Journal of Advances in Computer Research
Quarterly ISSN: 2008-6148
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 4, No. 1, February 2013), Pages: 39-53
www.jacr.iausari.ac.ir

An Improved Shuffled Frog Leaping Algorithm for Simultaneous Design of Power System Stabilizer and Supplementary Controller for SVC

Leila Esteki[1]✉, Maryam Shahriari-kahkeshi[2], Ehsan Bijami[3]

(1) Dep. of Electrical Engineering, Shahr-e-Kord Branch, Islamic Azad University, Shahr-e-Kord, Iran
(2) Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran.
(3) Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran.

l_esteki@iaushk.ac.ir, m.shahriyarikahkeshi@ec.iut.ac.ir, e.bijami@ec.iut.ac.ir

Received: 2012/07/15; Accepted: 2012/08/06

Abstract

This paper presents a new Modified Shuffled Frog Leaping Algorithm (MSFLA) applied to design simultaneous coordinated tuning of damping controllers to damp the power system low frequency oscillations. For this, a new frog leaping rule is proposed to improve the local exploration and performance of the original SFLA and the genetic mutation operator is employed for new frog generation instead of random frog generation to improve the performance and quicker algorithm convergence. In order to verify the effectiveness of the proposed method, a 2-area-4-machine and a 5-area-16-machine power system are considered which two power system stabilizers (PSSs) are designed coordinately for the first system and one PSS for a generator and one supplementary controller for a Static Var Compensator (SVC) are designed simultaneously for the second system. To show the effectiveness of the designed controllers, study systems are tested under two different operating conditions and simulation studies are presented.

Keywords: *Shuffled Frog Leaping Algorithm, Low frequency Oscillation, SVC*

1. Introduction

It is known that the Power System Stabilizers (PSSs) and the supplementary controllers of Flexible AC Transmission System (FACTS) devices are efficient tools for improving the stability of power systems through damping of low-frequency modes in the order of 0.2 to 2.5 Hz [1], [2]. Many control strategies, such as optimal control, robust control and adaptive control have been proposed to the PSS design problem. The works carried out in [3]-[7] are examples of such applied techniques. Among these, conventional PSS (CPSS) of the lead-lag compensation type is used by most utility companies because of its simple structure, flexibility and ease of physical realization [2].

In the last decade, meta-heuristic optimization techniques have gained an incredible recognition as the solution method for such type of designing damping controller problems. The Particle Swarm Optimization (PSO) method is one of the most used algorithms to optimal design of PSSs [8]. The authors in [9] presented an implementation using a genetic algorithm to seek the PSSs parameter. In [10], a procedure employing simulated annealing and particle swarm optimization is proposed to search for the solution. Some other heuristic optimization techniques, such as Immune Algorithm (IA) [11], AINet algorithm [12], Shuffled Frog Leaping Algorithm (SFLA) [13], Imperialist Competitive Algorithm (ICA) [14], Artificial Bee Colony (ABC) algorithm [15] and Harmony Search Algorithm (HSA) [16] were applied to design power system damping controllers.

This paper presents an alternative method to design stabilizing signals in power system based on an improved real version of Shuffled Frog Leaping Algorithm (SFLA). In fact since there are some shortcomings with the original SFL algorithm, a Modified SFL Algorithm (MSFLA) is introduced in this paper. To show the effectiveness, robustness and the optimization velocity of proposed MSFLA, numerical results are presented on two study systems: a 2-area 4-machine system by designing two PSSs and a 5-area 16-machine system by designing a PSS for a generator and a supplementary controller for the SVC.

The effectiveness and robustness of the designed damping controllers are illustrated by considering various operating conditions. The main Innovation of the paper are as follows: (i) presenting a modified SFL algorithm, (ii) introducing a new optimization formulation and cost function to design the robust coordinated PSS and supplementary controller for the SVC and finally proposing a new approach for power system stabilization.

The paper is organized as follows: to make a proper background, the basic concept of the original SFLA and proposed modified SFLA are explained in Section 2. Section 3 describes the study systems. In section 4, the design problem is formulated as a multi-objective optimization problem and the simulation results of the SFLA and MSFLA on two case studies are given in Section 5. Finally, Section 6 concludes paper.

2. Original SFL algorithm and proposed MSFL algorithm

SFLA is a population based optimization algorithm inspired from the memetic evolution of a group of frogs when searching for food and proven its superior capabilities, such as faster convergence and better global minimum achievement [17]. The original SFLA and proposed MSFLA are explained below.

2.1 Original SFL Algorithm Overview

The SFL is derived from a virtual population of frogs in which individual frogs represent a set of possible solution. The term frog in SFLA is similar to chromosome in genetic algorithm. Each frog is distributed to a different subset of the whole population described as memeplexes. Different memeplexes are considered as different culture of frogs that are located at different places in a solution space (i.e. global search). Each culture of frogs performs simultaneously an independent deep local search using a particle swarm optimization like method. To ensure global exploration, after a pre-defined number of memeplex evolution steps (i.e. local search iterations), information is passed among memeplexes in a shuffling process. Shuffling improves frog ideas quality after being infected by the frogs from different memeplexes; ensure that the cultural evolution towards any particular interest is free from bias. In addition, to improved information, random virtual frogs are generated and substituted in the population if the local search cannot find better solutions. Then a local search and shuffling processes (global relocation) will be continued until convergence is materialized. The flowchart of the SFLA is illustrated in Figure 1.

As it is shown in Figure 1, SFLA begins with an initial population of "N" frogs, $P=\{X_1,X_2,...,X_N\}$ that is created randomly within the feasible space Ω. For S-dimensional problems (S variables), the position of the i^{th} frog is represented as $X_i=[x_{i1},x_{i2},...,x_{is}]^T$. A fitness function is defined to evaluate the frog's position, while the performance value of each frog is computed based on its position and frogs will be sorted in a descending order according to their fitness values. The entire population is divided into m memeplexes, each of which consisting of n frogs (i.e. $N=n\times m$). The division is done with the first frog goes to the first memeplex, the second frog goes to the second memeplex, m^{th} frog goes to the m^{th} memeplex, and the $(m+1)^{th}$ frog backs to the first memeplex, etc. The local search block of Figure 1 is shown in Figure 2.

