Optimizing a Neuro-Fuzzy System based on nature inspired Emperor Penguins Colony optimization algorithm

Sasan Harifi, Madjid Khalilian, Javad Mohammadzadeh, and Sadoullah Ebrahimnejad

***Abstract*—A neuro-fuzzy system is a learning machine that finds the parameters of a fuzzy system using approximate techniques of neural networks. Both neural network and fuzzy system have common features. These can solve problems that there is no mathematical model for them. Adaptive Neuro-Fuzzy Inference System (ANFIS) is an adaptive network that uses supervised learning on learning algorithm. Selecting the optimization method in training, to achieve effective results with ANFIS is very important. Heuristics and metaheuristics algorithms attempt to find the best solution out of all possible solutions to an optimization problem. ANFIS training can be based on non-derivative algorithms. Heuristics and metaheuristics are non-derivative algorithms that can lead to better performance in ANFIS training. Most heuristic and metaheuristic algorithms are taken from the behavior of biological systems or physical systems in nature. The newly released Emperor Penguins Colony (EPC) algorithm is a population-based and nature-inspired metaheuristic algorithm. This algorithm has many potentials for solving various problems. In this paper, an optimized ANFIS based on the newly EPC algorithm is proposed. The optimized ANFIS is compared with other non-derivative algorithms on benchmark datasets. Eventually, the proposed algorithm is used to solve the classical inverted pendulum problem. The results show that the proposed ANFIS based on the EPC algorithm has less error and better performance than other state-of-the-art algorithms in both training and testing phase.**

***Index Terms*—Optimization, Neuro-Fuzzy system, Emperor Penguins Colony algorithm, nature inspired, ANFIS, Fuzzy inference system, Sugeno-type fuzzy**

I. Introduction

NEURO-FUZZY computing is a popular framework for solving complex problems. A neuro-fuzzy system is a fuzzy system that uses a learning algorithm. This system determines the fuzzy system parameters by processing data samples. The learning procedure of a neuro-fuzzy system takes the semantically properties of the basic fuzzy system into account. This results in constraints on the possible modifications applicable to the system parameters.

Optimization is the mathematical process, the process of adjusting the inputs of a device, or experiment to find the minimum or maximum output or result. Inputs include variables, such as process or function as cost function, target function, fitness function, and outputs include cost and fitness. Optimization algorithms attempt to find the best solution out of all possible solutions to an optimization problem.

Recently, the use of optimization and evolutionary algorithms have increased in the design of neuro-fuzzy systems. This increase in usage maybe due to their multifold advantages. Some of these advantages include scalability to high-dimensional solution spaces, parallel architecture, and simplicity of implementation [1]. Optimization and evolutionary algorithm, fuzzy theorem, and neural networks are three basic paradigms of soft computing [2].

The Fuzzy Inference System (FIS) has fuzzy sets, fuzzy operators and the knowledge base. A FIS needs to specify architecture and learning algorithm for a specific application, which is similar to the Artificial Neural Network (ANN). Both FIS and ANN have some drawbacks but these two approaches are complementary. So building an integrated system combining the concepts can be a good idea. ANN learning capability is an advantage from the viewpoint of FIS [3].

The two popular FIS are Sugeno-type and Mamdani type. Sugeno neuro-fuzzy computing method uses the theory of neural network and Mamdani-type fuzzy logic system technique as a robust tool for solving various problems with the high level of uncertainty in science and engineering problems concerned with issues such as pattern recognition, identification, controlling [4].

Many authors combine optimization algorithms with neuro-fuzzy systems. Zheng et al [1] used the Differential Biogeography-Based Optimization (DBBO) algorithm for parameter optimization of Sugeno type neuro-fuzzy system. Their experiments showed that the method provided by them was a good performance. Taghavifar et al [4] represented a Sugeno type neuro-fuzzy system in combination with the Differential Evolution (DE) optimization algorithm. They used their method to model wheel dynamics caused by road irregularities. Toosi and Kahani [5] utilized the Genetic

Sasan Harifi is with the Department of Computer Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran (email: s.harifi@kiau.ac.ir).

Madjid Khalilian is with the Department of Computer Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran (email: khalilian@kiau.ac.ir). This author is the corresponding author.

Javad Mohammadzadeh is with the Department of Computer Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran (email: j.mohammadzadeh@kiau.ac.ir).

Sadoullah Ebrahimnejad is with the Department of Industrial Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran. (email: ibrahimnejad@kiau.ac.ir)

Algorithm (GA) for the optimization of their neuro-fuzzy system. They used their system to detect intrusion. Obo et al [6] exploited GA, Evolutionary Programming (EP), and Evolution Strategy (ES) for parameter tuning and pruning of membership functions in the neuro-fuzzy system. Their purpose was the classification of human gestures. Chen et al [7] presented a Sugeno fuzzy system based on tuning the parameters using EP. Juang and Chang [8] designed fuzzy-rule-based systems using Ant Colony Optimization (ACO).

