Optimal Placement of PMUs in Power Systems Using Mixed Integer Non Linear Programming and Bacterial Foraging Algorithm

Kazem Mazlumi[1] and **Hesan Vahedi**[2], Non-members

ABSTRACT

This paper aims to optimize the PMU (Phasor Measurement Unit) placement for a full observation of the power network and the minimum number of PMUs. In this paper, competition of Mixed Integer Non-Linear Programming and heuristically algorithms such as Bacterial Foraging Algorithm is presented. The results are demonstrated with PMU placement optimization simulation and a redundancy measurement analysis by using IEEE14-bus and Tehran Regional electric company 41-bus networks.

Keywords: Optimal Placement, PMU, Bacterial Foraging Algorithm, MINLP

1. INTRODUCTION

Phasor Measurement Unit (PMU) was introduced in 1990s. It provides synchronized measurements of real-time phasors of bus voltages and branch currents. Synchronicity among PMUs is achieved by same-time sampling of the voltage and the current waveforms using a common synchronizing signal from the Global Positioning System (GPS). Obtaining phasors from different buses of a power system with the same time-space can improve performances of monitored control system in various fields of application, such as state estimation and stability analysis [1].

State estimation is one of the most important functions in the power system monitoring. It provides the platform for advanced security monitoring applications, such as contingency analysis and optimal power flow. Traditionally, state estimation is accomplished by the state estimator in the control center based on the measurements received from SCADA system. These measurements are commonly provided by the Remote Terminal Units (RTUs) at the substations and include the real/reactive power flows, power injections and magnitudes of bus voltages and branch currents. Due to non-linear relation between the measurements and system state variables, the state estimation becomes non-linear, as such, iterative calculation is required for finding the convergent point. This process has high computational burden and sometimes it fails to converge [2]. With the introduction of PMU and its capability for metering bus voltage and branch current phasors with high accuracy, the state estimation becomes linear and therefore, its speed and accuracy will be improved. This improvement for power system monitoring opens the way to Wide Area Monitoring System (WAMS), and has the great interest of literature [3]-[5].

The first step for the state estimation is gathering measurement data from substations. These measurements must be sufficient so as to make the system observable and the state estimation could be performed. If all buses of a system are placed with PMUs the system is completely observable and do not need any more calculation. However, a ubiquitous placement of PMUs is rarely conceivable, due to cost or non-existence of communication facilities in some substations. Hence, the problem is to find out the minimum number of PMUs so that the network becomes completely observable. Several works have been done in this topic [6-8]. The problem is first addressed in [6]. The authors used a modified bisecting search and simulated annealing method based on topological observability. In [7], the Integer Programming based on network observability and the cost of PMUs has been applied to the case of the mixed measurement set which has been implemented to find the PMU placement. Reference [8] proposed PMU placement via tree search method considering complete and incomplete observability. These papers find the minimal buses where PMUs should be installed such that the power network is observable. It is emphasized that the observability analysis is the main component of the optimal placement. Therefore, it must be performed so that maximum utilization of existing data is accomplished and by using a powerful optimization tool, the really minimum number of PMUs is derived. In this paper, Mixed Integer Non Linear Programming (MINLP) and Bacterial Foraging Algorithm (BFA) are considered for optimal placement of PMUs. The Integer Programming based on network

Manuscript received on July 28, 2010 ; revised on August 31, 2010.

This paper is extended from the paper presented in ECTI-CON 2010.

[1,2] The authors are with Electrical Engineering Department, Zanjan University , E-mail: kmazlumi@znu.ac.ir and h_vahedi@znu.ac.ir

observability is used for observability assessment of each candidate configuration [7].

The paper presents the competition of two optimization methods. The first optimization algorithm is the BFA that is developed in a MATLAB code and for the second one GAMS is used with MINLP.

2. PMU PLACEMENT PROBLEM FORMULATION

A PMU placed at a given bus is capable of measuring the voltage phasor of the bus as well as the phasor currents for all lines incident to that bus. Thus, the entire system can be made observable by placing PMUs at strategic buses in the system. The objective of the PMU placement problem is to accomplish this task by using a minimum number of PMUs. In [7], the problem is formulated and solved as an Integer Programming problem as shown below.

