International Journal of Control and Automation
Vol.7, No.7 (2014), pp.1-14
http://dx.doi.org/10.14257/ijca.2014.7.7.01

Using Phase Swapping to Solve Load Phase Balancing by ADSCHNN in LV Distribution Network

Chun-guo Fei and Rui Wang

College of Aeronautical Automation, Civil Aviation University of China, China
cgfei@cauc.edu.cn, ruiwang@cauc.edu.cn

Abstract

There are a large amount of advantages to make efficient load phase balancing, such as loss minimization, energy restoration, security, reliability and voltage balance. Optimal load phase balance is obtained by solving the load re-distribution problem as a combinatorial optimization problem. This enables the best switching option that gives a balanced load arrangement among the phases and minimizes power loss to be arrived at. In this paper, adding decaying self-feedback continuous neural network (ADSCHNN) is applied to realize phase swapping for load re-arrangement in the low voltage circuit of the distribution network. The network energy function of the ADSCHNN is constructed for objective function that defines the load phase balancing problem. The ADSCHNN is applied to solve the problem when load is represented in terms of current flow at the connection points, and when load is defined in terms of the real power. The results obtained using ADSCHNN are compared with those from a heuristic algorithm, and from fuzzy logic expert system. Simulations results on real practical data show that the ADSCHNN is very effective and outperforms other known algorithms in terms of the maximum difference of the phase currents or powers.

Keywords: *Phase swapping, load phase balancing, adding decaying self-feedback continuous neural network (ADSCHNN), power loss, load distribution*

1. Introduction

It is well known that customers are supplied three-phase or single-phase from the feeder of the secondary distribution networks. As a consequence, the currents in the three-phase sections are never completely balanced and, in many cases, can be significantly out of balance. It is not uncommon to have as much as 50% difference in magnitude between the highest and lowest loaded phases. In most practical cases, the asymmetry of the loads is the main cause of unbalance. At the high-voltage (HV) and the medium-voltage (MV) levels, the loads are usually three-phase and balanced, although large single- or dual-phase loads can be connected. At the low-voltage (LV) side, loads are usually single-phase, *e.g.*, household general consumption, which includes PCs, lighting systems, *etc*. A single-phase load may be connected to any of the three phases of the feeder. Each feeder in a distribution system usually has a mix of residential, commercial and industrial customers with varying demands depending on the season of the year. Because of load changes and the diversity of loads being on or off, the three-phase imbalances may be substantial. Balancing is accomplished by selecting the phase of the supply for each load so that the total load is distributed as evenly as possible between the phases for each section of feeder. The balancing procedure must consider all possible combinations of phase loads connecting to three phases.

There are a number of benefits that make efficient load phase balancing a worthwhile objective. Phase balancing reduces feeder losses because each phase peak reduction affects

ISSN: 2005-4297 IJCA

the losses for the phases as the square of the current magnitude. Loading on a feeder section is synonymous with the most heavily loaded phase and, in the case of significant imbalance, feeder capacity is used inefficiently. Balancing between phases tends to equalize the phase loading by reducing the largest phase peak load while increasing the load on other phases. This equates to releasing feeder capacity that can be used for future load increase without reinforcing feeder conductors. Released feeder capacity provides more reserve loading capacity for emergency loading conditions. Balancing not only reduces feeder losses, but also improves voltage on a feeder by equalizing the voltage drops in each phase along the feeder. It is realistic to assume that the benefits in improved use of feeder capacity and improved voltage quality are as significance as the value of loss reduction except when loading is already high. Conventionally in South Africa, to reduce the unbalance current in a feeder, the connection phases of some feeders are changed manually after some field measurement and analysis. Although in some cases this process can improve the phase current unbalance, this strategy is more time-consuming, requires supply interruption, unsafe and only last for a while before the process is repeated again [1].

There are two approaches for phase balancing [2]: one is feeder reconfiguration at the system level; the other is phase swapping at the feeder level. Feeder reconfiguration is a process of changing the topological structure of distribution systems by altering the open/closed status of single phase sectionalizing and tie switches, while phase swapping is a process of changing the topological structure of distribution systems by altering the open/closed status of single-phase sectionalizing and tie switches. Feeder reconfiguration is more popular for researchers than phase swapping. Many researches have studied the feeder reconfiguration in the past several decades. Several methods and heuristic algorithms have been proposed for feeder reconfiguration, among which are simulated annealing(SA)[3], neural networks [4, 5], genetic algorithms[6], tabu search (TS) [7, 8], particle swarm optimization(PSO) [9], and other heuristic algorithms [10-17].