In other studies, Pandiarajan and Babulal [9] utilized the integration of the fuzzy logic system with harmony search (HS) algorithm to find the optimal solution for optimal power flow problem in a power system. Precup et al [10] proposed a tuning approach for a fuzzy control system using Gray Wolf Optimizer (GWO). Shi et al [11] discussed evolutionary fuzzy systems in their paper. Hancer et al [12] proposed a feature selection approach using fuzzy mutual information based on the Artificial Bee Colony (ABC) algorithm.

The neuro-fuzzy system in combination with Particle Swarm Optimization (PSO) has been applied to many real-world engineering problems. Chatterjee et al [13] adopted the PSO fuzzy-neural network for voice-controlled robot systems. Araujo and Coelho [14] provided PSO fuzzy modeling for an experimental thermal-vacuum system. Sharma et al [15] designed a stable adaptive fuzzy controller based on combination with PSO. Wai et al [16] represented an intelligent daily load forecasting with a fuzzy neural network and PSO. Li et al [17] proposed solubility prediction of gases in polymers using a fuzzy neural network based on the PSO algorithm. Osório et al [18] combined the Adaptive Neuro-Fuzzy Inference System (ANFIS) with the PSO algorithm for wind power forecasting application. Chen et al [19] constructed three algorithms for training the ANFIS. The optimization algorithms used by them were GA, PSO, and DE. They used their method as a data mining technique for the geographic information system. Many ANFIS training approaches have been reviewed in the paper written by Karaboga and Kaya [20].

The recently published Emperor Penguins Colony (EPC) [21] algorithm, is a population-based and nature-inspired optimization algorithm. This algorithm inspired by the behavior of the emperor penguins in Antarctica. This algorithm has a lot of potential in solving various and high-dimensional problems. In this paper, the neuro-fuzzy system is combined with the EPC algorithm. In fact, using the EPC algorithm, the neuro-fuzzy system is optimally designed.

In general, the design steps of an optimal neuro-fuzzy system are as follows: Receive the training data; Creating a basic fuzzy system; Adjustment the parameters of the basic fuzzy system according to the modeling error function by the optimization algorithm; Returning of the fuzzy system has the best value of the parameters (minimum value of the error function) as the final result.

In this paper, a neuro-fuzzy system based on adaptive neuro-fuzzy inference system is designed. Then, the parameters of membership functions are extracted. After that, the parameters are optimized by the EPC optimization algorithm. Eventually, we return the optimized parameters to the system. The purpose is efficient optimizing the neuro-fuzzy system by using high flexibility and abundant potentials of the EPC algorithm.

Rest of the paper is structured as follows: Section II describes the EPC optimization algorithm in detail. Section III represents designing a neuro-fuzzy system. Section IV includes experimental results. Section V represents the statistical analysis. Section VI expresses the results of solving a classic engineering problem by the EPC algorithm. Finally, discussion and conclusions are provided in section VII and VIII, respectively.

II. Emperor Penguins Colony (EPC) Algorithm

In this section, the EPC algorithm is described briefly. Although the comprehensive description and explanatory details of the algorithm are presented by Harifi et al in [21].

The Emperor Penguins (*Aptenodytes forsteri*) are the largest species of penguins that live in Antarctica and on the sea-ice. The height of these penguins is about 110 to 130 cm in the walking mode and when extending its neck. The temperature of their living environment sometimes reaches -40 °C. To breeding and to escape from the harsh conditions, they have an interesting strategy. They form a mass composed of several penguins in their colonies, which is called the huddle. Fig 1 shows the sample of the emperor penguin huddling group.

Fig. 1. Emperor penguin huddling group.

When the huddle is formed, the amount of penguin's body heat dissipation is minimized and the temperature inside the huddle is raised. To distribute the heat uniformly in different parts of the huddle, the penguins perform spiral-like coordinated movements. The EPC algorithm is a population-based and nature-inspired algorithm that is inspired by the lifestyle of these penguins. This algorithm is controlled by the thermal radiation and spiral-like movement of penguins. Algorithm 1 describes the pseudo-code of the EPC algorithm.

In this algorithm, the heat absorption concept, thermal radiation, the attractiveness, and coordinated spiral movement must be calculated. To calculate the attractiveness, the heat radiation transfer must be calculated and for calculating heat radiation, the body surface area of the penguin is needed. The body surface area of the penguin is 0.56 m^2. Heat transfer occurs in three ways: thermal conduction, thermal convection, and thermal radiation. In the EPC algorithm, the criterion of the heat transfer by radiation is used. Radiation is a criterion of the attractiveness for penguins. The radiation emitted from the penguin body is obtained by the following equation,

$$Q_{\text{penguin}} = A\varepsilon\sigma T_s^4 \tag{1}$$

where A is body surface area of penguin and equal to 0.56 m^2. ε is emissivity of bird plumage which is considered 0.98 according to [22]. σ is Stefan-Boltzmann constant (5.6703×10^{-8} W/m^{2}K^4) and T is absolute temperature in Kelvin (K) which is considered 35 °C equal to 308.15 Kelvin.