For an n-bus system, the PMU placement problem can be formulated as follows:

$$\begin{cases} \min \sum_{i}^{n} w_i x_i \\ st.,\ f(x) \geq \hat{1} \end{cases} \tag{1}$$

Where x is a binary decision variable vector and i is the bus number, whose entries are defined as:

$$x_i = \begin{cases} 1 & \text{if a PMU is installed at bus i} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

w_i is the cost of the PMU installed at the i^{th} bus and $f(x)$ is a vector function, whose entries are nonzero if the corresponding bus voltage is solvable using the given measurement

$\hat{1}$ is a vector whose entries are all ones. set and zero otherwise.

The expressions for the nonlinear constraints are formed based on the knowledge about the locations and types of existing measurements. Given a PMU at a bus, it is assumed that the bus voltage phaser and all current phasers along the lines connected to that bus will be available. This also implies that the bus voltage, along with all adjacent bus voltages will also be available (solvable).

The procedure of providing the constraint equations is described for the three possible cases: (1) only PMU measurements, (2) PMU measurements and injections (they may be zero injections or measured injections) and (3) PMU measurements, injections and flows. The description of the procedure for each case will be given using a small 7-bus tutorial example for clarification. However, the entire procedure is actually programmed and successfully tested on different size systems with diverse measurement configurations in [7]. Consider the 7-bus system and its measurement configuration shown in Fig 1.

Case 1: A system which has no conventional measurements and/or zero injections. First, form the binary connectivity matrix A. The entries of A are defined as follows:

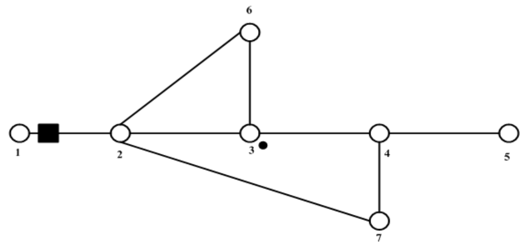

Fig.1: *7-Bus Test System*

$$A_{k,m} = \begin{cases} 1 & \text{k=m} \\ 1 & \text{k and m are connected} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Matrix A can be directly obtained from the bus admittance matrix by transforming its entries into binary form. The matrix of 7-bus system of Fig.1 as shown below:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The constraints for this case can be formed as:

$$f(x) = \begin{cases} f_1 = x_1 + x_2 & \geq 1 \\ f_2 = x_1 + x_2 + x_3 + x_6 & \geq 1 \\ f_3 = x_2 + x_3 + x_4 + x_6 + x_7 & \geq 1 \\ f_4 = x_3 + x_4 + x_5 + x_7 & \geq 1 \\ f_5 = x_4 + x_5 & \geq 1 \\ f_6 = x_2 + x_3 + x_6 & \geq 1 \\ f_7 = x_3 + x_4 + x_7 & \geq 1 \end{cases} \tag{5}$$

The operator "+" serves as the logical "OR" and the use of 1 in the right hand side of the inequality ensures that at least one of the variables appearing in the sum is non-zero.

For example, consider the constraints associated with the 1^{st} bus and the 2^{st} bus as given below:

$$f(x) = \begin{cases} f_1 = x_1 + x_2 & \geq 1 \\ f_2 = x_1 + x_2 + x_3 + x_6 & \geq 1 \end{cases} \tag{6}$$

The first constraint $f_1 \geq 1$ implies that at least one PMU must be placed at either one of buses 1 or 2 (or both) in order to make bus 1 observable. Similarly, the second constraint $f_2 \geq 1$ indicates that at least one PMU should be installed at any one of the buses 1, 2, 3, 6, or 7 in order to make bus 2 observable. Case 2 and Case 3 are completely explained in [7], respectively.

3. BACTERIAL FORAGING ALGORITHM

Natural selection tends to eliminate animals with poor foraging strategies and favor the propagation of genes of those animals that have successful foraging strategies since they are more likely to enjoy reproductive success. After many generations, poor foraging strategies are either eliminated or shaped into good ones. The E. coli bacteria that are present in our intestines also undergo a foraging strategy. The control system of these bacteria that dictates how foraging should proceed can be subdivided into four sections namely Chemotaxis, Swarming, Reproduction and Elimination and Dispersal [9].