Since feeder reconfiguration is primarily designed for load balancing among the feeders, it cannot effectively solve phase balancing problem [18]. Compared with feeder reconfiguration, the phase swapping method has been studied by few researches. Zhu *et al.*, [18] proposed a mixed-integer programming formulation for phase swapping optimization, which is suitable for the linear objective function. Zhu *et al.*, [19] applied simulated annealing to solve a power distribution phase balancing problem with phase swapping method, when phase balancing problems are modeled as non-linear integer programming. Siti *et al.*, [1] proposed a heuristic method for the phase balancing by phase swapping and compared the heuristic algorithm and neural network. Ukil and Siti [20] proposed a fuzzy logic-based load balancing system along with a combinatorial optimization-based implementation system for implementing the load changes to reduce the feeder unbalancing.

For phase swapping, the solution involves a search over relevant switches. Therefore, the loads distribution is a combinatorial optimization problem. Since phase swapping is a complicated combinatorial optimization problem, it is hard to get an optimal load distribution in a large-scale distribution system at a feasible computing time. In this paper, adding decaying self-feedback continuous Hopfield neural network (ADSCHNN) [21] is used to solve the low voltage feeder load phase balancing problem after the energy function is constructed for load phase balancing problem, which can solve the combinatorial optimization problem very well. The ADSCHNN is presented by adding an extra self-feedback to every neuron of continuous Hopfield neural network (CHNN). The extra self-feedback makes the energy of CHNN not to always decrease with time, but increase or maintain. Through the increasing of energy, the ADSCHNN may lead to avoiding the local optimal values. And the ADSCHNN can be realized by hardware. It can provide the fast searching operation. It has been successfully applied to solve combinatorial optimization problem-traveling salesman problem in [21]. Therefore, the ADSCHNN is applied to solve the load phase balancing problem. The comparisons between the ADSCHNN and algorithms in [1] and [20] are conducted. The simulation results showed that the ADSCHNN has a better performance.

2. Formatting Your Paper

Loads are connected to the low voltage distribution system through switches as shown in Figure 1. $L_1 \cdots L_n$ represent the loads and $SW_1 \cdots SW_n$ represent the corresponding switch. Each load is only connected to one of the three phases by the switch selector.

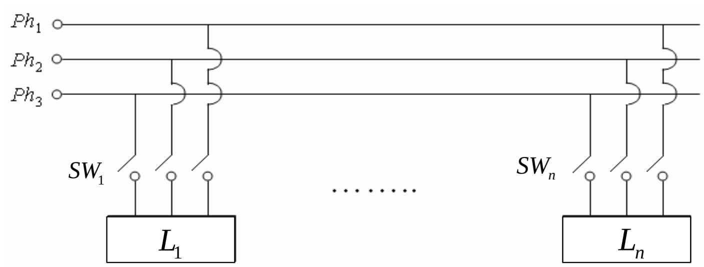

Figure 1. Example of Distribution Feeder

Load re-distribution involves the opening and closing of the switches to change the phase a load is connected such that the system power loss is minimized. To solve the loss reduction problem, the optimal operating condition of load re-distribution is obtained when line losses are minimized. The transfer of load must be conducted under certain objective function to minimize the total real power loss arising from line branches. Therefore, the total power loss function can be expressed as [1]:

$$P_{Loss} = \sum_{i=1}^{n} r_i \frac{P_i^2 + Q_i^2}{|V_i|^2}, \tag{1}$$

subject to the following constrains:

1. The voltage magnitude of each node of each branch V_j must lie within a permissible range, *i.e.*, $V_j^{min} \le |V_j| \le V_j^{max}$. Here a branch can be a transformer, a line section or a tie line with a sectionalizing switch.

2. The line capacity limits.

where r_i , P_i , Q_i , V_i are respectively the resistance, real power, reactive power and voltage of the branch *i*, and *n* is the total number of branches in the system. The aim of load balancing is to minimize the power loss represented by Equ.(1).

In fact, the generalized load balancing problem presents a considerable computational burden for a distribution system of even moderate size because load balancing is a combinatorial optimization problem. Due to the nonlinear nature of the distribution system, a load flow operation has to be performed to determine a new system operating point for each iteration of an optimization algorithm.