In physics science, there are three types of heat sources, including surface sources, point sources and linear sources, which their attenuation coefficient is different for each other. According to the type of penguin's body physics, in the EPC algorithm, the heat source is considered linear. From the combination of the emitted radiation of the penguin body and the linear heat source equation, the attractiveness equation is obtained,

$$Q = A\varepsilon\sigma T_s^4 e^{-\mu x} \tag{2}$$

where μ is the attenuation coefficient and x is the distance between two linear sources. The μ parameter is an important factor in determining the rate of convergence and is considered as a positive value. Heat absorption is a concept defined by changes in μ. So as the attenuation coefficient decreases, the concept of heat absorption increases. That is, more heat is absorbed. The attractiveness equation must be combined into the spiral movement equation. The spiral movement is usually done in a clockwise direction and uses a logarithmic spiral equation. Subjective perception of the spiral-like movement is shown in Fig 2.

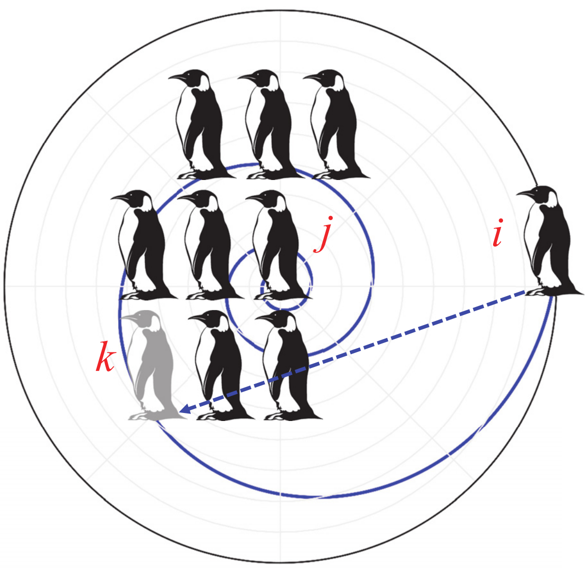

Fig. 2. Subjective perception of emperor penguin coordinated spiral movement.

In Fig 2, the penguin moves from point i to j, but does not reach point j. Depending on the amount of attractiveness, it moves along the logarithmic spiral path and stops at a point around the point j. Now the new position is k. The equation of this movement is as follows,

$$\begin{aligned} x_k &= ae^{b\frac{1}{b}\ln\{(1-Q)e^{b\alpha}+Qe^{b\beta}\}}\cos\left\{\frac{1}{b}\ln\{(1-Q)e^{b\alpha}+Qe^{b\beta}\}\right\} \\ y_k &= ae^{b\frac{1}{b}\ln\{(1-Q)e^{b\alpha}+Qe^{b\beta}\}}\sin\left\{\frac{1}{b}\ln\{(1-Q)e^{b\alpha}+Qe^{b\beta}\}\right\} \end{aligned} \tag{3}$$

where a and b are constant and are selected arbitrary. Q is attractiveness equation. x and y are parameters of logarithmic spiral [21]. α and β are $tan^{-1}\frac{y_i}{x_i}$ and $tan^{-1}\frac{y_j}{x_j}$, respectively.

Because the angle information is pre-determined, the spiral-like movement may become monotonous. To avoid the monotonous spiral path, the above equation summed with a random vector,

$$Eq.3 + \varphi\epsilon_i \tag{4}$$

where φ is the mutation factor in the change of path and ϵ is a random vector. If the distance between penguins is too big, then the attractiveness for most of the better penguins is zero value. As a result, the current penguin does not move in a spiral movement in the direction of a better one. This is an unpredictable case that may happen. In this case, $\varphi\epsilon_i$ helps the penguin to move a little bit from its position but does not move the spiral. It also helps the diversity.

Algorithm 1: Pseudo code of the Emperor Penguins Colony (EPC) algorithm.

```
STEP 1:
  generate initial population array of EPs (Colony Size);
  generate initial position and cost of each EP;
  calculate heat radiation (Eq. 1);
  determine initial attenuation coefficient (μ);
  determine initial mutation coefficient (φ);
  STEP 2: for FirstIteration to MaxIteration
    generate repeat copies of population array (as new_pop);
    STEP 3: for i=1 to n do (all n penguins)
      STEP 4: for j=1 to n do (all n penguins)
        if cost_j < cost_i then
          determine new position (Eq. 4);
          evaluate and store cost of new_pop;
        end if
      END STEP 4
    END STEP 3
    merge population array with new_pop array;
    sort and find best solution;
    update heat radiation (decrease);
    update attenuation coefficient (decrease);
    update mutation coefficient (decrease);
  END STEP 2
END STEP 1
```

In Algorithm 1, Equation 4 uses the result calculated in Equation 3 that is the coordinated spiral-like movement. Also, Equation 3 uses the result obtained from Equation 2, which is attractiveness. In the algorithm, we first get a copy of the population and then perform cost calculations on the new population. That is, the evaluation is done in the inner loop. Outside of the loop, the initial population (initial costs) is merged with the new population (new costs) and the best are selected. This process continues.