3.1 Chemotaxis

This process is achieved through swimming and tumbling via Flagella. Depending upon the rotation of Flagella in each bacterium, it decides whether it should move in a predefined direction (swimming) or altogether in different directions (tumbling), in the entire lifetime. To represent a tumble, a unit length random direction, say φ , is generated; this will be used to define the direction of movement after a tumble in a tumble, the position of the i_{th} bacterium is updated as:

$$\theta^i = (j+1,k,l) = \theta^i(j,k,l) + C \times \angle\phi \tag{7}$$

Where θ^i is the position of the i_{th} bacterium at the j_{th} chemotactic step of the k_{th} reproduction loop in the l_{th} elimination-dispersal event, C is the size of the step taken in the random direction specified by the tumble, $\angle\varphi$ is the angle of the direction which is randomly generated in the range of $[0,\, 2\pi]$.

3.2 Cell-to-cell Communications

E-coli bacterium has a specific sensing, actuation and decision-making mechanism. As each bacterium moves, it releases attractant to signal other bacteria to swarm towards it. Meanwhile, each bacterium releases repellent to warn other bacteria to keep a safe distance from it. BFA simulates this social behavior by representing the combined cell-to-cell attraction and repelling effect as:

$$\begin{aligned} J_{cc}(\theta^i(j,k,l)\theta(j,k,l)) &= \sum_{t=1}^{S} J_{cc}^t(\theta^i,\theta\theta) \\ &= \sum_{t=1}^{S}\left[-d_{attract}\exp(-\omega_{attract}\sum_{m=1}^{p}(\theta_m^i-\theta_m^t)^2)\right] \qquad (8)\\ &+ \sum_{t=1}^{S}\left[-d_{repellant}\exp(-\omega_{repellant}\sum_{m=1}^{p}(\theta_m^i-\theta_m^t)^2)\right] \end{aligned}$$

Where $j_{cc}(\theta^i,\theta)$ is the cost function value to be added to the actual cost function to be minimized to present a time varying cost function. 'S' is the total number of bacteria and 'P' the number of parameters to be optimized which are present in each bacterium. $d_{attract}$, $\omega_{attract}$, $d_{repellent}$, $\omega_{repellent}$ are different coefficients that are to be chosen properly.

3.3 Reproduction

In BFA, a fixed total number of reproduction steps, N_{re} is given. Only the first half of populations survive in each reproduce step a surviving bacterium splits into two identical ones, which occupy the same position in the environment as the one in previous step. Thus the population of bacteria keeps constant in each chemotactic step. After N_c chemotactic steps, the fitness values for the i_{th} bacterium in the chemotactic loop are accumulated and calculated by:

$$j_{health}^i = \sum_{j=1}^{N_c+1} j^i(j,k,l) \tag{9}$$

Where j_{health}^i presents the health of the i_{th} bacterium. The smaller the j_{health}^i is, the healthier the bacterium is. To simulate the reproduction character in nature and to accelerate the swarming speed, all the bacteria are sorted according to their health values in an ascending order and each of the first $S_r(S_r = S/2)$, for convenience S is assumed to be a positive even integer) bacteria splits into two bacteria. The characters including location and step length of the mother bacterium are reproduced to the children bacteria. Through this selection process the remaining S_r unhealthier bacteria are eliminated and discarded. To simplify the algorithm, the number of the bacteria keeps constant in the whole process.

3.4 Elimination-dispersal

For the purpose of improving the global search ability, elimination-dispersal event is defined after N_{re} steps of reproduction. The bacteria are eliminated and dispersed to random positions in the optimization domain according to the probability P_{ed}. This elimination-dispersal event helps the bacterium avoid being trapped into local optima. The number of the event is denoted as N_{ed}. The flowchart of the algorithm is shown in Fig. 2.

