Unbalanced three-phase distribution state estimation using cooperative particle swarm optimization

Ke-yan Liu, *Member, IEEE*, Wanxing Sheng, *Senior Member*, *IEEE*, Kaiyuan He

***Abstract*—Unbalanced three-phase state estimation for advanced distribution management system (DMS) in a smart distribution grid is presented in this paper. With the consideration of renewable distributed generation and variable loads, the distribution state estimation (DSE) is investigated. To solve the state estimation optimization problem, the cooperative particle swarm optimization (Co-PSO) algorithm is proposed based on the particle swarm optimization algorithm. The proposed method can estimate variable loads and renewable distributed generation (DG) values efficiently. Experiment is made on the IEEE 123-bus radial distribution case with variable loads and DG units. Test results and the comparisons with other evolutionary optimization algorithms such as original PSO, improved PSO algorithms, genetic algorithm (GA) demonstrate that the proposed Co-PSO is effective for the DSE problems.**

***Index Terms*—Distribution state estimation, unbalanced distribution system, particle swarm optimization**

I. Introduction

The idea of applying state estimation for electric power networks emerged in the early 1970s [1]. The state estimation was initially applied to transmission networks to determine the best estimate of all the generation, loads, power flow and voltages at given points based on available real time SCADA measurements. State estimation is an optimization problem including discrete and continuous variables, whose objective function is to minimize the difference between calculated and measured values of variables, i.e., voltage of nodes, and active/reactive powers in the branch.

Advanced distribution systems must be ready to face upcoming technical and economic constraints, such as increase of distributed generation connections, changes in network structure and voltage profiles. In this context, new centralized automation functions in distribution system control centers are needed in order to ensure the control of both distribution network and connected DGs. Consequently, state estimators need to be developed for future distribution systems to assess the network's state in real time, i.e., 10 minutes typical time frame, based on real, pseudo-, and virtual measurements. In many distribution networks, the measurements available are insufficient to make system observable [2]. Pseudo measurements from load estimates and short-term load forecasts can be used to allow a distribution state estimator to function [3].

This work was supported in part by the National Natural Science Foundation of China (51177152) and State Grid Corporation of China Research Program (PD71-12-010).

Some successful methods have been applied to state estimation studies, such as Kalman Filter and its variants, singular value decomposition, Least Squares and particle swarm optimization etc. In this paper, the authors propose the use of improved cooperative particle swarm optimization (PSO) to solve unbalanced three-phase distribution state estimation. Test results are presented on the IEEE 123-bus unbalance distribution feeder test case. The remainder of the paper is organized as follows. Section II states the formulation of the proposed DSE formulation for unbalanced three-phase distribution network with DG units. The proposed particle swarm optimization algorithm to solve the optimization problem is described in Section III. Section IV provides numerical results and comparisons with the proposed approach using various test systems with DG units. Section V summarizes the main contributions and conclusions of the paper.

II. Problem Statement

The DSE problem is an optimization problem with equality and inequality constraints [4]. The well-known weighted least of squares (WLS) technique is applied in the paper since it is the most common estimator because of its accuracy and efficiency provided an error-free set of measurements is available. The objective function is the summation of difference between the measured and calculated values. DSE including DG units can be expressed as follows.

$$\min f(\mathbf{x}) = \sum_{j=1}^{n_x} \omega_j [\frac{z_j - h_j(\mathbf{x})}{z_j}]^2 \tag{1}$$

$$s.t. \quad \begin{cases} P_s = P_e + {I_b}^2 R_b \\ {I_b}^2 = \dfrac{P_s^2 + Q_s^2}{U_s^2} = \dfrac{P_e^2 + Q_e^2}{U_e^2}, \forall b \in 1,2,\ldots,b_n \\ Q_s = Q_e + {I_b}^2 X_b \end{cases}$$

$$\mathbf{x} = [\overline{P_G}, \overline{P_{Load}}]_{1\times n}$$

$$\overline{P_G} = [P_G^1, P_G^2, \ldots, P_G^i, \ldots, P_G^{N_g}] \qquad i \in [1, N_g]$$

$$\overline{P_L} = [P_{Load}^1, P_{Load}^2, \ldots, P_{Load}^i, \ldots, P_{Load}^{N_L}] \quad i \in [1, N_L]$$

$$n = N_g + N_L$$

where $\mathbf{x}$ is the state variables vector including the output of load and DG units outputs. z_j is the measured value, ω_j is

978-1-4799-3656-4/14/$31.00 ©2014 IEEE

the weighting factor of the jth measured variable. $h_j(\mathbf{x})$ is the jth measured variable. n_x is the number of measurement. N_g is the number of DG units with variable outputs. N_L is the number of loads with variable outputs. P_G^i is the active power of the DG. P_{Load}^i is the active power of the load. n is the number of state variables.