Proper adjustment of the parameters used in the algorithm to achieve desirable results is an important step. In our implementation, the initial $\mu = 1$ and the initial $\varphi = 0.4$.

Decreasing the parameters could happen in any manner. These changes can be linear or exponential. In the implementation, we used exponential changes. Thus, in each iteration, the parameter value φ is multiplied by 0.2. The goal is to reduce φ gradually because we need exploitation over time. This decrease in the value of φ causes the exploration to decrease and the exploitation to increase. Also in each iteration, the value of μ is multiplied by 0.98. The benefit is that the EPC algorithm can initially searching the space very quickly and testing many values, then calming down and converging to the best found positions.

For the algorithm, it has been inspired by the nature that after a while the value of heat radiation by the penguin decreases (due to energy consumption and metabolism). This reduction is very slight but there is. Also, as the huddle becomes denser over time, the penguin does not need to generate more heat to attract other penguins and reduces his own heat. In nature, this results in a concept of thermal equilibrium within the huddle after a while. So in our implementation, we multiply heat radiation by 0.98 after each iteration to reduce it.

III. Designing a Simple Neuro-Fuzzy System

In this section, we designed a neuro-fuzzy system model based on the Adaptive Neuro-Fuzzy Inference System (ANFIS) [23]. The purpose of designing this Sugeno type system is to prepare algorithms and create experimental space. This system used a simple dataset. In the experiment section, benchmark datasets are connected or replaced to this system.

To design the above mentioned neuro-fuzzy system model, we create a simple dataset. In this model, a series of values as Input x has been expanded regularly with an arithmetic progression in the range of $[0, 2\pi]$. The number of these values is 50 and Target t is $\sin x$. First, we have a basic fuzzy system that we create by using the Fuzzy c-means (FCM) [24] method. The basic fuzzy system generated using FCM by extracting a set of rules that models the data behavior. This case requires separate sets of input and output data as input arguments. This method will ultimately receive the inputs and targets and deliver the basic Fuzzy Inference System (FIS). The IF-THEN rule is widely used by the fuzzy inference system to compute the degree to which the input data matches the condition of a rule. The FCM-based design approach in this paper is that for each cluster, one rule is considered. In the designed system the number of rules does not change but the parameters of the rules can be changed. The number of clusters determines the number of rules and membership function in generated FIS. The type of input membership functions are Gaussian. The output membership function is also as a linear type. In this state, we have a basic fuzzy system that we can train it by applying the optimization algorithms. Assume that the basic FIS has a vector of numbers ($\overrightarrow{p^0}$), that the numbers are the parameters. Therefore we have $p_1^0, p_2^0, \dots, p_n^0$. If we want to change the values of the vector parameters $\overrightarrow{p^0}$ with an optimization algorithm, the range of changes is unknown. One of the ways that we can solve this issue is to consider p_1^0 as the basis, and we want to change the default value to the optimal values of $p_1, p_2, \dots, p_n$. We assume that $p_i = x_i p_i^0$, where p_i is the optimal value and is unknown. p_i^0 is the analogous parameter value in the basic FIS. In this case, for example, if $x_i = 1$, then p_i is equal to p_i^0. Therefore, the optimization algorithm will determine the value of x_i. With a simpler description, we want to extract all the parameters of membership functions from their own place, optimize them and set these optimized parameters in their own place. If we want to see the result accurately, we can implement the testing part with 1000 data. In this way, the error can be calculated. Fig 3, shows the phases of this fuzzy system by the flowchart.

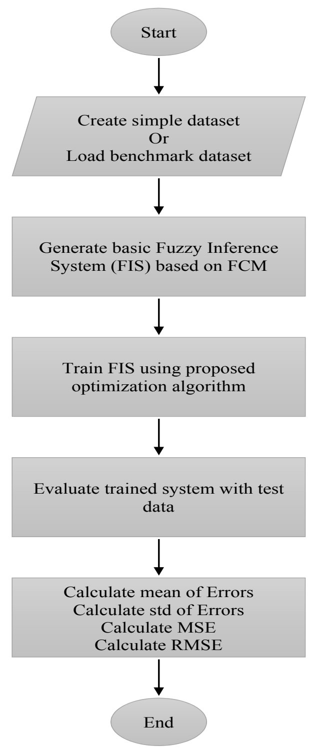

Fig. 3. Flowchart of designing a simple neuro-fuzzy system.

