World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:6, No:11, 2012

Identification of MIMO Systems Using Neuro-Fuzzy Models with a Shuffled Frog Leaping Algorithm

Sana Bouzaida, Anis Sakly, and Faouzi M'Sahli

Abstract**—**In this paper, a TSK-type Neuro-fuzzy Inference System that combines the features of fuzzy sets and neural networks has been applied for the identification of MIMO systems. The procedure of adapting parameters in TSK model employs a Shuffled Frog Leaping Algorithm (SFLA) which is inspired from the memetic evolution of a group of frogs when seeking for food. To demonstrate the accuracy and effectiveness of the proposed controller, two nonlinear systems have been considered as the MIMO plant, and results have been compared with other learning methods based on Particle Swarm Optimization algorithm (PSO) and Genetic Algorithm (GA).

Keywords**—**Identification, Shuffled frog Leaping Algorithm (SFLA), TSK-type neuro-fuzzy model.

I. INTRODUCTION

NEURO-fuzzy models have seen increasing interest in last decade. They have emerged from the fusion of neural networks and fuzzy inference systems. These techniques have advantages of excellent capability to deal with complex systems.

Many different structures for fuzzy neural networks have been proposed [1]. Among them the Takagi Sugeno model which is the most efficient type of the fuzzy models and has been applied widely because of his capability to approximate a wide range of non-linear systems [2].

Various learning algorithms have been applied to optimize the parameters of neuro-fuzzy systems. More recently, optimization techniques based on Evolutionary Algorithms, such as the particle swarm optimization algorithm "PSO" [3] and Genetic algorithm "GA" [4] were proposed for training the different parameters of ANFIS. In this study a novel metaheuristic algorithm called the Shuffled Frog Leaping Algorithm "SFLA" is used to train the ANFIS as a MIMO controller. The proposed algorithm "SFLA" is a mimetic meta-heuristic algorithm developed by Eusuff and Lansey [5], it combines the benefits of the local search tool of the Particle Swarm Optimization (PSO) and the competitiveness mixing of information of the shuffled complex evolution technique. The SFLA has been tested in several optimization problems [6,7,8] and found to be efficient in finding global solutions.

The rest of this paper is outlined as follows: Section II describes the concept of ANFIS model. Section III explains the overview of SFLA. Then, section IV illustrates simulation results and compares the performance of developed algorithm with PSO and GA. Conclusions are given in the last section.

S. Bouzaida is with the National School of Engineers of Monastir ENIM, University of Monastir (e-mail: bouzaida_sana@hotmail.fr).

A. Sakly is with the National School of Engineers of Monastir ENIM, University of Monastir (e-mail: Sakly_anis@yahoo.fr)

F. M'sahli is with the National School of Engineers of Monastir ENIM, University of Monastir (e-mail: faouzi.msahli@enim.rnu.tn).

II. STRUCTURE OF TSK-TYPE FUZZY MODEL

The TSK-type neuro-fuzzy model employs If-Then rules, where the rule consequents are usually constant values or linear functions of inputs. The j^{th} rule in the neuro-fuzzy model is represented as follows:

If x_1 is A_1^j and … and x_n is A_n^j **Then** y is f_j (1)

where $x_1,..,x_n$ are input variables, y is the system output variable, A_i^j is a fuzzy set, and f_j is a linear function given by:

$$f_j = a_{1j}x_1 + a_{2j}x_2 + \ldots + a_{nj}x_n + a_{(n+1)j} \tag{2}$$

Fuzzy set A_i^j uses a Gaussian membership function such that:

$$A_i^j = \exp\left\{-\frac{1}{2}\left(\frac{x_i - c_{ij}}{\sigma_{ij}}\right)^2\right\} \tag{3}$$

Fig. 1 shows the structure of a TSK model, where n and R are the number of input dimensions and the number of rules, respectively.

It is a five-layer network structure. In the proposed TSK model, the firing strength of a fuzzy rule is calculated by the following "and" operation:

$$\mu_j = \prod_{i=1}^{n} A_i^j \tag{4}$$

The output is given by:

$$y = \frac{\sum_{j=1}^{R} \mu_j f_j}{\sum_{j=1}^{R} \mu_j} \tag{5}$$

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:6, No:11, 2012

The fuzzy design can be considered as an optimization problem, where the parameter set of the premise part and consequent part of Takagi and Sugeno's model are required to be identified. The fuzzy rule antecedents are usually determined by clustering algorithms, and the least square method is applied to find the optimal consequent parameters. In recent literature, EAs such as GA [4] and PSO [3], have been applied to construct fuzzy models.