III. PROPOSED IMPROVED CO-PSO ALGORITHM IN DISTRIBUTION STATE ESTIMATION

A. *Basic PSO algorithm*

The PSO method is a relatively new optimization technique introduced by Kennedy and Eberhart [5]. The PSO searching mechanism for an optimal solution resembles the social behavior of a flock of flying birds during their searching for food. Each of the swarm's individuals is called a particle. The PSO particle represents a candidate potential solution for the optimization problem and each particle is assigned a velocity vector v_i and a position vector x_i. For a swarm of *m*-particle in R^n hyperspace, each particle is associated with the following position and velocity vectors:

$$s_i = [x_1^i, x_2^i, ..., x_n^i], \qquad i = 1, 2, ..., m \tag{2}$$

$$v = [v_1, v_2, ..., v_m] \tag{3}$$

where i is the particle index, v is the swarm velocity vector, and n is the optimization problem dimension. The new position of the particle is related to its previous location through the following relation:

$$s_i^{(k+1)} = s_i^{(k)} + v_i^{(k+1)} \tag{4}$$

The velocity update vector of particle i at *k*th iteration is calculated as follows:

$$\begin{aligned} v_i^{(k+1)} &= wv_i^{(k)} + c_1 * rand()(pbest_i^{(k)} - s_i^{(k)}) \\ &+ c_2 * rand()(gbest^{(k)} - s_i^{(k)}) \end{aligned} \tag{5}$$

where w is inertia weight, c_1 and c_2 are the cognitive factor and social factor, respectively; $rand()$ is in the range [0,1], $pbest_i^{(k)}$ expresses the individual best position associated with particle *i*, and $gbest^{(k)}$ denotes the global best position associated with the whole neighborhood at *k*th iteration.

B. *The proposed enhanced cooperative PSO algorithm*

Cooperative PSO is a method of casting particle swarm optimization into a cooperative framework. It was reported the Cooperative PSO has significant improvement in performance, especially in terms of solution quality and robustness [6]. The improved Cooperative PSO (Co-PSO) is proposed to solve three-phase state estimation in this section.

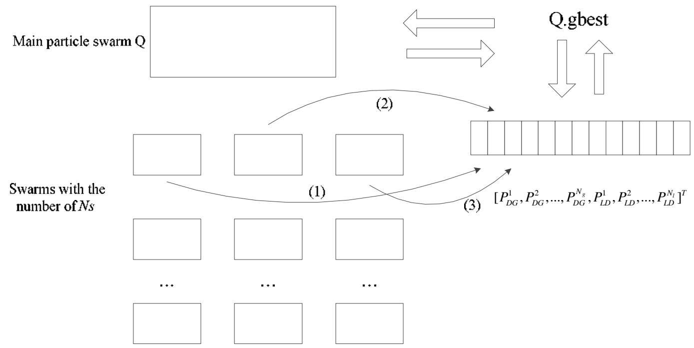

Fig. 1. The concept chart of proposed Co-PSO

The concept chart of the proposed Co-PSO is shown in Fig.1. As shown in Fig.1, it includes two groups particle swarm. One group is the main particle swam and the other group is the sub-swarms to make cooperation with the main particle swarm. The detail cooperation process between the main swam and sub-swarms is described as follows:

1) The main particle swarm searches feasible solution with iterations; if it iterates *N* generations, then the main particle swarm shares its swarm *gbest* to sub-swarms;
2) The first sub-swarm gets the whole *gbest* as its initial particle value, then searches the first dimension of state variable and it is one dimension searching process. Further, the first sub-swarm sends the best one dimension solution to the corresponding *gbest* dimension value;
3) The main particle swarm sends its new *gbest* vector to the second sub-swarm, and the second sub-swarm takes the *gbest* vector as its initial one particle value and sends back the second dimension best solution after the sub-swarm searching process;
4) The sub-swarms sequentially get the newest *gbest* vector and send back the corresponding best solution to the main particle swarm;
5) The main particle swarm continues its own searching using the cooperative *gbest* vector; if the main particle swarm needs new cooperative process, then go to step 1).