After designing the model, we run it. The basic FIS is created. Fig 4 (left) represents the basic FIS. The Target diagram (red dots) is clear in the figure. However, the Output diagram (blue line) of the model is very different from the Target diagram. Because the Output has not yet been trained. The training is done through the EPC optimization algorithm, which result is seen in Fig 4 (right). In the right figure, the Output and Target diagrams are exactly in the same path.

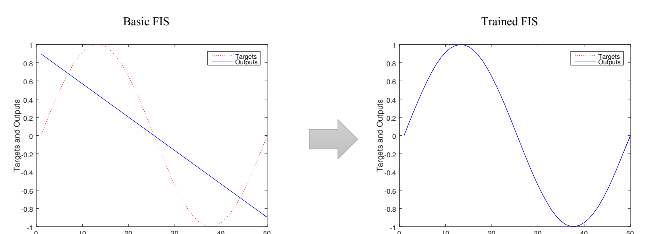

Fig. 4. The FIS before (left) and after (right) training using EPC optimization algorithm.

We want to evaluate the behavior of the fuzzy system with respect to the data. The training phase in this system is optimizing the behavior of the system to the behavior that should be. The general purpose of the system is to receive inputs and generate outputs. In Fig 5, Target is the output of the system, and Output is the output of the model. When the model accurately describes the system that for each input, Target and Output are equal. So, if we measure the difference between Target and Output, we get the Error. If for every input the Error is zero, it means that the model works exactly like the system, which in general is not possible. But the Error can be minimized.

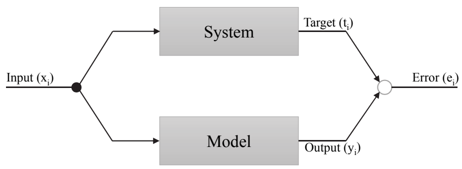

Fig. 5. Fuzzy system schema.

For each sample of Input x_i, Target t_i and Output y_i are available. From the calculation of the difference between t_i and y_i, the Error e_i is obtained ($e_i = t_i - y_i$). In this way, the Mean Square Error (MSE) for all possible instances can be written as Equation 5,

$$MSE = \frac{1}{N}\sum_{i=1}^{N} e_i^2 \tag{5}$$

Sometimes the MSE equation is also called the cost function, error function, and performance index. If the parameter settings change for the model, in fact, the behavior of the model becomes closer to the behavior of the system. The value of the cost function must be minimized. This is possible by changing the parameter settings. This phase is called the training phase. For directly interpretable in terms of measurement units, the Root Mean Square Error (RMSE) is also used. That is,

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_i^2} \tag{6}$$

After the evaluation, the result of RMSE at the training phase is 0.0016. In the testing phase, RMSE is 0.0015. This shows that training and testing are well done. Fig 6 shows the error diagram in the training and testing phase.

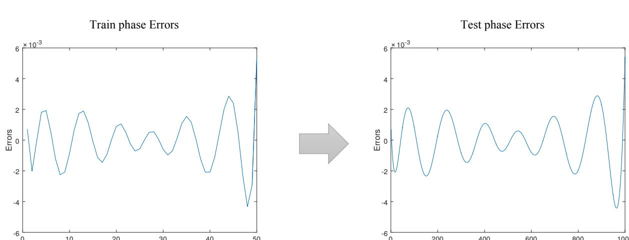

Fig. 6. Error diagram in the training phase (left) and the testing phase (right) obtained using EPC optimization algorithm.

In this section, a simple neuro-fuzzy system was created. The aim was the configuration and correct combination of the neuro-fuzzy system with the optimization algorithm. The system works correctly. Now the system is ready to experiment optimization algorithms by connecting benchmark datasets. In the next section, the details of the experiments are described.

IV. Experimental Results

This experiment section is divided into two parts. The first part (Part A) is based on the iterations of each algorithm. The second part (Part B) is based on the number of fitness function calls (NFC) or the number of function evaluations in the optimization algorithms.

Hence, four criteria have been used for evaluation of the iteration based experiment. As previously mentioned, if the Target is t_i, the Output is y_i, and the Error is e_i, the Error value is equal to $e_i = t_i - y_i$. So the first criterion is the mean of errors. The second criterion is the standard deviation of errors. Eventually, MSE and RMSE are the third and fourth criteria for experiments, respectively. For the NFC based experiment, in addition to the above criteria, the CPU time is also recorded.