According to Figure 1, during memeplex evolution, the position of ith frog (D_i) is adjusted based upon the difference between the frog with the worst fitness (X_w) and the frog with the best fitness (X_b) as shown in (1). The worst frog X_w leaps toward the best frog X_b and the position of the worst frog is updated based on the leaping rule, as shown in (2):

$$\text{Position change}\,(\mathrm{D_i}) = rand()\times(X_b - X_w) \tag{1}$$

$$X_w(new) = X_w + D, (\|D\| < D_{\max}) \tag{2}$$

Where $rand()$ is a random number in [0,1] and D_{max} is the maximum allowed change of frog's position in one jump. If this repositioning process produces a frog with a better fitness, it replaces the worst frog, otherwise, the calculation of (1) and (2) are repeated with respect to the global best frog (X_g), (i.e. X_g replaces X_b). If no improvement becomes possible, a new frog within the feasible space is randomly generated to replace the worst frog. Based on Fig. 1, the evolution process is continued until the termination criterion is met. The termination criterion could be the number of iterations or when a frog with an optimum fitness is found [17, 20].

2.2 Proposed Modified SFL Algorithm (MSFLA)

According to the original frog leaping rule, the possible new position of the worst frog is restricted in the line segment between its current position and the best frog's position, and the worst frog will never jump over the best one. As a result, this frog leaping rule limits the local search space during each memeplex evolution step.

Also, according to (1) and (2), the worst frog is only affected by the best frog; therefore the best frog has less chance of evolution during the leaping process. These issues make the algorithm having an insufficient learning mechanism and cause premature convergence and lead the algorithm to be trapped in local optimum easily. To obviate these problems and increase the ability of algorithm in the search space exploration, a new method is presented for local search in the memeplexes.

Modification of the frog leaping rule: In the proposed method, the frog leaping rule and the process of learning by the worst frog from the best frog in a memeplex is defined using the sum of the Minkowski distance of each memeplex member ($X(i)$) from the worst frog (X_{worst}) and the one between the worst frog in the memeplex and the best frog of the whole population, i.e (X_g).

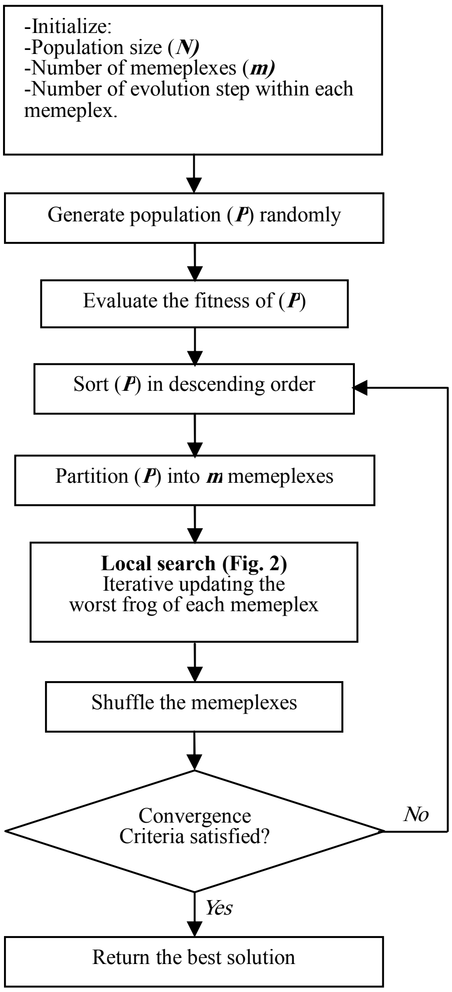

Figure 1. General principle of SFLA

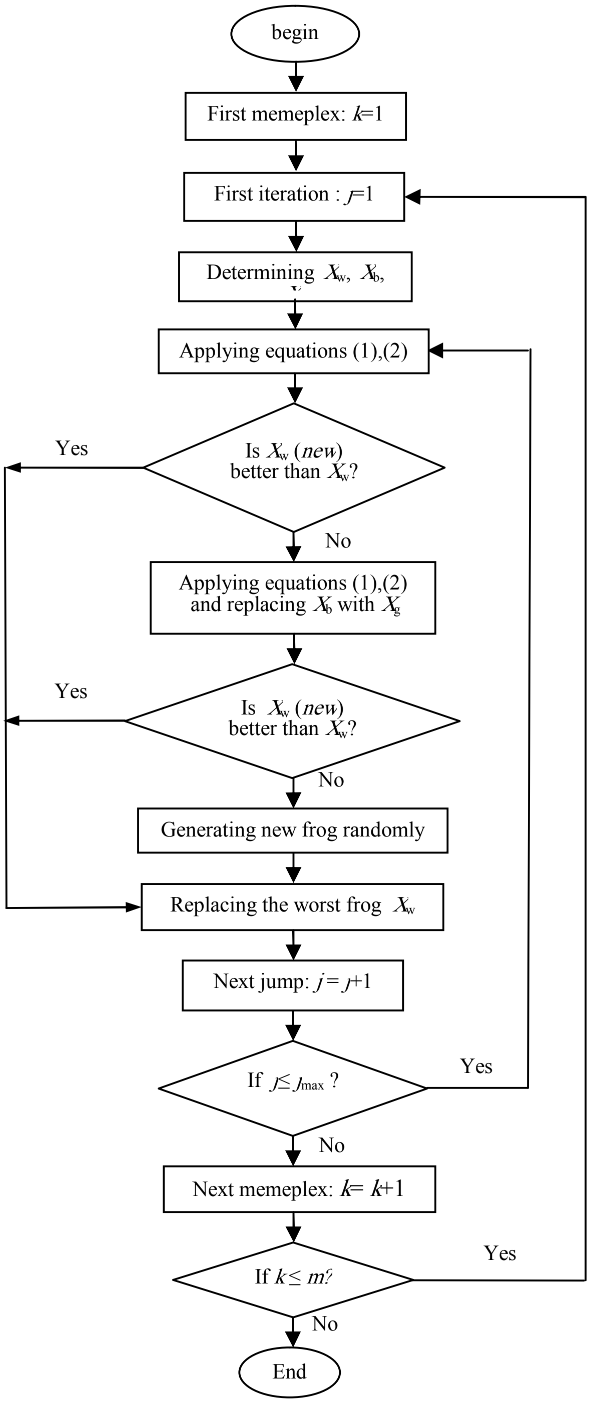

Figure 2. Local search block of Figure 1.