The initial population of the state variable can be expressed as follows:

$$\begin{aligned} X &= [X_1, X_2, ..., X_i, ..., X_{N_S}], \\ X_i &= [x_j]_{1\times n}, \qquad i = 1, 2, 3, ..., N_S \\ x_j &= rand(\cdot) \times (P_{DG,\max}^j - P_{DG,\min}^j) + P_{DG,\min}^j, \\ & j = 1, 2, 3, ..., N_g \\ x_j &= rand(\cdot) \times (P_{Ld,\max}^j - P_{Ld,\min}^j) + P_{Ld,\min}^j, \\ j &= N_g + 1, N_g + 2, N_g + 3, ..., N_g + N_l \end{aligned} \tag{6}$$

where is the position of the *j*th state variable and $rand(\cdot)$ is a random function generator between 0 and 1. N_S is the number of state variables.

Step 1: Randomly initialize *n*-position vector for each particle and *m*-velocity vector;
Step 2: Record the fitness values of the entire population, and save the initial *pbest* and *gbest* value;
Step 3: For each iteration
Step 4: For each particle

- Evaluate the fitness value and compare it to its *pbest*

  if $f(s_i^{(k)}) < f(pbest_i^{(k-1)}) \Rightarrow pbest_i^{(k)} = s_i^{(k)}$

  else if $f(s_i^{(k)}) \geq f(pbest_i^{(k-1)}) \Rightarrow pbest_i^{(k)} = pbest_i^{(k-1)}$

  end// For each particle.
- Save the pbest new vector;
- $gbest^{(k)} = \min\left\{f(s_1^{(k)}), f(s_2^{(k)}), ..., f(s_m^{(k)})\right\}$
- Update velocity vector and position vector, and reinforce solution bounds if violation occurs;

Step 5: If the current iteration number reaches t*N, then the main particle swarm shares the *gbest* to sub-swarm and collects the new *gbest* vector;
Step 6: If stopping criteria satisfied, then

- Maximum number of iterations is reached,
- Maximum change in fitness value is less than ε for q iterations:

$$\left|f(gbest^{(k)}) - f(gbest^{(k-h)})\right| \leq \varepsilon, \quad h = 1,2,...,q$$

end// For each iteration.
Otherwise go to Step3.

Fig. 2. The pseudocode of the proposed Co-PSO algorithm

The convergence condition is any of the following condition is satisfied: 1) the iteration number reaches the maximal limit; 2) the objective value is less than a threshold.

$$it > T_{\max} \quad or \quad F(Q.gbest) < \varepsilon$$

where $T_{\max}$ is the iteration limit, threshold $\varepsilon = k(m-n)$, m is the dimension of measurement, n is the dimension of status variables, k is one constant value.

The equation constraints can be met in the searching period of PSO by power flow computation. Through penaltying the in-equation constraints to the objective function, the fitness evaluation can be formed as follows:

$$F(X) = f(X) \tag{7}$$

where $f(X)$ is the objective function values of DSE problem.

The proposed Co-PSO algorithm could be summarized as shown in Fig.2.

IV. Experiments and Results

A. *Experiments setting*

To demonstrate the effectiveness of the proposed method, the algorithm was implemented and applied to DSE on distribution test systems with DG units. The IEEE 123-bus, three-phase unbalanced distribution network was considered with DG units and tested. The algorithm is implemented, evaluated and compared in the following environments:

• The proposed algorithm was coded in MATLAB and run on an Intel i5-3210M 2.5 GHz laptop with 4 GB RAM. In the implementation of Co-PSO algorithm, the number of main particle swarm is 60 and the sub-swarms number is 20, the maximal iteration number $T_{\max}$ is set to 1000, the inertia weight w is 0.7298, the cognitive and social parameters c_1, c_2 are all set to 1.496, which can be shown in TABLE I.

• The state variables are the values of load and DG whose outputs are variables, rather than the constant values as used by conventional state estimation.

• The algorithm verification and comparisons were made with original PSO, genetic algorithm (GA) and multiple-start PSO (MPSO), simulated annealing PSO (SA-PSO), Chaos-PSO [7]. These algorithms are also fully implemented and compared. Their parameters and settings are shown in TABLE II.