In this paper, we adopt a TSK-type neuro-fuzzy model to perform modelling problems. The different parameters of TSK model are optimized by a novel evolutionary algorithm that is called SFLA.

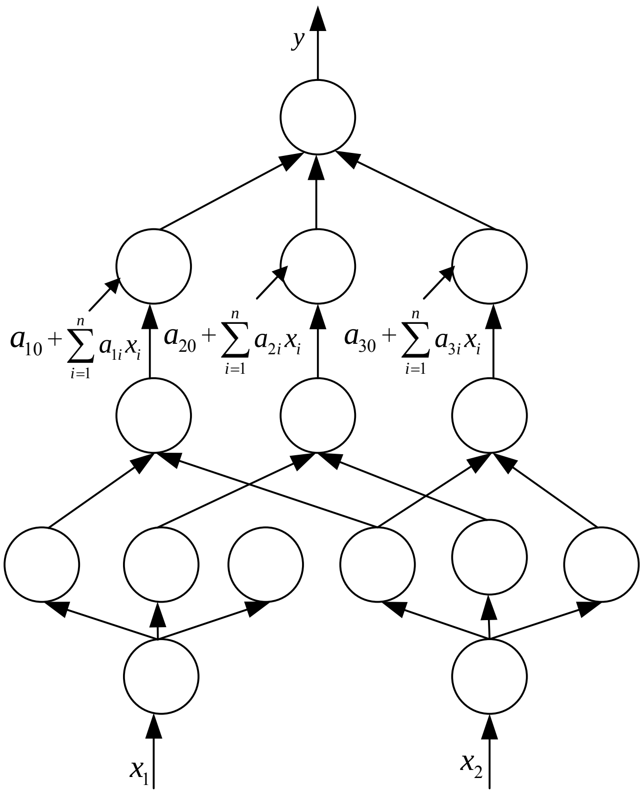


Fig. 1 Structure of TSK-type neuro-fuzzy model

III. The Proposed Shuffled Frog Leaping Algorithm Procedure for the TSK-Type Neuro Fuzzy Model

The following SFLA has been adopted for training the TSK-type neuro-fuzzy model.

Step 1: Generate randomly the initial population of frogs. Each frog represents the entire antecedent and consequent parameters for the MIMO controller.

Step 2: Sort the frogs in descending order according to their fitness.

The RMS error is used to test the performance indices, it is defined as follows:

$$RMSE=\sqrt{\frac{1}{n}\sum_{i=1}^{n}((y_1(i)-y_{d1}(i))^2+(y_2(i)-y_{d2}(i))^2)} \qquad (6)$$

Step 3: Divide the population in M memplexes. In this process, the first frog goes to the first memeplex, the second frog goes to the second memeplex, the frog m goes to the m^{th} memeplex, and frog m+1 goes to the first, etc.

Step 4: Improve the worst frog position in each memeplex using the equation (7) and equation (8):

$$D=r\cdot(X_b-X_w) \qquad (7)$$

$$X_w^{new}=X_w^{current}+D \qquad (8)$$

$$(D_{max}\geq D\geq -D_{max})$$

where r is a random number in the range of [0 1], D is the step size vector and D_{max} is the maximum allowed change in a frog's position. If the new frog position does not improve, then (7) and (8) are repeated with respect to the global best frog (X_g replaces X_b). If no improvement becomes impossible in this latter case, then a random frog is generated to replace the old frog position.

Step 5: Stop if the maximum number of generations is reached. Otherwise repeat Step 2-5 until convergence criteria is satisfied.

IV. Simulation Results

To demonstrate the effectiveness and efficiency of the proposed SFLA, two nonlinear MIMO plants were considered to be identified. In this work, two TSK-type neuro-fuzzy are used to identify the MIMO plant. Each of these two models has the same structure described above. The number of rules is fixed to 4, thus, resulting in 40 parameters for the SISO case. Hence, 80 parameters are required to represent the MIMO identifier. To evaluate the performance of the developed SFLA, TSK-type neuro-fuzzy model, using GA and PSO, respectively, were applied to the same examples. The simulation studies are carried out in MATLAB environment.