The proposed frog leaping rule is expressed in the following equation.

$$D(i) = rand \times c_1 \times M(X(i), X_{worst}) + c_2 \times (X_g - X_{worst}) + W \tag{3}$$

where, $X(i)$ and X_{worst} are the i^{th} ($i=1,\ldots,n$) and the worst member of each memeplex respectively. Also, $rand$ is a random number with homogenous distribution in the range between 0 and 1. c_1 is learning factor of the worst solution from the rest of the solutions in each memeplex. Also, c_2 is learning factor of the worst solution in a memeplex from the best solution in the whole population which is a random number between 0 and 2. Also, $M(X(i), X_{worst})$ stands for the Minkowski distance between the i^{th} solution from the worst solution in a memeplex. The Minkowski-r-distance between X and Y is defined using the following equation:

$$M(X, Y) = \sqrt[r]{\sum_{j=1}^{L} \left| x_j - y_j \right|^r} \quad (4)$$

where, $X=(x_1, x_2, \ldots x_L)$ and $Y=(y_1, y_2, \ldots y_L)$. Finally, in (3), W is defined using the following equation [20]:

$$W = [r_1 w_{1,\max}, r_2 w_{2,\max}, \ldots, r_S w_{S,\max}]^T \quad (5)$$

in which, r_i $(1 \le i \le S)$ are uniformly distributed random numbers in the interval [-1, 1]. W_{max} is the most allowed understanding and the uncertainty in the i^{th} dimension of the search space, defined as follow:

$$W_{\max} = [w_{1,\max}, w_{2,\max}, \ldots, w_{S,\max}]^T \quad (6)$$

As shown in (3) the learning process of the worst frog from the best frog is not done only in a single memeplex; but the learning process of the worst frog in the proposed strategy is from all of the frogs in the same memeplex and the best frog of the whole population.

The new frog leaping rule prevents trapping of the algorithm in the local optimum especially in more complex problems having high dimensions. Also, the convergence of the algorithm is improved by changing the learning process of the worst frog from the best frog in a memeplex to the best frog in the whole population. Moreover, using the Minkowski distance has caused the algorithm to be more powerful and precise in local search process around the worst frog. According to the new frog leaping rule, the new position of each frog is calculated using the following equation:

$$X_{worst}(new) = X_{worst}(old) + D(i) \quad (7)$$

In (7) if the position change of a frog leads to a better frog then the newly positioned frog will replace the old worst frog; otherwise, X_g will replace the $X(i)$ in (3). If no improvements are made to the worst frog in the new frog generation process in (7), then a frog should be generated and replace the worst frog. In the original SFL algorithm, this new frog is generated randomly.

Using the genetic mutation operator: One of the main drawbacks of the standard SFLA is generating a frog randomly and replacing it with the frog that is not improved. This replacement may lead the algorithm to premature convergence since the algorithm losses the population quality and the potential of those solutions that could reach the global optimum at the last iteration. To overcome the difficulties associated with this mechanism, the mutation operator is used for generating new frogs. Mutation is a powerful strategy which diversifies the SFLA population and improves the SFLA's performance on preventing premature convergence to local minima [22]. In this paper the Cauchy distribution is used for applying the mutation operator to the worst frog. When a solution is chosen to be mutated, each component is then mutated or not with probability $1/S$, where S is the number of components in the vector. On average, one component is mutated. To mutate a component, a number is generated randomly using Cauchy distribution which the probability density function of the Cauchy distribution is given in [22] that is defined by:

$$f(x) = \frac{a}{\pi}\frac{1}{x^2 + a^2} \qquad , \qquad a = 0.2 \tag{8}$$

The reason for using such a mutation operator is to increase the probability of escaping from a local optimum [22]. After generating new frog, X_i is leaped to explore the new positions as follows:

$$D(i) = rand \times c_1 \times M(X_g, X(i)) + W \tag{9}$$

Then, the new position of the frog is obtained based on following equation:

$$X_i(new) = \begin{cases} X_i + D_i & if \|D_i\| \le D_{\max} \\ X_i + \dfrac{D_i}{\sqrt{D_i^T D_i}} D_{\max} & if \|D_i\| > D_{\max} \end{cases} \tag{10}$$

If the repositioning process produces a frog with better fitness, it replaces X_i. Otherwise, the algorithm goes to the next jump and the evolution process is continued until the termination criterion is met. The termination criterion could be the number of iterations or when a frog with minimum fitness is found.

3. Case Studies

Two power systems are used to demonstrate the design of damping controllers, a 2-area-4-machine system and a 5-area-16-machine system:

3.1 2-area-4-machine system

This system is illustrated in Figure 3. The subtransient model for the generators, and the IEEE-type DC1 and DC2 excitation systems are used for machines 1 and 4, respectively. The IEEE-type ST3 compound source rectifier exciter model is used for machines 2 and 3. Two PSSs are going to be designed simultaneously for machines 2 and 3. Details of the system data are given in [23].

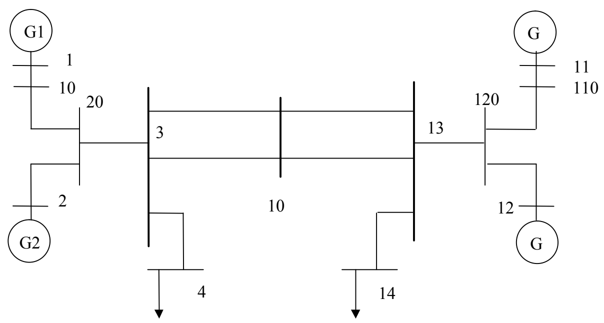

Figure 3. Single-line diagram of a 2-area-4-machine study system.

3.2. 5-area-16-machine system

This system is shown in Figure 4, consisting of 16 machines and 68 buses for 5 interconnected areas. The first nine machines, G1 to G9, constitute the simple representation of Area 1. Next four machines G10 to G13 represent Area 2. The last three machines, G14 to G16, are the dynamic equivalents of the three large neighboring areas interconnected to Area 2. The sub-transient reactance model for the generators, the first-order simplified model for the excitation systems, and the linear models for the loads and ac network are used. Details of the system data are given in [23]. Based on earlier studies in [24], a 546 MVar SVC is placed at bus 1 in the 5-area-16-machine system. The supplementary controller for the SVC and a PSS to be placed in machine 9 are going to be designed simultaneously.