3. Adding Decaying Self-feedback Continuous Hopfield Neural Network (ADSCHNN)

The ADSCHNN is proposed by adding extra self-feedback to every neuron of the CHNN and decaying them. The differential equations of the ADSCHNN for the networks are

$$\begin{cases} C_i \dfrac{dy_i}{dt} = -\dfrac{y_i}{R_i} + \sum_{j=1}^{n} w_{ij} x_j + I_i + T_{ii}(t) x_i \\ x_i = \psi(y_i) \\ T_{ii}(t) = T_{ii}(0) e^{-\beta t}, \end{cases} \tag{2}$$

where $C_i > 0$, $R_i > 0$, $i = 1, 2, \cdots, n$, ψ is the activation function, n is the number of neuron, y_i is the internal state of neuron i, x_i is the output of neuron i, I_i is the threshold value of neuron i, and w_{ij} is the symmetric synaptic weight, $T_{ii}(t)$ is extra self-feedback. Like CHNN, the ADSCHNN also can be realized by VLSI. The energy function of ADSCHNN is

$$E = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} x_i x_j - \sum_{i=1}^{n} x_i I_i + \sum_{i=1}^{n} \frac{1}{R_i} \int_0^{x_i} \psi^{-1}(x)\, dx. \tag{3}$$

When the $T_{ii}(t)$ decreases and approaches zero, the ADSCHNN is changed to CHNN. According to [22], the CHNN can move from any initial point in the state space in the direction of decreasing its energy E and converging at one stable equilibrium point that is a minimum of the energy function. During the $T_{ii}(t)$ decreases, the $T_{ii}(t)$ can make the energy E increase. Therefore, ADSCHNN may lead to avoiding the local optimal values and can converge at one stable equilibrium point.

In order to simulate this neural network by software, Equ.(2) should be discretized. Choose the sigmoid function as ψ, *i.e.*, $x_i = \psi(y_i) = \dfrac{1}{1 + e^{-y_i/\varepsilon}}$, $\varepsilon(\varepsilon > 0)$. After Euler discretization of Equ.(2), we have

$$\begin{cases} y_i(k+1) = \left(1 - \dfrac{\Delta t}{C_i R_i}\right) y_i(k) + \dfrac{\Delta t}{C_i} T_{ii}(k) x_i(k) + \dfrac{\Delta t}{C_i} \left(\sum_{j=1}^{n} w_{ij} x_j(k) + I_i \right) \\ x_i(k) = \dfrac{1}{1 + e^{-y_i(k)/\varepsilon}} \\ T_{ii}(k+1) = (1-\beta) T_{ii}(k), \end{cases} \tag{4}$$

where Δt is discrete time. Let $a = 1 - \dfrac{\Delta t}{C_i R_i}$, $\lambda = \dfrac{\Delta t}{C_i}$, $\dfrac{\Delta t}{C_i} T_{ii}(k+1) = (1-\beta) \dfrac{\Delta t}{C_i} T_{ii}(k)$ and $z_i(k) = \dfrac{\Delta t}{C_i} T_{ii}(k)$, then the discrete model of ADSCHNN is gotten as follows,

$$\begin{cases} y_i(k+1) = \alpha y_i(k) + \lambda\left(\sum_{j=1}^{n} w_{ij} x_j(k) + I_i\right) + z_i(k) x_i(k) \\ x_i(k) = \dfrac{1}{1+e^{-y_i(k)/\varepsilon}} \\ z_i(k+1) = (1-\beta) z_i(k), \end{cases} \tag{5}$$

which has energy function $E^{'} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j - \sum_{i=1}^{n} x_i I_i$. So Equ.(5) is changed to

$$\begin{cases} y_i(k+1) = \alpha y_i(k) + \lambda\left(-\dfrac{\partial E^{'}}{\partial x_i}\right) + z_i(k) x_i(k) \\ x_i(k) = \dfrac{1}{1+e^{-y_i(k)/\varepsilon}} \\ z_i(k+1) = (1-\beta) z_i(k), \end{cases} \tag{6}$$

where $\frac{\partial E^{'}}{\partial x_i} = -\sum_{j=1}^{n} w_{ij} x_i(k) - I_i$. Here $E^{'}$ is not only energy function of the network, but also the cost function to be minimized in a given Combinatorial Optimization Problem (COP). When the ADSCHNN is used to solve COP, the COP should be constructed to the energy of ADSCHNN.

4. Problem Analysis and Energy Function Construction

4.1. Load Balancing Problem Analysis

From the description in Section 2, it is expressed that the load balancing problem is solved in terms of minimizing real power loss. In a three-phase four-wire system, Equ.(1) becomes

$$\sum_{i=1}^{3} r_i \frac{P_i^2 + Q_i^2}{|V_i|^2} = \sum_{i=1}^{3} r_i \frac{|V_i|^2 |I_i|^2 \cos^2\varphi + |V_i|^2 |I_i|^2 \sin^2\varphi}{|V_i|^2} = \sum_{i=1}^{3} r_i |I_i|^2 = r_1 |I_1|^2 + r_2 |I_2|^2 + r_3 |I_3|^2 . \tag{7}$$