Table I lists the optimization parameters of the proposed SFLA, PSO and GA algorithms.

TABLE I
Performance Comparison with Different Methods

SFLA	Number of memplexes : 6
	Number of frogs in each memplex : 10
	Number of iterations in each memplex :10
PSO	Number of populations: 60
	w_{max}=0.9,w_{min}=0.4,c_1=2,c_2=2
GA	Number of population: 60
	Mutation probability: 0.2
	Crossover probability: 0.8

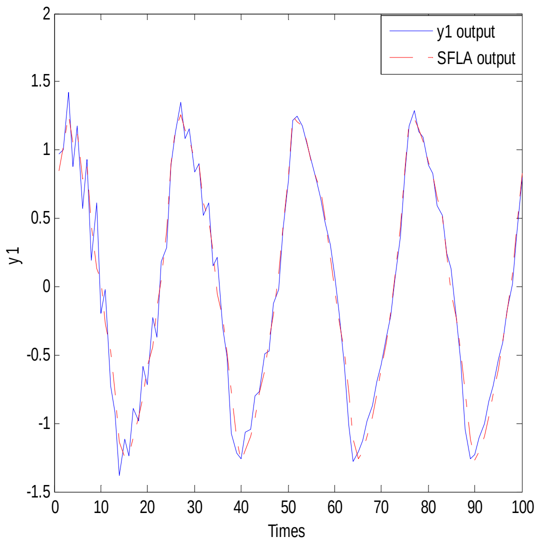
Example 1: In this example, the nonlinear MIMO system to be identified is described by [9].

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} \dfrac{y_2(k)}{1+y_1^2(k)} \\ \dfrac{y_1(k)}{1+y_2^2(k)} \end{bmatrix} + \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \tag{9}$$

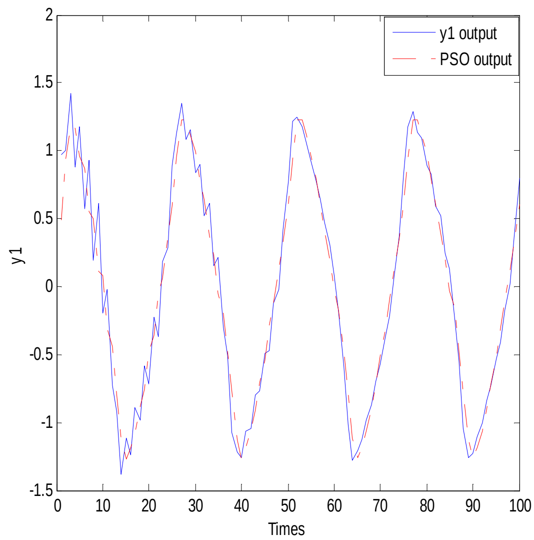
Where $\left(x_1(k), x_2(k)\right) = \left(\cos\left(\dfrac{2\pi k}{100}\right), \sin\left(\dfrac{2\pi k}{100}\right)\right)$ for $0 \le k \le 100$ and 101 training input-output data are generated by substituting $\left(x_1(k), x_2(k)\right)$ into (9) sequentially. The evolution process progressed for 100 generations.

The details of the simulation results are shown in Figs. 2-4, respectively.

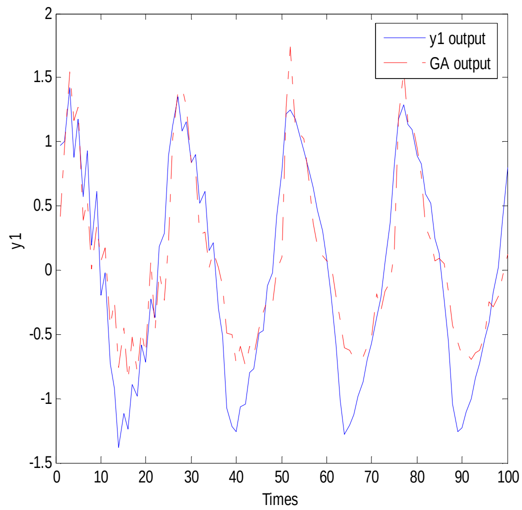
As shown in Fig. 4 which illustrates the RMSE convergence characteristic of different methods, the SFLA has a fast convergence speed and also a higher convergence precision.