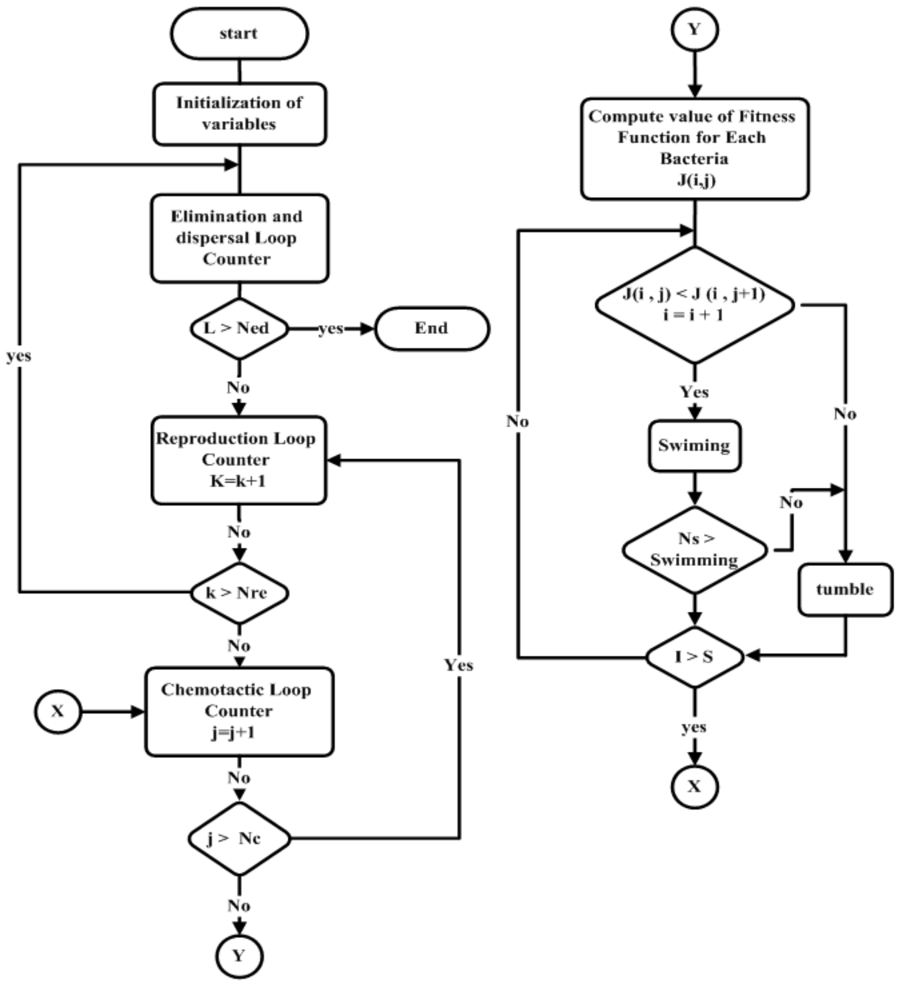

Fig.2: *Flowchart of the bacteria foraging algorithm*

4. GAMS AND MILNP SOLVER

The years since the 1950s have seen the rapid development of algorithms and computer codes to analyze and solve large mathematical programming problems. One of the important parts of this growth was the development in the early 1980's of modeling systems. One of the earlier of which was the Generalized Algebraic Modeling System (GAMS). GAMS is a high-level modeling system for mathematical programming problems and Modeling linear, nonlinear and mixed integer optimization problems [10].
Models with objective functions are:

1. LP for Linear Programming;
2. NLP for Nonlinear Programming;
3. MIP for Mixed Integer Programming;
4. MINLP for Mixed Integer Non Linear Programming;
5. RMIP for Relaxed Mixed Integer Programming;
6. MIQCP for Mixed Integer Quadratically Constrained Program;
7. DNLP for Nonlinear Programming with Discontinuous Derivatives;
8. MPEC for Mathematical Program with Equilibrium Constraints.

In this paper, authors used MINLP solver. Mathematically, the MINL problem looks like:

$$\begin{gathered} \text{Maximize or Minimize} f(x) + d(y) \qquad (10) \\ g(x) + h(y)\ \alpha\ 0 \\ L \quad < x > \quad U \\ y = \{0, 1, 2, \ldots\} \end{gathered}$$

Where x is a vector of variables that are continuous real numbers, $f(x) + d(y)$ is the objective function, $g(x)+h(d)$ is represents the set of constraints, is some mixture of $=$, $=$ and $=$ operators and L and U are vectors of lower and upper bounds on the variables. Names of the solvers that can be used on the MINLP problem class are shown in TABLE I [10].

5. SIMULATION RESULTS

The PMU placement algorithm presented before is tested on IEEE 14-bus and Tehran Regional Electric Company 41-bus networks that shown in Fig. 3 [11, 12]. At first, IEEE 14-bus test system was considered. This system has got only one zero-injection bus. The initial placement results in installing 5 PMUs and the output of the BFA algorithm is only 3 PMUs. For the Second case, Tehran Regional Electric Company 41-bus network is considered. This system has got two zero-injection buses. Results in installing 28 PMUs and the output of the BFA algorithm are only 11 PMUs.