To validate the performance of training, the six well-known algorithms are chosen for comparison and training the FIS. These are Genetic Algorithm (GA) [25], Particle Swarm Optimization (PSO) [26], Artificial Bee Colony (ABC) [27], Differential Evolution (DE) [28], Ant Colony Optimization (ACO) [29], and Invasive Weed Optimization (IWO) [30]. In addition to the above mentioned optimization algorithms, the ANFIS technique [23] was also used in experiments. ANFIS used a hybrid learning algorithm to identify parameters of Sugeno-type fuzzy inference systems. It applies a combination of the least-squares estimation method with the backpropagation gradient descent method for training FIS membership function parameters to emulate a given training dataset.

Also, seven real-world and benchmark datasets were selected for experiments. These datasets are shown in Table I. These selected datasets are Abalone Shell Rings, Body Fat Percentage, Breast Cancer Wisconsin, Chemical Sensor, House Pricing, Iris, and Wine, which are available in the repository of the machine learning databases [31].

TABLE I
Datasets Used for the Experiments

Dataset	Attributes	FCM-based intended clusters	Number of Parameters (Dimensions)
Abalone Shell Rings	8	10	250
Body Fat Percentage	13	10	400
Breast Cancer Wisconsin	9	2	56
Chemical Sensor	8	10	250
House Pricing	13	10	400
Iris	4	3	39
Wine	13	3	120

Table I shows the number of attributes, the number of FCM-based intended clusters, and the number of parameters available in the neuro-fuzzy system based on each dataset. In fact, the number of parameters also describes the dimensions of the problem. For example, in our implementation for Abalone Shell Rings dataset which has eight inputs, 10 clusters are considered based on FCM, and for each input 10 membership functions and 10 rules are considered. We have two parameters for input membership functions because it is Gaussian. In this way, input membership functions are 160. Also, we have three parameters for the output membership function because it is linear. So, the output membership function parameters are 90. Overall, there are 250 parameters that represent the dimensions.

In order that the experiment results be comparable, the settings of all algorithms are similar to each other. In the iteration based experiment, the number of iteration is considered 1000. This means that the algorithms stop as soon as they reach to iteration number 1000. In the NFC based experiment, if the number of fitness function calls reaches 15000, the algorithm stops. The initial population is considered 30 in all algorithms. All parameters settings of the EPC algorithm mentioned in Section II have been used for the experiments without changes. Some parameters are selected for some other algorithms through manual tuning. For example, the mutation and crossover rate in the GA are tune-up to get the best training. All evaluation experiments have been run on an Intel® Pentium® processor CPU G645 2.90 GHz with 2 GB RAM. Implementations have been run on MATLAB R2015b for coding.

To get reliable results, the collection of very diverse datasets are considered. There are small and simple or large and complex datasets in the selected datasets for experiments. For each dataset, 70% of their data are selected for training.

A. Iteration based experiment

As described in the previous section, assume that the basic fuzzy system is a FIS, which has a vector of numbers ($\overrightarrow{p^0}$), that the numbers are the parameters. So there are $p_1^0, p_2^0, \dots, p_n^0$. If we want to change the values of the vector parameters $\overrightarrow{p^0}$ with an optimization algorithm, the range of changes is unknown. We can consider p_1^0 as the basis, and now we want to change the default value to the optimal values of $p_1, p_2, \dots, p_n$. Assume that $p_i = x_i p_i^0$, where p_i is the optimal value and is unknown. p_i^0 is the analogous parameter value in the initial FIS as previously mentioned. For example, if $x_i = 1$, then p_i is equal to p_i^0. Therefore, the optimization algorithm will determine the value of x_i. Table II shows the results of four criteria obtained by applying optimization algorithms on benchmark datasets for the training phase.

Table II shows the results of training the neuro-fuzzy system using 70% of data. The ANFIS technique used in this paper,

TABLE II
THE RESULTS OF FOUR CRITERIA OBTAINED BY APPLYING OPTIMIZATION ALGORITHMS ON BENCHMARK DATASETS FOR TRAINING PHASE BASED ON ITERATION

