator reactive power, reactive power limit
  \item \( \lambda_o, \lambda_c \): Cross-border penalty coefficients
  \item \( Q_{\text{C}} \): Capacitor compensation capacity
  \item \( U_{\text{G}} \): Generator voltage
  \item \( T_{\text{i}} \): Adjustable transformer
  \item \( P_{\text{G}} \): Generator active power
  \item \( G_{ij} \): mutual conductance between bus i and j
  \item \( B_{ij} \): mutual susceptance between bus i and j
  \item \( \delta_{ij} \): Phase difference
  \item \( n_{\text{PQ}} \): PQ nodes number
  \item \( P_{L, Q_{\text{L}}} \): Lode nodes active power and reactive power
\end{itemize}

III. IMPROVED CLOUD PARTICLE SWARM

In view of cloud digital features (Ex, En, He) code of CPSO algorithm existing problems, do the following improvement:

With the help of population alternative solution space transform, the global search and local search combined

Basic CPSO algorithm is most of time running update on population, in addition in the later evolution, often appear evolution slower limitations, so introduces population alternative solution space change.

The particle swarm optimization algorithm of population alternative is mainly through a lot of a particle swarm, the solution space use different search way, in this several particle swarm, main search group is one of the particle swarm, auxiliary search group is major in the search process, in order to make particle diversity of main search group get maintain, so the part of auxiliary search group of particles and the main search group of particles alternative, to avoid the lack of diversity, and into the early maturity, ensure that main search group can search to global advantages.

In CPSO, the particles traverse space is [-1,1] of every dimension, for calculating superiority-inferiority of the cloud particles current position, need the solution space transform, from unit space \( I = [-1,1]^n \) mapping to optimization solution space, the i cloud operator of \( \Phi_j \) is \([\alpha', \beta']^T\), so corresponding solution space variables:

\[
X_{\theta_i}^j = \frac{1}{2} [b_i (1 + \alpha') + a_i (1 - \alpha')]
\]

(5)

Then optimize the solution space, If the optimal value is better than optimal solution, so replace it.

According to cloud generator of cloud model X conditions random generation particles, using the cloud genetic cross and normal clouds variation operation

Crossover probability algorithm:

\[
\begin{align*}
Ex = & \bar{f} = \frac{F_f}{F_f + F_m} x_f + \frac{F_m}{F_f + F_m} x_m \\
En = & (f_{\text{max}} - \bar{f}) / c_1 \\
He = & En / c_2 \\
En = & \text{RANDN(En,He)} \\
p_c = & \begin{cases} 
\bar{k}_1 \sqrt{2(En)^2} & f \geq \bar{f} \\
\bar{k}_3 & f < \bar{f}
\end{cases}
\end{align*}
\]

where,

\begin{itemize}
  \item \( x_f, x_m \): father individual, mother individual
  \item \( F_f, F_m \): father individual fitness, mother individual fitness
  \item \( c_1, c_2 \): Control variables
  \item \( \bar{f} \): average fitness
\end{itemize}

Definition 1: Prior given threshold N and K, when global extremum continuous N generation didn’t happen or evolution process evolution by less than K, think particles into the local extremum, of all the particles according to global extremum, through the normal clouds generator on variation operation[12].

Definition 2: One dimensional normal clouds operator is \( \Phi_{\text{Forward}}^t (\text{Ex,En,He}),\) the global feature of qualitative concept change to mapping \((\pi : C \rightarrow [0,1])\) of quantitative said[12]. Meet the following conditions:

\[
\begin{align*}
\Theta = \{ t|\text{Norm(En,He)}, \ t \in 1..N \}; \\
X = \{ x|\text{Norm(Ex, t)}, \ t \in \Theta, t = 1..N \}; \\
\Pi = \{ (x,y) | x \in X, t \in \Theta, y = \exp((x-Ex)^2/(2\sigma^2)) \}
\end{align*}
\]

In the process of CPSO evolution, the more evolutionary deviates from the optimal solution, improvement measures are as follows:

Population initialization, record the current position and velocity of the each particle initial value, and then calculating the fitness of each particle, judge whether to variation threshold N, If meet the requirements, according definition 1 to each particle on variation operation; Otherwise according to type (6), and (7) particles to update operation. After the end of each generation, take optimal from the optimal solution of the three, as the algorithm of
IV. CLOUD PARTICLE SWARM OPTIMIZATION BP NEURAL NETWORK

Compared with basic PSO, CPSO have a good (Ex, En, He) to represent the characteristics of the cloud model, figure1 shows Ex=20; En=3; He=0.1.the population is divided into three subgroup, use X conditions cloud generator nonlinear dynamic adjustment inertia weight of the particle, thus can be a very good selection weights. The improved algorithm, Ex, En, He, K and N have important influence, at the same time, the inertia weight \( \omega \), accelerated factor \( C_2 \), \( C_1 \) have also very important influence. Obviously, the improvement measure enhances the diversity, improves the ability to search algorithm, and reflect the normal cloud variation operator.