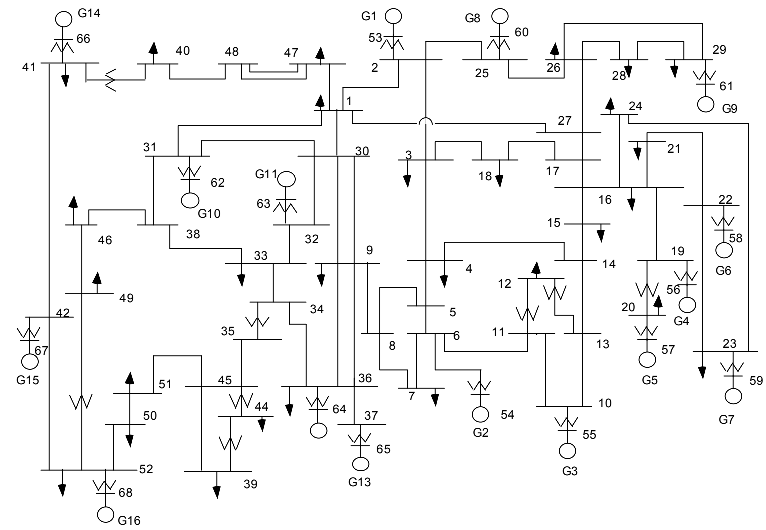

Figure 4. Single line diagram of a 5-area-16-machine study system.

4. Problem Formulation

The structure shown in Figure 5 is used for both the PSS and the supplementary controller, where the generator speed (GS) is considered as input to the PSS and the input to the supplementary controller is the active power flow in line 1-27 for the second system.

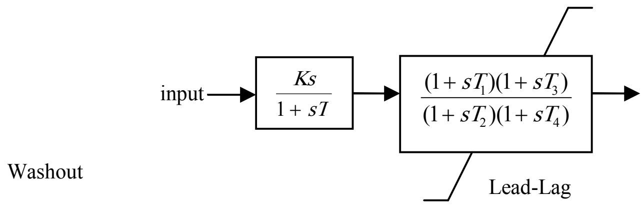

Figure 5. Block diagram for PSS and supplementary controller of SVC.

The PSSs are designed by using the suggested approach in [13]. In this approach, the SFLA and MSFLA are employed to search for optimal settings of PSS parameters using eigenvalue-based objective functions.

The control strategy is to choose the best PSSs parameters i.e. [K, T, T_1, T_2, T_3, T_4; K, T, T_1, T_2, T_3, T_4] in such a manner that the lightly damped and undamped electromechanical modes of all machines are shifted to a prescribed zone in the left-hand side of *S*-plane as far as possible. The tuning of the PSS parameters for a multi-machine power system is usually formulated as an objective function with constraints consisting of the damping factor and damping ratio.

In the first study system (without lose of generality, the procedure can be applied to the second study system), the parameters to be tuned through the SFLA and MSFAL are PSSs parameters; i.e. [$K, T, T_1, T_2, T_3, T_4; K, T, T_1, T_2, T_3, T_4$]. In the SFL algorithms, each population (N frogs) represents a candidate solution for the problem. Thus, each frog is considered as [$K, T, T_1, T_2, T_3, T_4; K, T, T_1, T_2, T_3, T_4$] which determines the parameters of the PSSs. By placing each solution (PSS1 and PSS2) into the study system, related eigenvalues is obtained for the system. For each solution the worst eigenvalue is selected and the corresponding damping ratio (ξ) is calculated.

Now, two vectors are defined as follows: $\underline{\xi} = \{\xi_1, ..., \xi_N\}$ and $\underline{\sigma} = \{\sigma_1, ..., \sigma_N\}$ as those elements are the damping ratio of the worst eigenvalue for each solution and the real parts of the eigenvalues with the damping ratios less than 0.36, respectively. With these two vectors the following objective functions are considered:

$$f_1 = \min_{i \in \{1,...,n\}} (\xi_i) \qquad (11)$$

$$f_2 = \min_{i \in \{1,...,n\}} (-\sigma_i) \qquad (12)$$

and the optimization problem can be formulated as maximize $\{ f_1, f_2 \}$. According to these objectives function, the PSSs are designed so that the damping ratio of the close-loop system is increased as well as shifting the eigenvalues of the close-loop system to the left hand side of *S*-plane. In other words, this fitness function will place the system closed-loop eigenvalues in the *D*-shape sector characterized by $\sigma_i < \sigma_0$ and $\xi_i > \xi_0$ as shown in Figure 6.

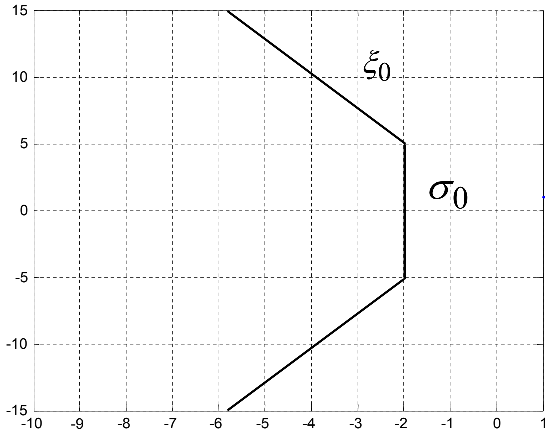

Figure 6. A D-shape sector in the s-plane.

To implement the SFLA a weighted-sum-approach is used for (11) and (12). The weighted-sum-approach considers the above two objective to a single objective function. Therefore to restrict the system closed-loop eigenvalues in the *D*-shape sector illustrated in Figure 6, the following objective function is defined:

$$\max F = f_1 + f_2 \qquad (13)$$

Furthermore, the design problem can be formulated as the constrained optimization problem, where the constraints are the bounds on the PSS parameters:

$$\begin{cases} 1 \le K \le 50 \\ 1 \le T \le 10 \\ 0 \le T_i \le 2,\ i = 1,2,3,4 \end{cases} \tag{14}$$

For the second study system, the bounds on the SVC supplementary controller parameters are the same as the bounds on PSS parameters, except the gain:

$$1 \le K \le 100 \tag{15}$$

5. Design of Power System Stabilizer and Supplementary Controller for SVC

To provide a reasonable dynamic performance for the considered multi-machine power systems, damping controllers are designed using proposed approach. The results obtained by the proposed method are compared with conventional damping controller designed by SFLA.

5.1. The 2-area-4-Machine System

The goal of the optimization is to find the best value for the two PSSs in the 2-area-4-machine system. Therefore, a configuration is considered for each solution as a vector [$K, T, T_1, T_2, T_3, T_4; K, T, T_1, T_2, T_3, T_4$]. According to the fitness function defined in (13), the two PSSs are designed simultaneously so that the damping ratio of the close-loop system is increased as well as the eigenvalues of the close-loop system are shifted to the left-hand side. The boundary of D-shape sector for eigenvalues (see Figure 6) is chosen as $\sigma_0 = -0.2$ and $\xi_0 = 0.36$.