Dataset	Criteria	Algorithms							
		ANFIS	GA	PSO	ABC	DE	ACO	IWO	**EPC**
Abalone Shell Rings	Mean of Errors	2.75e-08	0.0437	0.0045	0.0004	0.0032	0.0006	0.0053	0.0354
	StD of Errors	2.1342	2.2150	2.1029	2.2302	2.2459	2.2206	2.1210	2.0584
	MSE	4.5530	4.9064	4.4206	4.9719	5.0424	4.9294	4.4971	**4.2369**
	RMSE	2.1338	2.2150	2.1025	2.2298	2.2455	2.2202	2.1206	**2.0584**
Body Fat Percentage	Mean of Errors	-6.28e-05	0.2056	2.50e-14	-2.13e-15	9.75e-15	-5.39e-15	-1.05e-14	0.3484
	StD of Errors	3.8965	4.0045	4.0436	4.1883	4.1411	3.9246	3.7956	3.4545
	MSE	15.0964	15.9870	16.2582	17.4426	17.0512	15.3148	14.3249	**11.8670**
	RMSE	3.8854	3.9984	4.0321	4.1764	4.1293	3.9134	3.7848	**3.4449**
Breast Cancer Wisconsin	Mean of Errors	3.09e-09	0.0133	-0.0079	0.0232	-0.0059	-1.32e-15	-0.0003	-6.88e-06
	StD of Errors	0.1544	0.1723	0.1499	0.1911	0.1947	0.1904	0.1523	0.1270
	MSE	0.0238	0.0298	0.0225	0.0369	0.0378	0.0361	0.0231	**0.0161**
	RMSE	0.1542	0.1726	0.1500	0.1923	0.1946	0.1902	0.1521	**0.1269**
Chemical Sensor	Mean of Errors	-4.8424	-0.1118	5.62e-14	1.65e-13	0.0090	1.12e-14	-6.35e-14	0.0147
	StD of Errors	31.3137	2.2561	2.2593	2.3753	2.3002	2.4688	2.4169	1.9839
	MSE	1001.1843	5.0881	5.0898	5.6259	5.2758	6.0776	5.8247	**3.9246**
	RMSE	31.6415	2.2557	2.2561	2.3719	2.2969	2.4653	2.4134	**1.9811**
House Pricing	Mean of Errors	3.44e-06	-0.0772	-0.0069	6.82e-15	-0.0701	3.73e-15	-0.0476	0.0197
	StD of Errors	3.0636	4.3918	3.2515	4.7533	4.3101	4.6696	3.5389	2.9222
	MSE	9.3590	19.2391	10.5424	22.5298	18.5296	21.7440	12.4908	**8.5157**
	RMSE	3.0593	4.3862	3.2469	4.7466	4.3046	4.6630	3.5342	**2.9182**
Iris	Mean of Errors	5.80e-09	0.0043	0.0001	-2.66e-17	0.0143	1.98e-16	0.0015	-1.37e-07
	StD of Errors	0.0160	0.0488	0.0356	0.1434	0.1315	0.1448	0.0342	1.94e-05
	MSE	0.0002	0.0023	0.0012	0.0203	0.0173	0.0207	0.0011	**3.71e-10**
	RMSE	0.0160	0.0487	0.0354	0.1428	0.1316	0.1441	0.0341	**1.93e-05**
Wine	Mean of Errors	9.49e-07	-0.0150	8.08e-05	2.13e-17	0.0007	-7.67e-16	0.0006	-0.0002
	StD of Errors	0.0432	0.1404	0.1462	0.1606	0.1642	0.1707	0.0827	0.0298
	MSE	0.0018	0.0197	0.0212	0.0256	0.0267	0.0289	0.0067	**0.0008**
	RMSE	0.0430	0.1406	0.1456	0.1600	0.1636	0.1700	0.0823	**0.0297**

which uses the hybrid optimization algorithm, has good results. The PSO and IWO algorithms also have acceptable results and are better than ANFIS. The GA is average. It seems that the ABC, DE and ACO algorithms are not suitable for the model presented in this paper. Maybe these algorithms need to change their settings for use in this type of problem. Therefore, checking the suitability of these algorithms can be the subject of further researches. However, these algorithms do not provide good results. The proposed EPC algorithm has very good results. This algorithm has the lowest RMSE in seven out of seven datasets. Fig 7 shows the difference between Targets and Outputs in the data training state for each dataset obtained by the EPC optimization algorithm.

The testing phase is the remaining part of the data used to provide an unbiased evaluation of a final model fit on the training dataset. If a model fit to the training dataset also fits the testing dataset well, minimal over fitting has taken place. A better fitting of the training dataset, as opposed to the testing dataset, usually points to over fitting. Table III shows the results of four criteria obtained by applying optimization algorithms on benchmark datasets for the testing phase.

The results obtained by the EPC optimization algorithm are good and acceptable. Although the results can be improved. For example, some techniques such as cross-validation can be used and can be the subject of further researches. However, the results show that the EPC algorithm is also successful in the testing phase. This algorithm has the lowest rate of error in seven out of seven datasets. The ANFIS technique failed to perform the testing phase. After EPC, the IWO and PSO algorithms have good performance. The genetic algorithm has not provided the expected results. Other algorithms such as ACO, DE, and ABC do not provide acceptable results. Fig 8 shows the Error graph in data testing mode for each dataset obtained by EPC optimization algorithm.