The structure is 3-5-2, the three layers of BP network, from input layer to the hidden layer excitation function:

\[
y_j = f'(net_j) = f[\sum_{i=0}^n v_{ij} \cdot x_i]
\]

From hidden layer to the output layer excitation function:

\[
o_k = f'(net_k) = f[\sum_{j=0}^m w_{jk} \cdot y_j]
\]

Definite output error of the actual and expect:

\[
E = \frac{1}{2} \cdot (D - O)^2 = \frac{1}{2} \sum_{i=1}^l (d_i - o_i)^2
\]

The function relation of E and weights:

\[
E = \frac{1}{2} \sum_{k=1}^l \sum_{j=0}^m (d_i - f(\sum_{j=0}^n v_{ij} \cdot x_i))^2
\]

From type (11), adjustment weight can effect the size of the error E, to satisfy adjustment weight, continually reduce E, the weight adjustment quantity and error E should be proportional to the gradient descent. a gradient descent constant is \( \eta \in (0,1) \), namely represents learning rate of the algorithm.

From hidden layer to output layer weight \( \Delta w_{jk} \) as follows:

\[
\Delta w_{jk} = -\eta \cdot \frac{\partial E}{\partial w_{jk}} = -\eta \cdot \frac{\partial E}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{jk}} = -\eta \cdot \frac{\partial E}{\partial net_k} \cdot y_j
\]

Error signal of from hidden layer to output layer:

\[
\sigma_k = -\frac{\partial E}{\partial net_k} = -\frac{\partial E}{\partial o_k} \cdot \frac{\partial o_k}{\partial net_k} = (d_i - o_i) \cdot f'(net_i)
\]

Get the weight adjustment of hidden layer to the output layer:

\[
\Delta w_{jk} = \eta \cdot \sigma_k \cdot y_j = \eta \cdot (d_i - o_i) \cdot f'(net_i)
\]

From input layer to the hidden layer weights \( \Delta v_{ij} \) as follows:

\[
\Delta v_{ij} = -\eta \cdot \frac{\partial E}{\partial v_{ij}} = -\eta \cdot \frac{\partial E}{\partial net_j} \cdot \frac{\partial net_j}{\partial v_{ij}} = -\eta \cdot \frac{\partial E}{\partial net_j} \cdot x_i
\]
Error signal of from input layer to the hidden layer:
\[
\delta_j = -\frac{\partial E}{\partial \hat{y}_j} = -\frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial \hat{y}_j} = -\frac{\partial E}{\partial \hat{y}_j} f_j'(net_j) \\
= -\left(\frac{\partial E}{\partial \hat{y}_1} f_1'(net_1) + \frac{\partial E}{\partial \hat{y}_2} f_2'(net_2) + \cdots + \frac{\partial E}{\partial \hat{y}_k} f_k'(net_k)\right) \\
= \left(\sigma_{wi_j} + \sigma_{wi_j} + \cdots + \sigma_{wi_j}\right) f_j'(net_j) \\
= \left(\sum_{i=1}^{j} (\sigma_{wi_j})\right) f_j'(net_j) \\
= \left[w_{f_{net}} x \right] f_j'(net_j) \\
\]
Get the weight adjustment of input layer to the hidden layer:
\[
\Delta w_{ij} = \eta \cdot \delta_j \cdot x_i = \eta \cdot \left(\sum_{j=1}^{k} (\sigma_{wi_j})\right) f_j'(net_j) \\
(17)
\]
Excitation function is continuous differentiable single polarity sigmoid function:
\[
f(x) = \frac{1}{1 + e^{-x}} \\
(18)
\]
According to the above of the layers of weights fixed, use all the sample to iterative calculation, the final weights is set to a value which make the system error converge to precision scope or maximum iterating times, complete the training, output network parameters.

V. SIMULATION RESULT AND ANALYSIS

Taking the IEEE30 bus system as illustration (Fig. 3). IEEE30 node system is 41 line, 22 load node and 6 generators node. the node 1, 2, 5, 8, 11, 13 is generators node, and 1 is balance node, the rest for PV nodes; 10 and 24 is reactive power compensation node; The adjustment step length of the capacitor 10 is 0.1; The adjustment step length of 24 node is 0.02, branch 6-9, 6-10, 4-12, 27-28 is transformer branch, transformer range is ±8x1.25% ,power benchmark S_B = 100 MVA.