In SFL algorithms, during each generation, the frogs are evaluated with some measure of fitness, which is calculated from the objective function defined in (13), subject to (14). Then the best frogs are chosen. In the current problem, the best frog is the one that has minimum fitness. Based on Figure1 the local search and shuffling processes (global relocation) continue until the last iteration is met. The first step to implement the SFLA is generating the initial population (N frogs) where N is considered to be 100 (c.f. Figures. 1 and 2). For both SFLA and MSFLA, the number of memeplex is considered to be 10 and the number of evaluation for local search is set to 10. The other initial parameters are obtained by try-and error and are same for both algorithms. Also, $D_{\max}$ is chosen as $inf = \infty$ and the maximum number of iteration is set to be 100. Moreover, based on the author's previous experience, C_1 is chosen as 1.5.

To find the best value for the controller parameters, [$K, T, T_1, T_2, T_3, T_4; K, T, T_1, T_2, T_3, T_4$]; the algorithms are run for 10 independent runs under different random seeds. The results obtained by the SFL algorithms are shown in Table 1. Also, Table 2 shows the system close-loop eigenvalue with minimum damping ratio for designed PSSs by SFLA and MSFLA. It shows that the MSFLA has better performance to move the worst eigenvalue to the D-shape sector.

Table 1. The results obtained by SFLA and MSFLA

Algorithm	PSSs	K	T	T_1	T_2	T_3	T_4
SFLA	PSS1	48.81	4.67	0.613	0.252	1.769	0.521
	PSS2	48.98	4.563	1.943	0.426	0.671	1.023
MSFLA	PSS1	42.16	4.871	1.698	1.987	1.317	0.211
	PSS2	49.31	4.150	1.760	0.499	0.876	1.891

Table 2. The system closed-loop eigenvalues with minimum damping ratio for designed PSSs by each algorithm

Algorithm	Low Frequency Mode	Frequency	Damping Ratio
Without PSSs	0.165 ± 3.485i	0.555	-0.472
SFLA	-1.865 ± 5.78i	0.919	0.307
MSFLA	-2.401 ± 6.22i	0.969	0.361

For the designed PSSs, the average best-so-far of each run are recorded and averaged over 10 independent runs. To have a better clarity, the convergence characteristics in finding the best values of PSSs parameters is given in Figure 7, where shows MSFLA performs better than SFLA at early iterations.

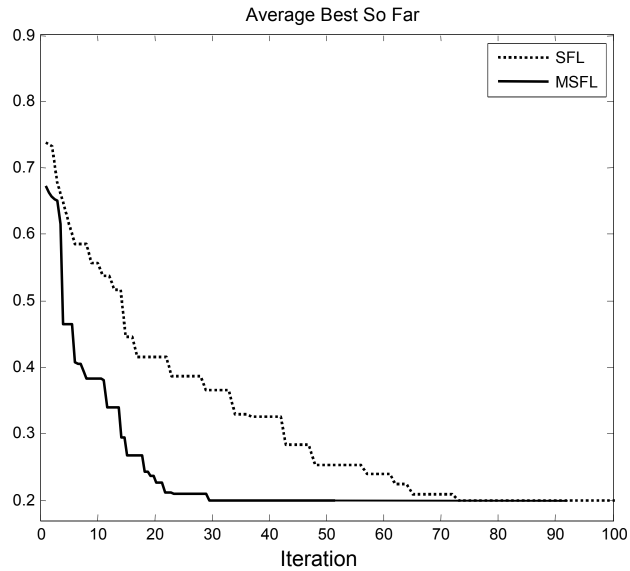

Figure 7. Convergence characteristics of SFLA and MSFLA on the average best-so-far function in PSSs design.

The obtained PSSs by two algorithms are placed in the study system (Figure 3). To show the effectiveness of the designed controllers, a time-domain analysis is performed for the study system. A line-to-ground fault is applied in one of the tie lines at bus 3. The fault persisted for 70.0 *ms*. The behavior of the system was evaluated for 15 s. The machine angles, δ, with respect to a particular machine (machine 1), were computed over the simulation period and shown in Figures. 8 and 9. These figures show that two controllers provide a good damping for the study system, but the one designed by MSFLA performs better.

Once again to show the robustness of the designed controllers, a three-phase fault is applied in one of the tie circuits at bus 3. The dynamic behavior of the system was evaluated for 15 s. The machine angles, δ, were computed over the simulation period and shown in Figures. 10 and 11. These responses, similar to Figures 8 and 9, show the robustness of the designed controllers.

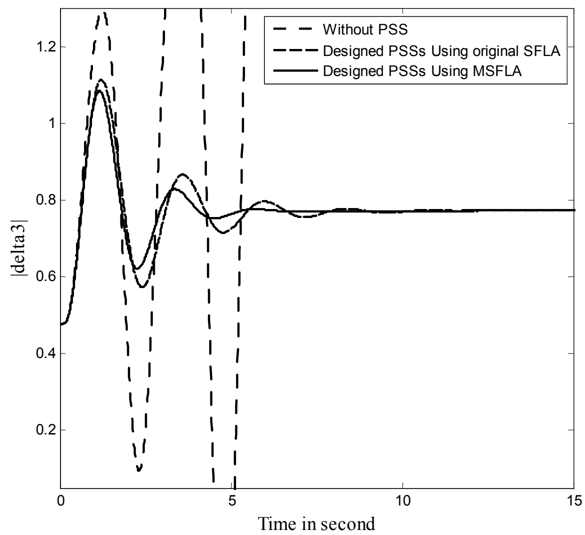

Figure 8.The response of the generator 3 to a line-to-ground fault at bus 3.

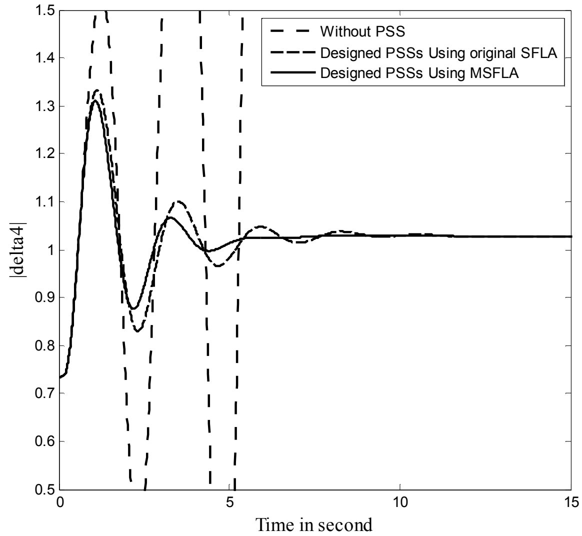

Figure 9.The response of the generator 3 to a line-to- ground fault at bus 4.