(a)

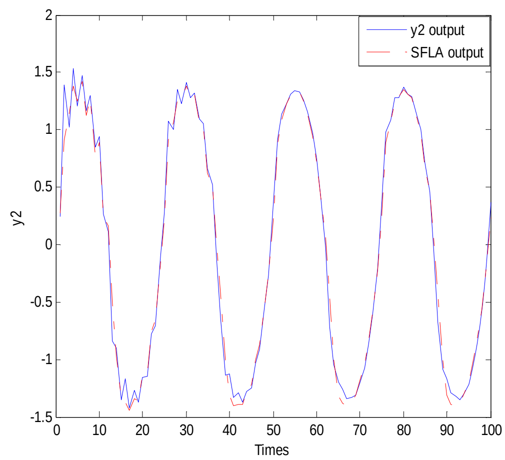


(b)

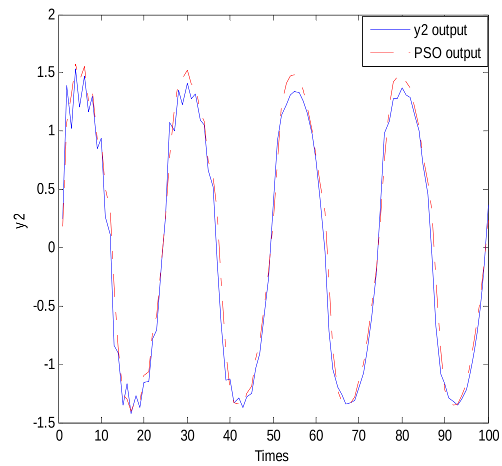


(c)

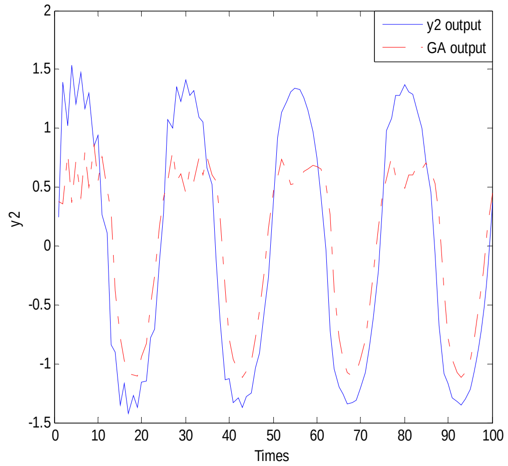
Fig. 2 The desired output for y_1, (a) SFLA output, (b) PSO output and (c) GA output



(a)



(b)



(c)

Fig. 3 The desired output for y_2, (a) SFLA output, (b) PSO output and (c) GA output

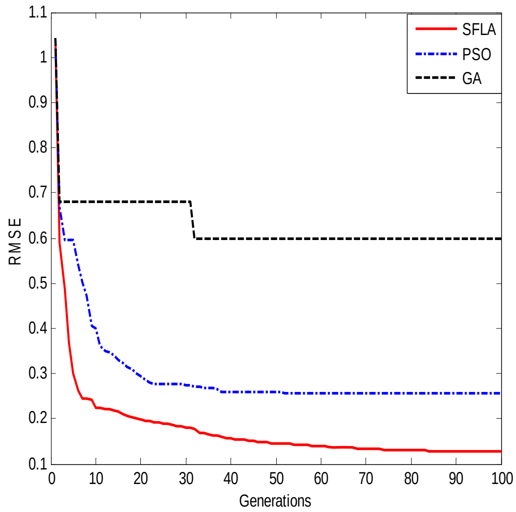


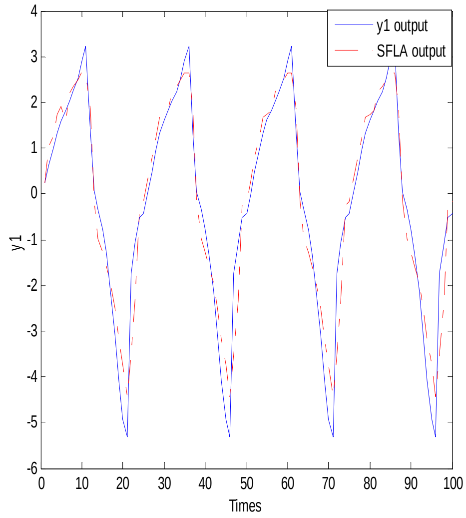
Fig. 4 RMSE curves for SFLA, PSO and GA methods