In general, each phase has the same internal resistance r which is constant. Therefore, Equ.(7)

$$\sum_{i=1}^{3} r_i \frac{P_i^2 + Q_i^2}{|V_i|^2} = r\left(|I_1|^2 + |I_2|^2 + |I_3|^2\right), \tag{8}$$

constraining to $|I_1| + |I_2| + |I_3| = C$, C can be a complex or real constant depending on the load. To minimize the total real power losses means

$$\begin{aligned} &\min\left(|I_1|^2 + |I_2|^2 + |I_3|^2\right), \\ &\text{subject to } |I_1| + |I_2| + |I_3| = C. \end{aligned} \tag{9}$$

The method of Lagrange multipliers are used to solve Equ.(9). Create the non-constrained function as

$$L(|I_1|,|I_2|,|I_3|,\lambda) = |I_1|^2 + |I_2|^2 + |I_3|^2 + \lambda(|I_1| + |I_2| + |I_3| - C) \tag{10}$$

The gradient for this new function Equ.(10) is

$$\begin{aligned}
\frac{\partial L}{\partial |I_1|} &= 2|I_1| + \lambda_1 = 0 \\
\frac{\partial L}{\partial |I_2|} &= 2|I_2| + \lambda_1 = 0 \\
\frac{\partial L}{\partial |I_3|} &= 2|I_3| + \lambda_1 = 0 \\
\frac{\partial L}{\partial \lambda_1} &= |I_1| + |I_2| + |I_3| - C = 0.
\end{aligned} \tag{11}$$

From Equ.(11), $|I_1| = |I_2| = |I_3| = \frac{1}{3}C$ can be obtained. Therefore, when $|I_1| = |I_2| = |I_3| = \frac{1}{3}C$, the total real power losses are minimal. If the loads are pure resistance, the minimum power losses are achieved when $P_1 = P_2 = P_3 = \frac{P}{3}$, where $P_i\ (i = 1,2,3)$ is the real power per phase and P is the sum of three phases real powers. So we can solve the load balancing problem by distributing equally the load current or power to three phases, according to the load property.

4.2. Energy Function Construction for the ADSCHNN

From the above analysis, we know the load balancing problem means all the loads are distributed to three phases equally, with minimum differences among the individual sums of three phases. These two parameters can be used for power or current. So there is an ideal phase balance of load $Load_{ideal}$, which is equal to the one-third of the sum of all the loads $Load_{ideal} = \frac{1}{3}\sum_{j=1}^{n} Load(j)$. n is the number of all the loads. The load balancing is complete, if the sum of every phase loads satisfies $\sum_{i=1}^{m} Load_{phase}(i) = Load_{ideal}$. m is the number of load points which are connected to one phase. Therefore, in a three-phase four-wire system, we have load balancing when $Load_{phase1} = Load_{phase2} = Load_{phase3}$.

In order to solve the load balancing problem, the solution of load balancing problem is mapped to the ADSCHNN. So a transposition matrix with $3 \times n$ is needed to show the configuration of all the loads. The component of the transposition matrix is either 1 or 0. The transposition matrix also indicates the neuron output. At the same time, the energy function for load balancing problem is constructed considering under certain restrictions

1) The transposition matrix has only one “1” component in the one column.
2) The sum of the all component of transposition matrix is n.
3) The difference among the individual sums of all the loads in three phases is minimum, which means the total line losses are minimum.

Point 1) above means each load is connected to only one feeder. Point 2) above indicates the number of closed switches equals the number of loads connected. Point 3) above is an object function. From 1, 2 and 3, we construct the energy function as

Equ.(12), where $Load$ is a matrix with $1 \times n$ containing all the loads. A, B, C are the coupling parameters corresponding to the constraints and the object function.

$$E = \frac{A}{2} \sum_{j=1}^{n} \sum_{i=1}^{3} \sum_{\substack{l \neq i \\ l=1}}^{3} x_{ij} x_{lj} + \frac{B}{2} \left(\sum_{i=1}^{3} \sum_{j=1}^{n} x_{ij} - n \right)^2 + \frac{C}{2} \sum_{i=1}^{3} \left(\sum_{j=1}^{n} x_{ij} Load(j) - Load_{ideal} \right)^2 \tag{12}$$

The first two terms in Equ.(12) correspond to 1) and 2). It is, if 1) and 2) are satisfied at the same time, that the first two terms of Equ.(12) are equal to zero, otherwise, they are not zero. So the two terms are the constrained terms. When the constrained terms are equal to zero, Each load only belongs to one phase. The third term is object function. $x_{ij} = 1$ denotes that load j is connected to Ph_i, while $x_{ij} = 0$ denotes that load j is not connected to Ph_i. If the difference among the individual sums of all the loads current in three phases is the smallest, the object function is minimum.