From Table II, under the initial state, set generator voltage and the change of transformer are 1.0, the system total active network loss is 0.0771. Use CPSOBPNN algorithm, after the reactive power optimization, the system network loss reduces to 0.0573. Network loss rate is 25.16%, so CPSOBPNN algorithm is better than PSO and CPSO, it can enhance the accuracy of convergence, and each node voltage amplitude are qualified, the scope of the reactive power generator were not more limited. The optimal control variables are shown in Table III.

![Fig. 3. Network of IEEE 30 bus system](image)

### Table I: Limits of the Active Power PV Nodes and PV Nodes and Balance of the Node Reactive Power Generation

<table>
<thead>
<tr>
<th>Nodes number</th>
<th>PG</th>
<th>Q_{conv}/pu</th>
<th>Q_{conv}/pu</th>
<th>U_{conv}/pu</th>
<th>U_{conv}/pu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>—</td>
<td>0.596</td>
<td>-0.298</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>0.8</td>
<td>0.48</td>
<td>-0.24</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>0.6</td>
<td>-0.3</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>8</td>
<td>0.2</td>
<td>0.53</td>
<td>-0.265</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>11</td>
<td>0.2</td>
<td>0.15</td>
<td>-0.075</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>13</td>
<td>0.2</td>
<td>0.155</td>
<td>-0.0775</td>
<td>1.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

### Table II: Comparison of Optimal Results for Different Methods About IEEE30 Bus System

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Power losses/pu</th>
<th>Network loss rate(%)</th>
<th>voltage rate (%)</th>
<th>qualified</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO</td>
<td>0.0608</td>
<td>21.01</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>CPSO</td>
<td>0.0591</td>
<td>23.21</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>CPSOBPNN</td>
<td>0.0573</td>
<td>25.16</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

### Table III: Values of Control Variables after Optimization

<table>
<thead>
<tr>
<th>Control variable</th>
<th>Nodes number</th>
<th>voltage</th>
<th>control variable</th>
<th>Nodes number</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_1</td>
<td>1</td>
<td>1.0733</td>
<td>T_1</td>
<td>4-12</td>
<td>6.0000</td>
</tr>
<tr>
<td>V_2</td>
<td>2</td>
<td>1.0703</td>
<td>T_2</td>
<td>6-9</td>
<td>3.0000</td>
</tr>
<tr>
<td>V_5</td>
<td>5</td>
<td>1.0409</td>
<td>T_5</td>
<td>6-10</td>
<td>1.0000</td>
</tr>
<tr>
<td>V_8</td>
<td>8</td>
<td>1.0490</td>
<td>T_8</td>
<td>27-28</td>
<td>1.0000</td>
</tr>
<tr>
<td>V_{11}</td>
<td>11</td>
<td>1.0666</td>
<td>Q_{10}</td>
<td>10</td>
<td>2.0000</td>
</tr>
<tr>
<td>V_{13}</td>
<td>13</td>
<td>1.0727</td>
<td>Q_{24}</td>
<td>24</td>
<td>3.0000</td>
</tr>
</tbody>
</table>

Fig. 4 is PSO, CPSO and CPSOBPNN three algorithm convergence characteristic curves.

Yu Men power grid is 12 generator node, 54 load node, 71 line, 14 reactive power compensation nodes, 19 adjustable transformer branch. The population scale is set to 50; maximum iterating times is set to 100, compare CPSO algorithm and CPSOBPNN algorithm, the average optimization results is shown in Table IV.

### Table IV: Comparison of Optimal Results for Different Methods About YuMen Power Grid

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Network loss /MW</th>
<th>Network loss rate (%)</th>
<th>iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>initial state</td>
<td>16.5</td>
<td>7.12</td>
<td>—</td>
</tr>
<tr>
<td>CPSO</td>
<td>14.2</td>
<td>6.53</td>
<td>56</td>
</tr>
<tr>
<td>CPSOBPNN</td>
<td>12.1</td>
<td>4.96</td>
<td>43</td>
</tr>
</tbody>
</table>
VI. CONCLUSION

This paper used the improved CPSO to optimize the BP neural network to reactive power optimization. Make full use of randomness and stability of the cloud drops, combined the global with local search by the solution space transformation, and cloud genetic and normal cloud realized the across and variation operation of particles, the simulation results show that the improved algorithm can speed up the optimal rate and improve the training precision of the BP neural network, and there is a very good computation efficiency and convergence stability.

ACKNOWLEDGEMENT

The authors would like to thank the editor-in-chief and all the reviewers for their valuable comments and helpful suggestions to improve the quality of this article.

REFERENCES