TABLE I The Parameters of Co-PSO

Particle number	w	c_1	c_2	Threshold of convergence	Iteration limit
60	0.7298	1.496	1.496	10^{-6}	1000

TABLE II Values of Parameters of Each Algorithm

PSO		GA		MPSO	
Parameter	Value	Parameter	Value	Parameter	Value
Swarm	60	Population	200	Swarm	60
c1=c2	1.4962	Crossover	0.5	c1=c2	1.4962
w	0.7298	Mutation	0.05	w	0.7298
Iterations	1000	Iterations	300	Iterations	1000
				Interval	50
SA-PSO		**Chaos-PSO**		**GA-PSO**	
Parameter	Value	Parameter	Value	Parameter	Value
Swarm	60	Swarm	60	Swarm	60
c1=c2	1.4962	c1=c2	1.4962	c1=c2	1.4962
w	0.7298	w	0.7298	w	0.7298
Crossover	0.5	u	3.75	Crossover	0.5
Mutation	0.05	M	30	Mutation	0.05
Iterations	1000	Iterations	1000	Iterations	1000
Cooling coefficient	0.8				
Control parameter	1.0e+6				

B. *Simulation results on IEEE 123-bus*

The IEEE 123-bus test case, as shown in Fig.3, were considered with DG and tested. The nominal voltage is 4.16kV and detail network parameters can be got from reference [8]. It is a mixed multi-phase unbalanced system with variant values of DG units and loads. It is assumed that there are 4 DG units whose specifications are presented in TABLE III. And there are ten variable loads whose specifications are demonstrated in TABLE IV. There are four measurement devices installed on node 135, 152, 160 and 197, whose simulated values are calculated based on the converged results of distribution load flow by adding Gaussian-distributed random errors to the corresponding true values. The output power characteristics of the two wind power generators are shown in Fig.4. The predicted one-day-ahead PV values and wind power output are shown in Fig.5 and Fig.6, respectively.

TABLE III Specification of DG units

Location	Type	Active power (kW)	Power factor	Deviation
21	Photovoltaic	300	0.95	15%
78	Photovoltaic	300	0.95	15%
91	Wind	450	0.95	15%
101	Wind	450	0.95	15%

TABLE IV SPECIFICATION OF VARIABLE LOADS

Location	Active power with three-phase (kW)	Reactive power with three-phase (kVar)	Deviation (%)
5	[0, 0, 20]	[0, 0, 10]	20%
35	[40, 0, 0]	[20, 0, 0]	15%
38	[0, 20, 0]	[0, 10, 0]	20%
47	[35, 35, 35]	[25, 25, 25]	20%
55	[20, 0, 0]	[10, 0, 0]	25%
58	[0, 20, 0]	[0, 10, 0]	20%
65	[35, 35, 70]	[25, 25, 50]	30%
76	[105, 70, 70]	[80, 50, 50]	25%
100	[0, 0, 40]	[0, 0, 20]	25%
112	[20, 0, 0]	[10, 0, 0]	20%

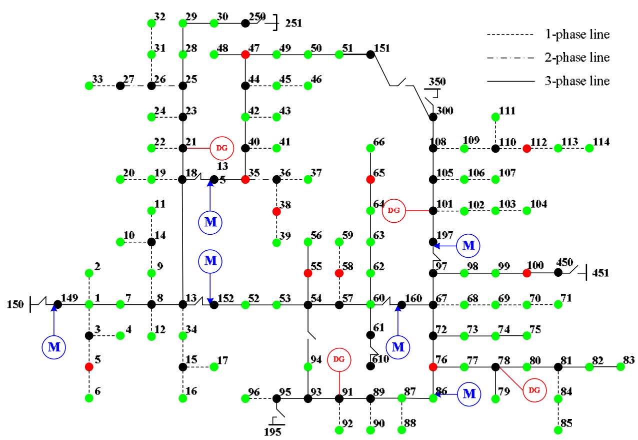

Fig. 3. Schematic of the IEEE 123-bus distribution feeder

TABLE V THE RESULTS FOR THE ACTUAL AND ESTIMATION VALUES OF LOADS

Position	Active power (kW)	
	Real value	Estimation value
5	[0, 0, 20]	[0, 0, 20.01]
35	[40, 0, 0]	[39.99, 0, 0]
38	[0, 20, 0]	[0, 20.01, 0]
47	[35, 35, 35]	[34.97, 34.96, 35.97]
55	[20, 0, 0]	[19.98, 0, 0]
58	[0, 20, 0]	[0,20.05,0]
65	[35, 35, 70]	[35.07, 35.08, 70.08]
76	[105, 70, 70]	[104.29, 69.25, 69.20]
100	[0, 0, 40]	[0, 0, 40.01]
112	[20, 0, 0]	[20.03, 0, 0]
Computation time (s)	281	

According to the proposed Co-PSO algorithm, the simulation results of the proposed algorithm for estimation of loads and DG units are illustrated in TABLE V and TABLE VI. As shown in TABLE V, the three-phase estimation values of the loads about active power and reactive power are all listed, and the difference between real value and estimation value is very small. Since the DG units are with fixed power factor, the estimated reactive power values have not been shown in TABLE VI.