From Equ.(12), we have

$$\frac{\partial E}{\partial x_{ij}} = A \sum_{\substack{l \neq i \\ l=1}}^{3} x_{lj} + B \left(\sum_{i=1}^{3} \sum_{j=1}^{n} x_{ij} - n \right) + C \left(\sum_{j=1}^{n} x_{ij} Load(j) - Load_{ideal} \right). \tag{13}$$

putting Equ.(13) into Equ.(6), we get the discrete dynamics of the ADSCHNN for the load balancing problem as follows,

$$\begin{cases} y_{ij}(k+1) = a y_{ij}(k) + \lambda \left[-A \sum_{\substack{l \neq i \\ l=1}}^{3} x_{lj} - B \left(\sum_{i=1}^{3} \sum_{j=1}^{n} x_{ij} - n \right) - C \left(\sum_{j=1}^{n} x_{ij} Load(j) - Load_{ideal} \right) \right] \\ \qquad\qquad + z_{ij}(k) x_{ij}(k) \\ x_{ij}(k) = \dfrac{1}{1 + e^{-y_{ij}(k)/\varepsilon}} \\ z_{ij}(k+1) = (1 - \beta) z_{ij}(k). \end{cases} \tag{14}$$

5. Simulation Results

5.1. Current Loads

The ADSCHNN is used to optimize the practical field data used in [1], where the loads were addressed in terms of currents. The parameters for the ADSCHNN are set as $\alpha = 1$, $\lambda = 0.25$, $A = 0.1$, $B = 0.1$, $C = 0.35$, $z_{ij}(0) = 0.13$, $\beta = 0.001$. Initial conditions of the ADSCHNN are $y_{ij} = 1 \quad (i = 1, 2, 3 \quad j = 1, 2, \cdots, n)$. The ADSCHNN needs 1962 iterations to get the optimal result. The results are shown in Table 1, where "1" means the respective load is connected to ph_1, "2" to ph_2, "3" to ph_3, $\triangle I_{ph\text{-}max}$ is the maximum difference of the phase currents, which ideally should be zero if there is totally balanced. The ADSCHNN gives a better phase balancing result, which is almost equal to zero, compared to the parameter $\triangle I_{ph\text{-}max}$. The ADSCHNN give a $\triangle I_{ph\text{-}max}$ of 0.17 compared to 42 obtained for neural network (NN) and 24.7 obtained for heuristic method (HE) in [1]. This indicates that the ADSCHNN gives a better solution to the load balancing problem compared to other algorithms investigated in [1].

Table 1. The Comparison Between the ADSCHNN and the Algorithm of [1]

Current	Unbalanced		Balanced		
		Switch	NN	HE	ADSCHNN
I_1(A)	40.16	1	1	1	1
I_2(A)	92.61	2	2	1	2
I_3(A)	90.77	3	3	2	3
I_4(A)	40.61	1	1	3	1
I_5(A)	88.47	2	3	3	1
I_6(A)	5.73	3	1	1	3
I_7(A)	34.93	1	3	3	2
I_8(A)	80.50	2	1	2	3
$I_{\backslash 9}$(A)	0.97	3	2	2	2
I_{10}(A)	13.75	1	1	1	2
I_{11}(A)	20.07	2	3	2	2
I_{12}(A)	19.67	3	2	2	1
I_{13}(A)	59.77	1	3	1	2
I_{14}(A)	26.94	2	2	3	1
I_{15}(A)	19.68	3	2	3	3
I_{16}(A)	1.51	1	1	1	3
I_{17}(A)	73.93	2	2	2	3
I_{18}(A)	44.06	3	3	3	1
I_{19}(A)	92.24	1	1	2	2
I_{20}(A)	46.13	2	1	1	1
I_{21}(A)	41.44	3	2	2	3
I_{22}(A)	83.77	1	3	3	1
I_{23}(A)	51.99	2	1	1	3
I_{24}(A)	20.06	3	3	2	2
I_{25}(A)	66.54	1	2	3	2
I_{26}(A)	82.97	2	2	1	3
I_{27}(A)	1.94	3	3	3	1
I_{28}(A)	67.44	1	1	3	1
I_{29}(A)	37.56	2	1	2	2
I_{30}(A)	82.34	3	1	1	2
I_{31}(A)	94.06	1	1	1	3
I_{32}(A)	22.88	2	2	2	1
I_{33}(A)	60.07	3	1	1	1
I_{34}(A)	48.11	1	3	3	2
I_{35}(A)	88.23	2	1	3	1
I_{36}(A)	75.44	3	1	3	3
I_{37}(A)	45.19	1	2	2	2
I_{38}(A)	1.83	2	3	3	2
I_{39}(A)	81.31	3	2	3	2
I_{40}(A)	60.92	1	1	1	3
I_{41}(A)	78.40	2	3	2	1
I_{42}(A)	91.25	3	2	1	3
I_{443}(A)	73.08	1	2	2	2
I_{43}(A)	17.45	2	2	2	1
I_{45}(A)	44.02	3	1	1	1
I_{ph1}(A)	822.1	-	746.1	761.4	770.24
I_{ph2}(A)	809.9	-	778.5	786.1	770.36
I_{ph3}(A)	678.8	-	788.1	771.4	770.19
$\triangle I_{ph-max}$(A)	143.3	-	42	24.7	0.17