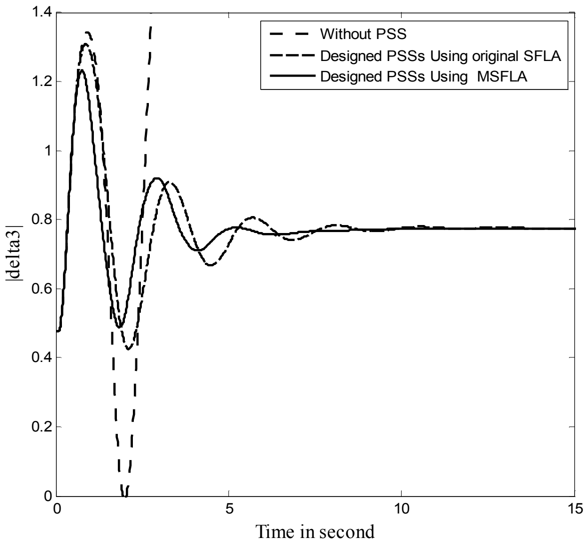

Figure 10.The response of the generator 3 to a three-phase fault at bus 3.

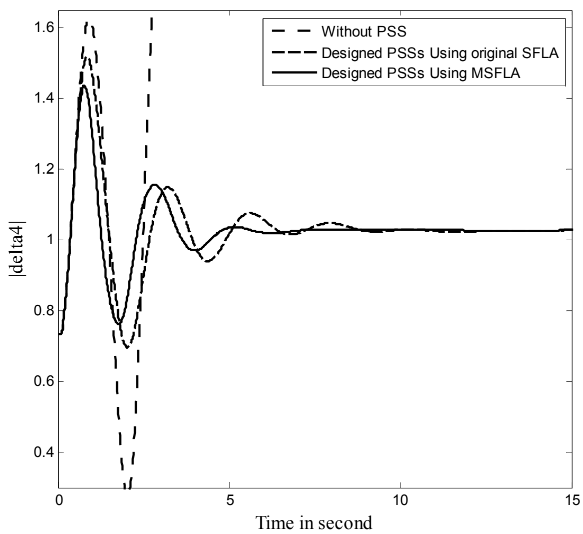

Figure 11.The response of the generator 3 to a three-phase fault at bus 4.

5.2. The 5-Area-16-Machine System

The goal of the optimization is to find the best value for PSS and supplementary controller in the 5-area-16-machine system. Therefore, a configuration is considered for each solution as a vector , [K, T, T_1, T_2, T_3, T_4; K, T, T_1, T_2, T_3, T_4].The initial population (N frogs) is considered to be 200 (c.f. Figures 1 and 2). For both SFLA and MSFLA, the number of memeplex is considered to be 15 and the number of evaluation for local search is set to 10. Also, $D_{\max}$ is chosen as $inf = \infty$ and the maximum number of iteration is set to be 100. Moreover, based on the author's previous experience, C_1 is chosen as 1.5. The results obtained for both algorithms are shown in Table 3. It should be noted that the boundary of *D*-shape sector for eigenvalues (Figure 6) is defined with $\sigma_0 = -0.16$ and $\xi_0 = 0.049$. Table 4 shows the system close-loop eigenvalue with minimum damping ratio for designed controllers by each algorithm. It shows that the MSFLA performs better than original SFLA.

Table 3. The Results Obtained By SFLA and MSFLA

Algorithm	PSSs	K	T	T_1	T_2	T_3	T_4
SFL	PSS	30.41	2.446	1.126	0.965	1.786	0.959
	Supp-Controller	14.20	8.770	1.880	1.113	0.004	0.195
MSFL	PSS	33.60	5.838	1.638	0.552	1.735	0.508
	Supp-Controller	10.39	5.665	0.007	0.903	0.115	1.285

Table 4. The system closed-loop eigenvalues with minimum damping ratio for designed damping controllers By Each Algorithm

Algorithm	Low Frequency Mode	Frequency	Damping Ratio
Without PSS	0.01 ± 7.48i	1.2	-0.0013
SFL	-0.129 ± 3.21i	0.510	0.049
MSFL	-0.185 ± 3.78i	0.601	0.048

To clarify performance of each of algorithms; especially the convergence characteristics in finding the best values of parameters, each algorithm are run 10 times independently and their average best so far is depicted in Figure 12.

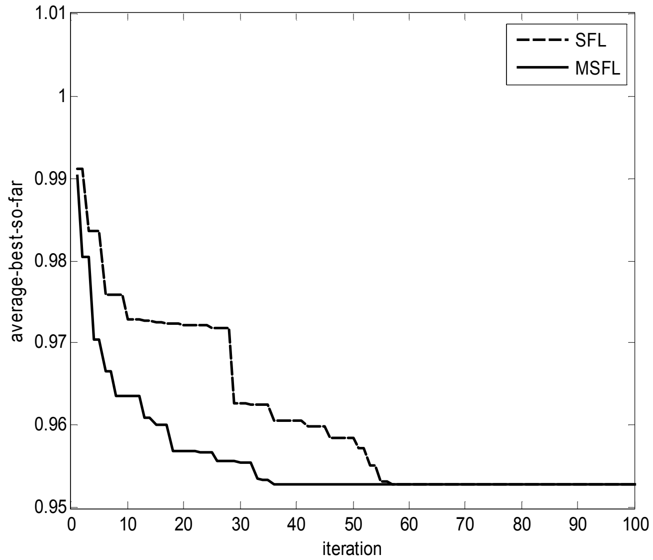

Figure 12. Convergence characteristics of MSFL and SFLA on the average best-so-far function in PSS and supplementary controller design.

The obtained PSS and supplementary controller by SFLA and MSFLA are placed in the 5-area-16-machine system (Figure 4). To show the effectiveness of the designed controller, a time-domain analysis is performed for this system. A line-to-ground fault is applied in one of the tie lines at bus 26. The fault persisted for 70.0 *ms*. The behavior of the system was evaluated for 20 s. The machine angles, δ, with respect to a particular machine (machine 13), were computed over the simulation period and shown in Figures 13-15. These figures show that both controllers provide a good damping for the study system, but the one designed by MSFLA performs better.