For the IEEE 14-bus system, we have considered $S = 4$, $p = 14$, $N_c = 5$, $N_s = 20$, $N_{re} = 100$, $N_{ed} = 2$, $P_{ed} = 0.25$ and for the Tehran Regional Electric Company 41-bus network we have considered $S = 10$, $p = 41$, $N_c = 20$, $N_s = 20$, $N_{re} = 100$, $N_{ed} = 4$, $P_{ed} = 0.25$ and swarm signal weren't implemented in these cases. Fig. 4 shows the convergence rate of BFA for finding the optimal placement point of the two networks.

Finally MINLP was used to optimize the PMUs placement of the two networks. It is necessary to mentioned that GAMS contains lots of solvers that are capable of solving optimization programs using MINLP [10]. System specifications of two cases are shown in Table II. The results of some solvers are shown in TABLE III. It should be mentioned that the corresponding configuration of the required number of PMUs is not necessarily unique. However, from the observability point of view, there is no difference between different configurations with the same number of PMUs, so in this paper, only one configuration per each case is presented. In the 1st and the 2nd network there are no conventional power flow and injection measurements installed in the system. The numbers of zero injections are 1 and 2, respectively.

Table 1: GAMS Solvers

	CNS	DNLP	LP	MCP	MINLP	MIP	MIQCP	MPEC	NLP	QCP
ALPHAECP					I		I			
BARON		I	I		I	I	I		I	I
BDMLP			I			I				
COINOS		I	I		I	I	I		I	I
COINSCIP			I			I				
CONOPT	I	I	I						I	I
DECISC			I							
DICOPT					I		I			
KNITRO		I	I						I	I
LGO		I	I						I	I
LINDOGLOBAL		I	I		I	I	I		I	I
MILES				I						
MINOS		I	I						I	I
MOSEK		I	I			I	I		I	I
MPSGE										
MSNLP		I							I	I
NLPEC				I				I		
OQNLP		I			I		I		I	I
OSL			I			I				
OSLSE			I							
PATH	I			I						
SBB					I		I			
SNOPT		I	I						I	I
XA			I			I				

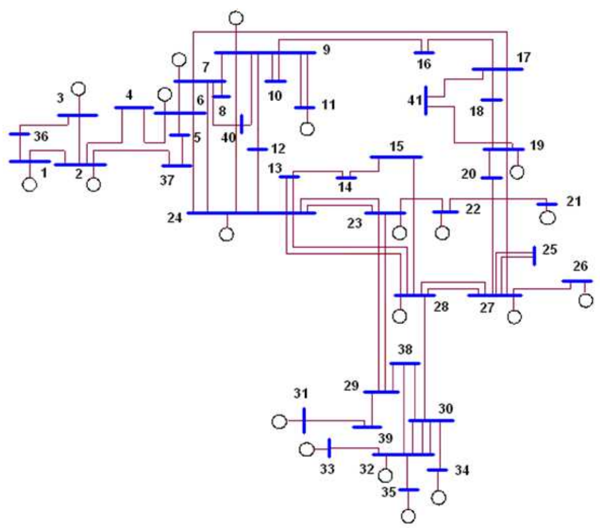

Fig.3: Tehran 41-bus System

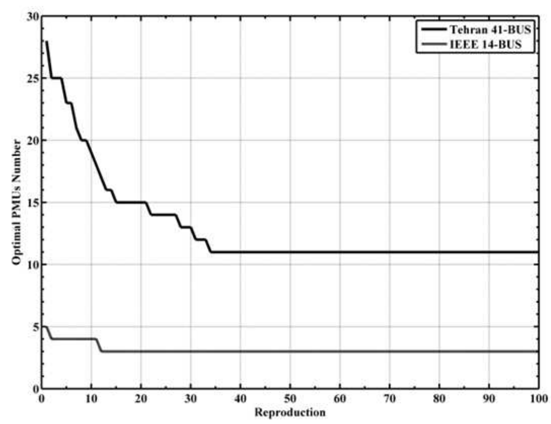

Fig.4: Convergence Rate of BFA

Table 2: *System Specification*

Case	Power System	Zero Injection	Bus No.	Flow Meas.
1	IEEE 14-BUS	1	7	0
2	Tehran 41-BUS	2	10,25	0