5.2. Power Loads

Now we use the ADSCHNN to another kind of load balancing problem, *i.e.*, power loads. The input power loads for simulation are acquired from a load data survey in a South African city, which were used in [20]. All the loads are power and integer. The sum of loads for three phases are represented by a as 3×1 matrix. The simulation results are presented in Table 2. Table 2 gives the simulation results and results comparison. The ADSCHNN needs 1830 iterations, 1841 iterations, 1806 iterations, 1823 iterations, 1808 iterations and 957 iterations to get optimal results in Case 1 to Case 6, respectively. The parameters and initial conditions of the ADSCHNN are the same as simulation in Section 5.1, except $\beta = 0.002$ for Test Case 2 to Test Case 5. Table 2 shows that the ADSCHNN gets the complete load balancing for Test Case 1 to Test Case 5, i.e. the sum of every phase loads is equal, because the $Load_{ideal}$ is integer. For Test Case 6, complete load balancing is not achieved, since the $Load_{ideal}$ is not integer. In order to show the effect of the ADSCHNN, Table 3 and Table 4 give the load distribution of Test Case 1 before and after optimization by the ADSCHNN. The simulations show that the ADSCHNN outperforms the algorithm in [20].

Table 2. Simulation Results and Comparison between the ADSCHNN and Algorithm of [20]

Test Case	Initial Data		Algorithm of [20]		The ADSCHNN	
	Initial Load (KW)	△$P_{ph\text{-}max}$	Final Load (KW)	△$P_{ph\text{-}max}$	Final Load (KW)	△$P_{ph\text{-}max}$
1	$\begin{bmatrix}245\\120\\82\end{bmatrix}$	163	$\begin{bmatrix}146\\150\\151\end{bmatrix}$	5	$\begin{bmatrix}149\\149\\149\end{bmatrix}$	0
2	$\begin{bmatrix}157\\134\\120\end{bmatrix}$	17	$\begin{bmatrix}139\\133\\139\end{bmatrix}$	6	$\begin{bmatrix}137\\137\\137\end{bmatrix}$	0
3	$\begin{bmatrix}140\\145\\156\end{bmatrix}$	16	$\begin{bmatrix}145\\145\\151\end{bmatrix}$	6	$\begin{bmatrix}147\\147\\147\end{bmatrix}$	0
4	$\begin{bmatrix}205\\170\\162\end{bmatrix}$	43	$\begin{bmatrix}181\\177\\179\end{bmatrix}$	4	$\begin{bmatrix}179\\179\\179\end{bmatrix}$	0
5	$\begin{bmatrix}170\\95\\83\end{bmatrix}$	87	$\begin{bmatrix}115\\121\\112\end{bmatrix}$	9	$\begin{bmatrix}116\\116\\116\end{bmatrix}$	0
6	$\begin{bmatrix}117\\74\\42\end{bmatrix}$	75	$\begin{bmatrix}72\\80\\81\end{bmatrix}$	9	$\begin{bmatrix}78\\78\\77\end{bmatrix}$	1

Table 3. Load Distribution for the Three Phases from [20] Before Optimization

SL. No.	Phase 1 (KW) Per house	Phase 2(KW) Per house	Phase 3 (KW) Per house
1	5	4	1
2	4	2	1
3	4	3	2
4	3	1	1
5	2	3	3
6	1	2	2
7	2	1	1
8	3	3	3
9	5	1	1
10	2	2	2
11	5	5	2
12	2	4	2
13	6	2	3
14	1	1	1
15	2	4	2
16	2	2	1
17	5	3	2
18	1	4	1
19	2	2	2
20	2	2	2
21	3	1	1
22	6	2	1
23	8	3	2
24	10	2	1
25	9	1	1
26	4	3	2
27	2	4	1
28	3	3	2
29	5	2	1
30	6	1	2
31	6	2	3
32	6	3	1
33	9	2	2
34	3	2	1
35	10	2	1
36	12	3	2
37	15	4	1
38	3	1	1
39	2	2	3