Example 2: In this example, the nonlinear MIMO system to be identified is described by [10]

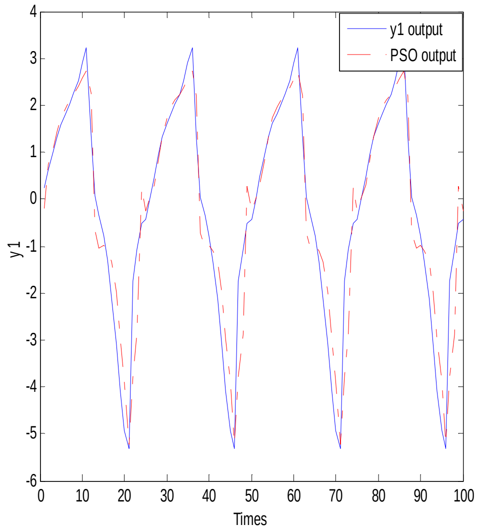
$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} \dfrac{y_1(k)}{1+y_2^2(k)} \\ \dfrac{y_1(k)y_2(k)}{1+y_2^2(k)} \end{bmatrix} + \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \tag{10}$$

Where $(x_1(k), x_2(k)) = \left(\sin\left(\dfrac{2\pi k}{100}\right), \cos\left(\dfrac{2\pi k}{100}\right)\right)$ for $0 \le k \le 100$ and 101 training input-output data are generated by substituting $(x_1(k), x_2(k))$ into (10) sequentially. The evolution process progressed for 200 generations.

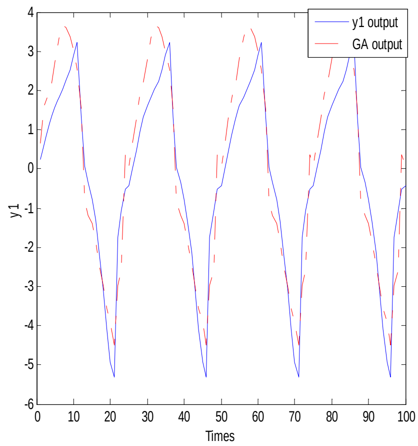
The details of the simulation results are shown in Figs. 5-7, respectively.



(a)

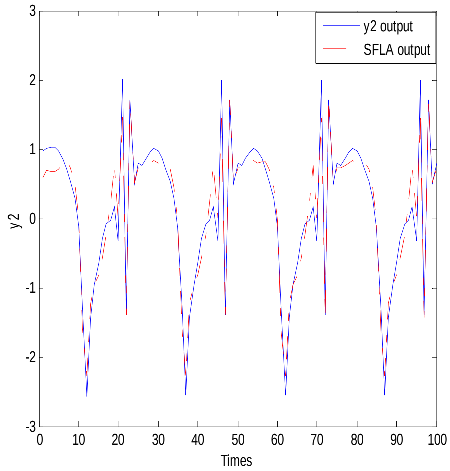


(b)

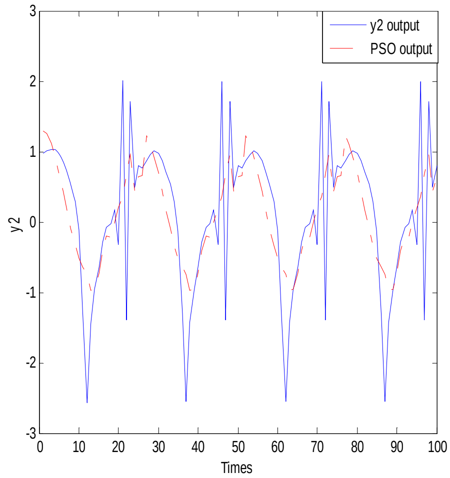


(c)