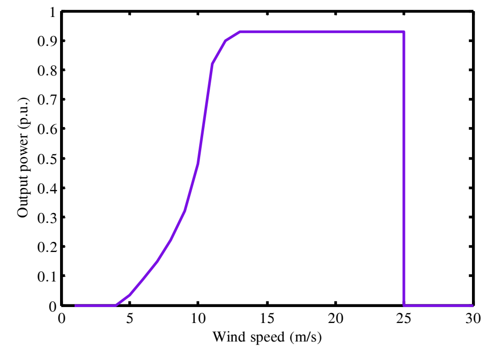

Fig. 4. The output power characteristics according to wind speed

TABLE VI THE RESULTS FOR THE ACTUAL AND ESTIMATION VALUES OF DG UNITS

Position	Active power (kW)	
	Real value	Estimation value
21	100*3Φ	99.93*3Φ
78	100*3Φ	99.37*3Φ
91	150*3Φ	148.81*3Φ
101	150*3Φ	149.90*3Φ

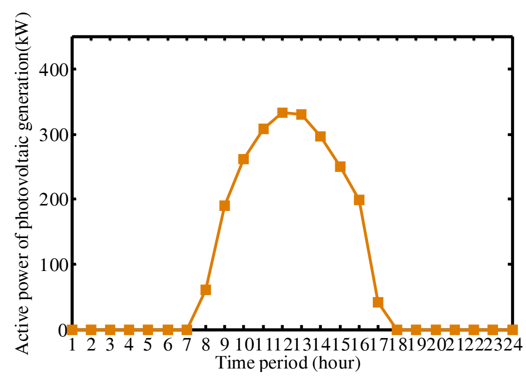

Fig. 5. The forecasted PV value

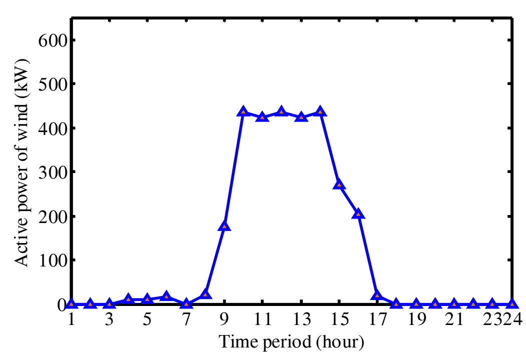

Fig. 6. the forecasted wind power output

C. *Comparison with different algorithms*

In order to quantify the error from different algorithm, the relative error and absolute error are defined as follows:

Maximum Individual Relative Error:

$$MIRE(\%) = \max(|X_{est}(i) - X_{true}(i)| / |X_{true}(i)|) \times 100, \quad i \in N_m \tag{8}$$

Maximum Individual Absolute Error:

$$MIAE(\%) = \max(|X_{est}(i) - X_{true}(i)|), \; i \in N_m \tag{9}$$

TABLE VII ERROR COMPARISONS FOR ESTIMATED VARIABLE LOADS

Algorithm	MIRE (%)		MIAE (%)	
	Value	Location	Value (kW)	Location
PSO	2.63/4.15/4.15	76a/76b/76c	2.76/2.90/2.91	76a/76b/76c
GA	18.39/13.37/11.18	55a/58b/76c	10.18/7.46/7.83	76a/76b/76c
GA-PSO	3.32/5.39 /5.49	76a/76b/76c	3.48/3.77/3.84	76a/76b/76c
MPSO	2.85/3.43/4.24	55a/58b/100c	0.62/0.69/1.70	35a/58b/100c
SA-PSO	2.03/3.40/3.43	76a/76b/76c	2.13/2.38/2.40	76a/76b/76c
Chaos-PSO	1.51/2.28/2.08	55a/76b/76c	1.52/1.60/1.46	76a/76b/76c
Co-PSO	0.67/1.08/1.15	76a/76b/76c	0.71/0.75/0.80	76a/76b/76c

where $X_{est}(i)$ and $X_{true}(i)$ are the estimated and actual values of ith measurement, respectively, N_m is the total number of measurement.