40	1	1	1
41	3	2	2
42	2	4	1
43	1	3	2
44	5	2	2
45	4	1	1
46	15	1	1
47	12	2	2
48	10	5	3
49	9	2	1
50	2	3	2
Total	245	120	82

Table 4. Load Distribution for the Three Phases after Optimization by ADSCHNN

SL. No.	Phase 1 (KW) Per house	Phase 2(KW) Per house	Phase 3 (KW) Per house
1	5	4	1
2	4	2	1
3	3	2	4
4	1	1	3
5	2	3	3
6	1	2	2
7	2	1	3
8	1	3	1
9	3	1	2
10	5	2	5
11	5	2	4
12	6	2	3
13	1	2	2
14	2	2	2
15	2	2	1
16	2	1	3
17	4	1	1
18	2	4	1
19	3	5	2
20	1	2	2
21	8	2	1
22	1	2	2
23	9	1	3
24	2	6	10
25	3	2	3

26	2	2	4
27	6	1	2
28	3	1	1
29	1	4	1
30	9	2	2
31	1	1	2
32	3	3	6
33	2	5	2
34	4	6	3
35	1	3	1
36	1	2	2
37	1	2	12
38	2	10	3
39	1	15	2
40	1	2	3
41	1	4	3
42	2	5	2
43	1	15	3
44	1	3	2
45	2	2	4
46	2	2	1
47	1	2	2
48	1		2
49	1		10
50	12		9
51	5		
52	1		
53	3		
Total	149	149	149

6. Conclusion

The total power losses of distribution systems can be effectively reduced by the proper load distribution which is necessary to achieve load balancing under the certain objective function of the total power loss. Therefore, the load re-distribution is a combinatorial optimization problem aimed at giving the best switching option that enables a balanced load arrangement among the phases and minimizes the power loss. According to the analysis reported in this work, the ADSCHNN is used to solve the load balancing problem, after constructing its energy function. The paper has shown that through simulation results based on real practical data, load balancing is achieved by the ADSCHNN and the total power loss is minimized. The results are compared with those from a heuristic algorithm [1] and those from a fuzzy logic expert system [20]. The ADSCHNN showed better load balancing than other two algorithms.

Acknowledgements

This work is supported by The Natural Science Foundation of Tianjin (Grant.13JCYBJC39000), Central College Basic Research Foundation of China (ZXH2012H003, 3122013H001) and Scholars of Civil Aviation University of China (2012QD21X).

References

[1] M. W. Siti, D. V. Nicolae, A. A. Jimoh, and A. Ukil, "Reconfiguration and Load Balancing in the LV and MV Distribution Networks for Optimal Performance", IEEE Transactions on Power Delivery, vol. 22, **(2007)**, pp. 2534-2540.

[2] J. Zhu, G. Bilbro, and C. Mo-Yuen, "Phase Balancing using Simulated Annealing", Power Systems, IEEE Transactions on, vol. 14, **(1999)**, pp. 1508-1513.

[3] C.-T. Su and C.-S. Lee, "Feeder Reconfiguration and Capacitor Setting for Loss Reduction of Distribution Systems", Electric Power Systems Research, vol. 58, **(2001)**, pp. 97-102.

[4] H. Kim, Y. Ko, and K. H. Jung, "Artificial Neural-network based Feeder Reconfiguration for Loss Reduction in Distribution Systems", Power Delivery, IEEE Transactions on, vol. 8, **(1993)**, pp. 1356-1366.

[5] D. Bouchard, A. Chikhani, V. I. John, and M. M. A. Salama, "Applications of Hopfield Neural Networks to Distribution Feeder Reconfiguration", in Neural Networks to Power Systems, 1993. ANNPS '93., Proceedings of the Second International Forum on Applications, **(1993)**, pp. 311-316.

[6] C. Tsai-Hsiang and C. Jeng-Tyan, "Optimal Phase Arrangement of Distribution Transformers Connected to a Primary Feeder for System Unbalance Improvement and Loss Reduction Using a Genetic Algorithm", Power Systems, IEEE Transactions on, vol. 15, **(2000)**, pp. 994-1000.

[7] H. Mori and Y. Ogita, "A Parallel Tabu Search based Method for Reconfigurations of Distribution Systems, in Power Engineering Society Summer Meeting, 2000. IEEE, **(2000)**, pp. 73-78 vol. 1.

[8] D. Zhang, Z. Fu, and L. Zhang, "An Improved TS Algorithm for Loss-minimum Reconfiguration in Large-scale Distribution Systems", Electric Power Systems Research, vol. 77, **(2007)**, pp. 685-694.