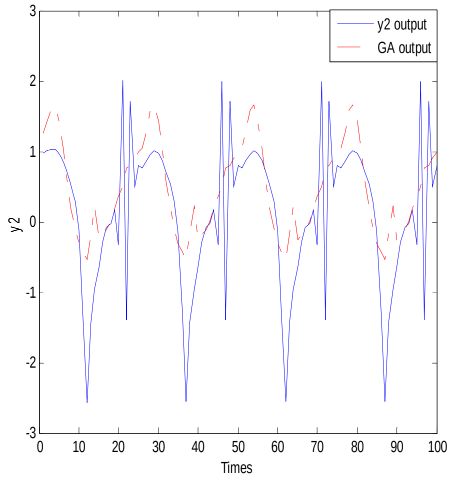
Fig. 5 The desired output for y_1, (a) SFLA output, (b) PSO output and (c) GA output



(a)



(b)



(c)

Fig. 6 The desired output for y_2, (a) SFLA output, (b) PSO output and (c) GA output

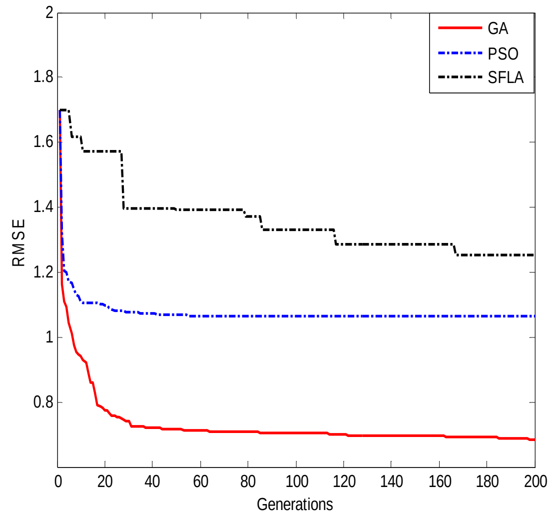


Fig. 7 RMSE curves for SFLA, PSO and GA methods

Fig. 7 shows that SFLA obtains smaller RMSE than the other optimization algorithms with the faster convergence speed.

TABLE II
THE FINAL VALUES OF RMSE AFTER 200 EPOCHS

	Example 1			Example 2		
	SFLA	PSO	GA	SFLA	PSO	GA
RMSE	0.11	0.25	0.6	0.45	1.1	1.21

The RMSE values for the above two examples are summarized in Table II, from which one can conclude that SFLA has good performance in identifying the two different MIMO systems.

V. CONCLUSION

In this paper, a new meta-heuristic search technique, known as SFLA is applied for tuning parameters of TSK-type neuro-fuzzy model. By comparing with PSO and GA in the aspects of optimal solution, results show that SFLA has much better optimizing accuracy in the different examples of modeling MIMO systems.

REFERENCES

[1] Y. Zhou, S. Li, R. Jin, "A new fuzzy neural network with fast learning algorithm and guaranteed stability for manufacturing process control", Fuzzy Sets Syst, pp 201–216, December 2002.
[2] C.-C. Chuang, S.-F. Su, S.-S. Chen, "Robust TSK fuzzy modeling for function approximation with outliers", IEEE Transactions on Fuzzy Systems, pp 810–821, 2001.
[3] J. Kennedy, RC. Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.
[4] G. Leng, T.M. McGinnity, G. Prasad, Design for self-organizing fuzzy neural networks based on genetic algorithms, IEEE Transactions on Fuzzy Systems, pp 755-766, 2006.
[5] M. Eusuff, K. Lansey, and F. Pasha, Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization, Engineering Optimization, pp 12-154, 2006.

[6] T.-H. Huynh, A modified shullfed frog leaping algorithm for optimal tuning of multivariable PID controllers, IEEE International Conference on Industrial Technology, 2008, pp 20-23.
[7] Eslamian, M., S.H. Hosseinian, and B. Vahidi, Bacterial Foraging-Based Solution to the Unit-Commitment Problem, IEEE Transactions on Power Systems, pp 1478 -1488, 2009.
[8] A. Rahimi-Vahed and A. H. Mirzaei, Solving a bi-criteria permutation flow-shop problem using shuffled frog-leaping algorithm, Soft Computing, pp 435-52, 2008.
[9] S. Purwar, I.N. Kar, A.N. Jha, On-line system identification of complex systems using Chebyshev neural networks, Applied Soft Computing 7, pp. 364–372, 2007.
[10] K.S. Narendra, Identification and control of dynamical systems using neural network, IEEE Transactions on Neural Networks, pp 4–27, 1990.