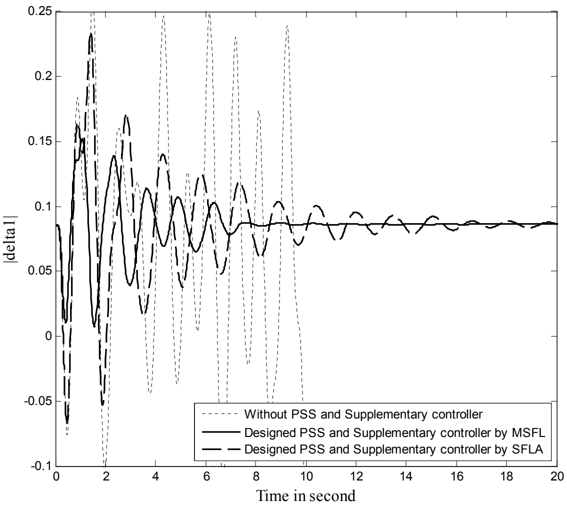

Figure 13. The response of generator 1 to a line-to ground fault at bus 26.

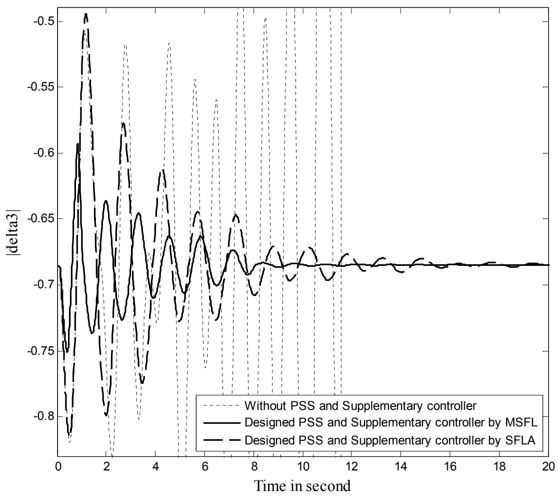

Figure 14. The response of generator 3 to a line-to-ground fault at bus 26.

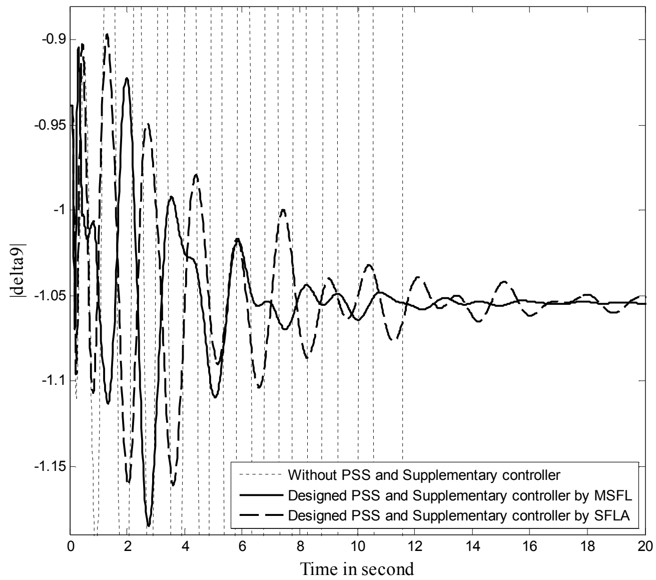

Figure 15. The response of generator 9 to a line-to-ground fault at bus 26.

Once again to show the robustness of the designed controllers for different operating conditions, a three-phase fault is applied in one of the tie circuits at bus 26. The dynamic behavior of the system was evaluated for 20 s. The machine angles, δ, were computed over the simulation period and is shown in Figures 16-18. These responses show that the robustness of the designed controllers by MSFLA is better.

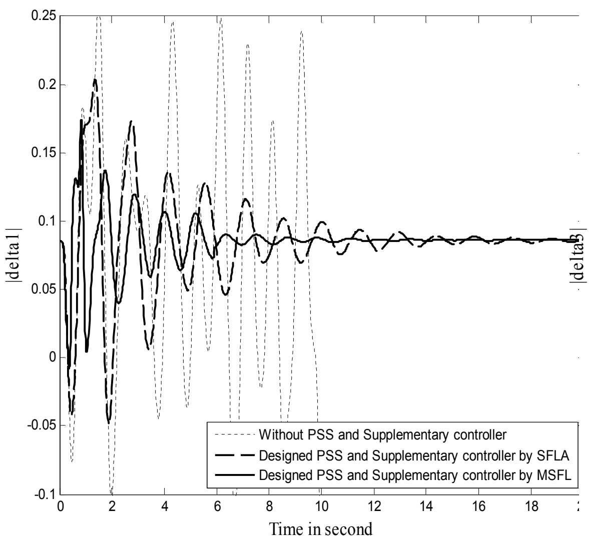

Figure 16. The response of generator 1 to a three-phase fault at bus 26.

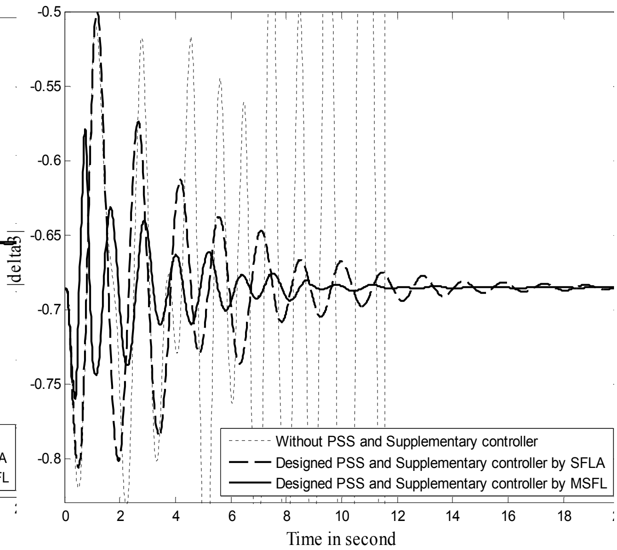

Figure 17. The response of generator 3 to a three-phase fault at bus 26.