TABLE VII and TABLE VIII show the simulation results about MIRE and MIAE for different algorithms. The three-phase values and their locations of maximal error are listed from column 2 to column 5. The three elements at the location column denote the node number and phase corresponding to the maximal error. The maximal error may appear on different node and different phase toward different DSE algorithm. Taking MPSO DSE algorithm as one example, the maximal relative individual error with different phases are on node 55 with 2.85%, node 58 with 3.43%, and node 100 with 4.24% respectively. Since the real value of the measurement on different node is different, the two location sets of MIRE and MIAE are different. For example, the A phase' MIRE is on node 55, while the MIAE for A phase is on node 35. As shown in TABLE VII, the MIRE of Co-PSO algorithm is better than any other algorithms. And the MIAE sum of three-phase is the minimal. So the proposed Co-PSO is the best algorithm for the estimated variable loads among the seven algorithms.

TABLE VIII ERROR COMPARISONS FOR ESTIMATED DG FOR SEVEN ALGORITHMS

Algorithm	MIRE (%)		MIAE (%)	
	Value	Location	Value (kW)	Location
PSO	1.75	78	1.75	78
GA	14.07	78	14.07	78
GA-PSO	2.76	78	2.76	78
MPSO	1.41	101	2.11	101
SA-PSO	1.53	78	1.53	78
Chaos-PSO	1.26	78	1.26	78
Co-PSO	0.63	78	0.63	78

TABLE IX TIME COMPARISONS FOR THE SEVEN ALGORITHMS

Algorithm	Time (seconds)
PSO	195
GA	278
GA-PSO	283
MPSO	221
SA-PSO	563
Chaos-PSO	776
Co-PSO	281

As shown in TABLE VIII, the MIRE and MIAE for the proposed Co-PSO algorithm are less than that of other algorithms. Since simultaneous three-phase control DG modeling is utilized in this paper, the MIRE and MIAE of the three-phase values are the same. The comparison about computation time is shown in TABLE IX. It shows that the proposed Co-PSO can get better performance than GA-PSO, SA-PSO and Chaos-PSO. Although simple PSO, GA and MPSO can perform fast, their MIRE and MIAE are very high, which can be shown in TABLE VII and TABLE VII.

V. CONCLUSIONS

The distribution state estimation has been influenced a lot by the DG integration, unbalanced line and loads. The Co-PSO DSE algorithm has been proposed to solve the unbalanced three-phase DSE problem in this paper. The Co-PSO makes the local searching on every one dimension state variable and the global searching on all the state variables. The experiments results and comparisons have shown that the proposed algorithm has higher computation accuracy and stable searching performance. The Co-PSO shows very competitive performance to original PSO, GA and multiple-start PSO, simulated annealing PSO and Chaos-PSO considering the computation expense, the number of function evaluations and errors for estimated values.

The power flow analysis in the proposed state estimation mainly uses the load forecasting data to make state estimation. Through load forecasting, the data for all nodes of the network can be got. Since power flow computation convergence problem is another topic, power flow analysis in this paper is assumed to be always convergent on the data.

REFERENCES

[1] F. C. Schweppe, J. Wildes. Power system Static-state estimation, Part I-III, IEEE Trans. on PAS, 1970, 89(1): 120-135.

[2] M. E. Baran, A. W. Kelly. State estimation for real-time monitoring of distribution systems. IEEE Trans. on Power Systems, 1994, 9(3):1601-1609.

[3] A. Abur, A. G. Exposito, Power system state estimation. New York: Marcel Dekker, 2004.

[4] E. Caro, A. J. Conejo, State estimation via mathematical programming: a comparison of different estimation algorithms, IET Generation, Transmission & Distribution, 2011, 6(6), 545-553.

[5] J. Kennedy and R. Eberhart, Particle swarm optimization, IEEE International Conferences on Neural Networks, 1999, vol.4, pp.1942-1948.

[6] F. van den Bergh, A. P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Trans. on Evol. Comput., 2004, 8(3): 225-239.

[7] D. C. Huynh, M.W. Dunnigan, Parameter estimation of an induction machine using advanced particle swarm optimisation algorithms, IET Electric Power Applications, 2010, 4(9): 748-760.

[8] W. H. Kersting, Radial distribution test feeders, IEEE Trans. on Power Syst., 1991, 6(3): 975–985.