Table 3: *Test Results*

Network	Solver		PMU's Location	No. of PMUs
Net.1	BFA		2,6,9	3
	MINLP	ALPHAECP	2,6,9	3
		BARON	2,6,9	3
		DICOPT	2,6,9	3
		OQNLP	2,6,9,10	4
		SBB	2,6,9	3
Net.2	BFA		6, 7, 8, 9, 11, 13, 23, 24, 26, 29, 30	11
	MINLP	ALPHAE CP	6,7 , 8, 9, 11, 13, 17, 23, 24, 29, 30	11
		BARON	6, 7, 8, 9, 11, 13, 23, 24, 26, 29, 30	11
		DICOPT	6, 7, 8, 9, 11, 13, 17, 23, 24, 29, 30	11
		OQNLP	4, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 21, 22, 23, 24, 25, 27, 29, 31, 32, 33, 34, 36, 37, 38, 39, 41	27
		SBB	6, 7, 8, 9, 11, 13, 23, 24, 26, 29, 30	11
			Execution Time (sec)	
Net. 1,2	BFA		2 - 3	
	MINLP		0.016 – 0.017	

6. CONCLUSION

An integer programming based formulation and the associated solution to the problem of PMU placement in power systems has been used in this paper. Numerical results are given for different size systems where a minimum number of PMUs are placed with and without other conventional measurements. As the power systems become more populated by PMUs, the presented approach may assist the system planners in deciding on

References

[1] A. G. Phadke, J. S. Thorp, and K. J. Karimi, "State Estimation with Phasor Measurements," *IEEE Transactions on Power Systems*, Vol. 1, No. 1, pp. 233-241, February 1986.

[2] A. J. Wood, and B. F. Wollenberg, *Power Generation, Operation and Control*, New York: Wiley, 1996.

[3] M. Zima, M. Larsson, P. Korba, c. Rehtanz, and G. Andersson, "Design aspect for wide-area monitoring and control systems," *Proceeding of. 2005IEEE*, vol. 93, pp. 980-996.

[4] M. Begovic, D. Novosel, D. Karlsoon, C. Henville, G. Michel, "Widearea protection and emergency control," *Proceeding of. 2005 IEEE*, vol. 93, pp. 876-891.

[5] M. G. Adamiak, A. P. Apostolov, M. Begovic, C. F. Henville, K. E. Martin, G. L. Michel, A. G. Phadke, and J. S. Thorp, "Wide-area protection_technology and infrastructure," *IEEE Trans. Power Delivery*, vol. 21, pp. 601-609, Apr. 2006.

[6] T. L. Baldwin, L. Mili, M. B. Boisen, R. Adapa, "Power system observability with minimal phasor measurement placement," *IEEE Trans. Power System*, vol. 8, pp. 707-715, May. 1993.

[7] B. Xu, A. Abur, "Observability Analysis and measurement placement for system with PMUs," in *Proc. 2004 IEEE Power System Conf. Expo*, vol. 2, pp. 943-946.

[8] R. F. Niqui, and A. G. Phadke, "Phasor measurement unit placement techniques for complete and incomplete observability," *IEEE Trans. Power Delivery*, vol. 20, pp. 2381-2388, Oct. 2005.

[9] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp.52-67, Jun. 2002.

[10] B. A. McCarl, A. Meeraus, P. V. D. Eijk, M. Bussieck, S. Dirkse, P. Steacy, McCarl Expanded GAMS User Guide Version 22.9, 2008, pp. 42. Available Online: http://www.gams.com.

[11] Power Systems Test Case Archive, Available Online at: www.ee.washington.edu/research/pstca/

[12] Mazlumi, K.; Abyaneh, H.A.; Sadeghi, S.H.H.; Geramian, S.S.; "Determination of optimal PMU placement for fault-location observability," *IEEE Electric Utility Deregulation and Restructuring and Power Technologies*, 2008. DRPT 2008.

Kazem Mazlumi was born in Tehran, Iran in 1976. He received the B.S. degree in electrical engineering from Amirkabir University of Technology, Tehran, Iran, in 2000, the M.S. degree from Sharif University of Technology, Tehran, Iran, in 2003, and the Ph.D. degree from Amirkabir University of Technology, Tehran, Iran, in 2009. He is currently an assistant professor of electrical engineering at Zanjan University, Zanjan, Iran. His current research interests include power system protection, reliability and power quality.

Hesan Vahedi was born in Tehran, in 1985. He received his B.S. and M.S. degrees in electrical engineering in 2007 and 2010 from Islamic Azad University of Saveh, Saveh, Iran and Zanjan University (ZNU), Zanjan, Iran, respectively, graduating with the First Class Honor. He is currently working toward the Ph.D. degree in electrical engineering at the Amirkabir University of Technology (AUT), Tehran, Iran. His research interests include control systems, power electronic, applications of power electronics in power systems, distributed generation, power quality and artificial intelligent.