[9] J. Olamaei, T. Niknam, and G. Gharehpetian, "Application of Particle Swarm Optimization for Distribution Feeder Reconfiguration Considering Distributed Generators", Applied Mathematics and Computation, vol. 201, **(2008)**, pp. 575-586.

[10] M. Y. Cho, T. E. Lee, C. S. Chen, and C. N. Lu, "A New Approach for Feeder Reconfiguration to Minimize Distribution System Loss", Proceedings. Joint International Power Conference, **(1993)**, pp. 596-600.in Athens Power Tech, 1993. APT 93.

[11] T. Niknam, "An Efficient Hybrid Evolutionary Algorithm based on PSO and HBMO Algorithms for Multi-objective Distribution Feeder Reconfiguration", Energy Conversion and Management, vol. 50, **(2009)**, pp. 2074-2082.

[12] M. A. Kashem, G. B. Jasmon, and V. Ganapathy, "A New Approach of Distribution System Reconfiguration for Loss Minimization, International Journal of Electrical Power & Energy Systems, vol. 22, **(2000)**, pp. 269-276.

[13] Y.-J. Jeon and J.-C. Kim, "Application of Simulated Annealing and Tabu Search for Loss Minimization in Distribution Systems", International Journal of Electrical Power & Energy Systems, vol. 26, **(2004)**, pp. 9-18.

[14] J.-H. Ji, "The Refined Strategy for Substation Main Transformer and Feeder Load Balancing", International Journal of Electrical Power & Energy Systems, vol. 19, **(1997)**, pp. 87-91.

[15] J. A. Martin and A. J. Gil, "A New Heuristic Approach for Distribution Systems Loss Reduction", Electric Power Systems Research, vol. 78, **(2008)**, pp. 1953-1958.

[16] T. Niknam and E. Azad Farsani, "A Hybrid Self-adaptive Particle Swarm Optimization and Modified Shuffled Frog Leaping Algorithm for Distribution Feeder Reconfiguration", Engineering Applications of Artificial Intelligence, vol. In Press, Corrected Proof, **(2010)**,

[17] D.-J. Shin, J.-O. Kim, T.-K. Kim, J.-B. Choo, and C. Singh, "Optimal Service Restoration and Reconfiguration of Network Using Genetic-Tabu Algorithm", Electric Power Systems Research, vol. 71, **(2004)**, pp. 145-152.

[18] J. Zhu, M.-Y. Chow, and F. Zhang, "Phase Balancing Using Mixed-integer Programming", IEEE Transactions on Power Systems, vol. 13, **(1998)**, pp. 1487-1492.

[19] J. Zhu, G. Bilbro, and M.-Y. Chow, "Phase Balancing Using Simulated Annealing", IEEE Transactions on Power Systems, vol. 14, **(1999)**, pp. 1508-1513.

[20] A. Ukil and W. Siti, "Feeder Load Balancing Using Fuzzy Logic and Combinatorial Optimization-based Implementation", Electric Power Systems Research, vol. 78, **(2008)**, pp. 1992-1932.

[21] C. Fei, G. Qi, and A. Jimoh, "Adding Decaying Self-feedback Continuous Hopfield Neural Network and its Application to TSP", International Review on Modelling and Simulation, vol. 3, **(2010)**, pp. 272-282.

[22] M. M. Gupta, L. Jin, and N. Homma, "Static and dynamic neural networks", Canada: IEEE Press and John Wiley &Sons, **(2003)**.

Authors

Chunguo Fei received the M.Sc. degree in Department of Automation from the Tianjin University of Science and Technology, Tianjin, China, in 2003, and Ph.D. degree at Department of Automation from Shanghai Jiaotong University, Shanghai, China, in 2006. From 2009 to 2010, He was a post-doctoral fellow in Electrical Engineering Department, Tshwane University of Technology, Pretoria North, South Africa. Now, he is an Associate Professor at the College of Aeronautical Automation, Civil Aviation University of China, Tianjin, China. His research interests include neural network, power optimization, power fault location and classification.

Rui Wang received the B.S. degree in industry automation from Tianjin University of Science and Technology, Tianjin, China, in 2003, the M.S. degree in electrical engineering from Guizhou University, Guiyang, Guizhou, China, in 2007, the Ph.D. degree in electrical engineering from Old Dominion University, Norfolk, VA, in 2011, and later as a postdoctoral researcher in the System Research Laboratory at Old Dominion University for one year.

She is currently with the Electrical and Automation Engineering Faculty at Civil Aviation University of China, Tianjin, China. Her research interests include modeling control systems, control theory and its application.