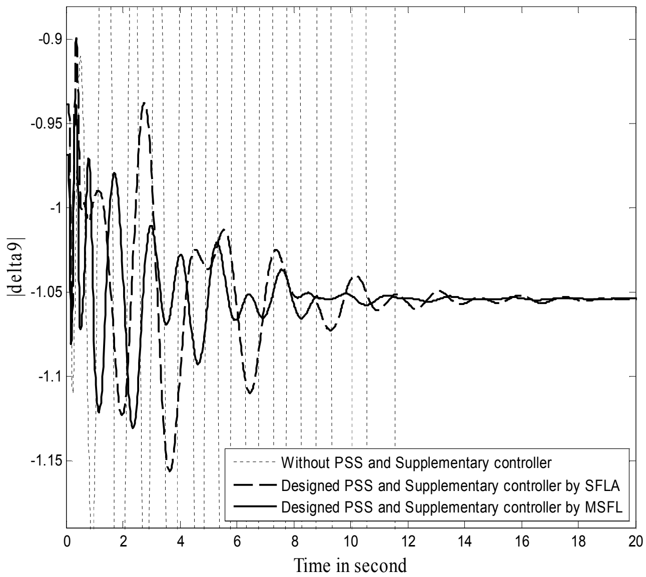

Figure 18. The response of generator 9 to a three-phase fault at bus 26.Conclusion.

6. Conclusion

In this paper a modified shuffled frog leaping (MSFL) algorithm, is used to simultaneously design coordinated damping controllers. For this purpose, the parameters of the controllers are determined using an eigenvalue-based objective function. In SFL, the local search is done through the evolution in memeplexes. The issue of exploration and exploitation is taken into account by a frog leaping rule for local search and a memetic shuffling rule for global information exchange. In this paper, learning mechanism is improved. To show the effectiveness and robustness of the designed controller, a line-to-ground fault and a three-phase fault are applied at a bus. The simulations studies show the designed controllers improve the stability of the system. Also, the obtained results show that the MSFLA has better performance in compression to SFLA for the current problem.

7. References

[1] P. Kundur, "power system stability and control," *New York, McGraw-Hill*, 1994.

[2] J.H. Chow, J.J. Sanchez-Gasca, "Pole-Placement Design of Power System Stabilizers," *IEEE Transactions on Power Systems*, 1989.

[3] R.J. Fleming, J. Sun, "An Optimal Multivariable Stabilizer for a Multimachine Plant," *IEEE Transactions on Energy Conversio*n, 1990.

[4] C. Mao, O.P. Malik, G.S. Hope, J. Fun, "An Adaptive Generator Excitation Controller Based on Linear Optimal Control," *IEEE Transactions on Energy Conversion*, 1990.

[5] V. Samarasinghe, N. Pahalawaththa, "Damping of Multimodal Oscillations in Power Systems Using Variable Structure Control Techniques," *IEE Proc. Gener. Transm. Distrib*, 1997.

[6] T. Hussein, M.S. Saad, A.L. Elshafei, A. Bahgat, "Damping inter-area modes of oscillation using an adaptive fuzzy power system stabilizer," Electric Power Systems Research, 2010.

[7] W. Liu, G.K. Venayagamoorthy, D.C. Wunsch, "Adaptive Neural Network Based Power System Stabilizer Design," Proceedings of the International Joint Conference on Neural Network, 2003.

[8] M. A. Abido, "Optimal Design of Power–System Stabilizers Using Particle Swarm Optimization", IEEE Transaction on Energy Conversion, 2002.

[9] Y.L. Abdel-Magid, M.A. Abido, "Optimal Multiobjective Design of Robust Power System Stabilizers Using Genetic Algorithms," IEEE Transaction. Power System, 2003.

[10] M.A. Abido, "Robust Design of Multimachine Power System Stabilizers Using Simulated Annealing," IEEE Transaction. Energy Conversion, 2000.

[11] S. Kyanzadeh, M.M. Farsangi, H. Nezamabadi-pour, K.Y. Lee, "Design of Power System Stabilizer Using Immune Algorithm," 14th International Conference on Intelligence Systems Application to power Systems, 2007.

[12] M. Khaleghi, M.M. Farsangi, H. Nezamabadi-pour, "Design of a Damping Controller for SVC Using AINet," In Proc. Technical and Physical Problems of Power Engineering, 2008.

[13] E. Bijami, J. Askari and M.M. Farsangi, "Power System Stabilizers Design by Using Shuffled Frog Leaping," The 6th International Conference on Technical and Physical Problems of Power Engineering, 2010.

[14] E. Bijami, J. Askari, "Imperialist Competitive Algorithm for Simultaneous Coordinated Tuning of Damping Controller," International Journal on Technical and Physical Problems of Engineering, 2012.

[15] E. Bijami, M. Shahriari-kahkeshi, H. Zam zam, "Simultaneous Coordinated Tuning of Power System Stabilizers Using Artificial Bee Colony Algorithm," 26th International Power System Conference, 2011.

[16] E. Bijami, R. Abshari, J. Askari, S. Hoseinnia and M.M. Farsangi, "Optimal Design of Damping Controllers for Multi-machine Power Systems Using Metaheuristic Techniques," International Review of Electrical Engineering, 2011.

[17] M.M. Eusuff, K.E. Lansey, F. Pasha, "Shuffled Frog Leaping Algorithm: a Memetic Meta-heuristic for Discrete Optimization," Engineering Optimization, 2006.

[18] E. Elbeltagi, "A Modified Shuffled Frog Leaping Algorithm for Optimizing Bridge-desk Repairs," International conference on bridge management systems, 2006.

[19] E. Elbeltagi, "Evolutionary Algorithms for Large Scale Optimization In Construction Management," The Future Trends in the Project Management, 2007.

[20] T.H. Huynh, "A Modified Shuffled Frog Leaping Algorithm for Optimal Tuning of Multivariable PID Controllers," Fifth IEEE International Conference on Industrial Technology, 2008.

[21] X. Zhang, X. Hu, G. Gui, Y. Wang, Y. Niu, "An Improved Shuffled Frog Leaping Algorithm with Cognitive Behavior," 7th World Congress on Intelligent Control and Automation, 2008.

[22] A. Stacey, M. Jancic, I. Grundy, "Particle Swarm Optimization with Mutation," The 2003 Congress on Evolutionary Computation (CEC '03), 2003.

[23] J. Chow, "Power System Toolbox: A Set of Coordinated m-Files for Use with MATLAB," ON, Canada: Cherry Tree Scientific Software, 1997.

[24] M. M. Farsangi, H. Nezamabadi-pour, Y. H. Song, K. Y. Lee, "Placement of SVCs and Selection of Stabilizing Signals in Power Systems," IEEE Transaction on Power Systems, 2007.