B. *NFC based experiment*

As mentioned earlier, we once again performed the

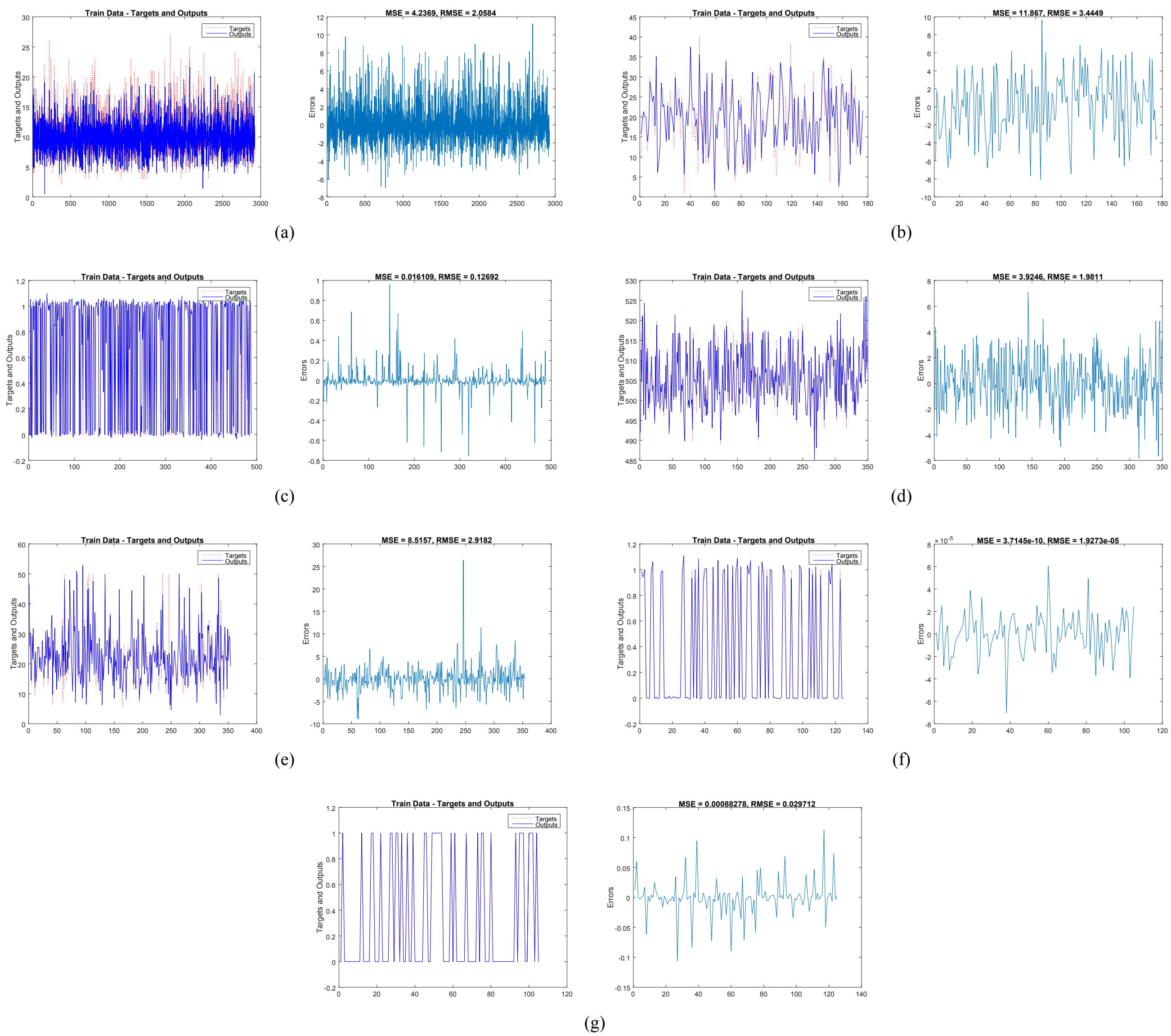

Fig. 7. Difference between Targets, Outputs and Errors in data training mode by applying EPC optimization algorithm. (a) is Abalone Shell Rings dataset. (b) is Body Fat Percentage dataset. (c) is Breast Cancer Wisconsin dataset. (d) is Chemical Sensor dataset. (e) is House Pricing dataset. (f) is Iris dataset. (g) is Wine dataset. The results obtained by iteration based experiment.

TABLE III
THE RESULTS OF FOUR CRITERIA OBTAINED BY APPLYING OPTIMIZATION ALGORITHMS ON BENCHMARK DATASETS FOR TESTING PHASE BASED ON ITERATION

Dataset	Criteria	Algorithms							
		ANFIS	GA	PSO	ABC	DE	ACO	IWO	**EPC**
Abalone Shell Rings	Mean of Errors	0.0311	-0.0031	-0.0218	-0.0538	-0.0360	0.0867	-0.0738	-0.0475
	StD of Errors	2.2311	2.1196	2.1242	2.1821	2.1420	2.2028	2.1032	2.0130
	MSE	4.9747	4.4891	4.5092	4.7607	4.5856	4.8561	4.4254	**4.0510**
	RMSE	2.2304	2.1188	2.1235	2.1819	2.1414	2.2036	2.1037	**2.0127**
Body Fat Percentage	Mean of Errors	-0.0708	-0.5682	0.7012	-0.6549	-0.2461	-0.7238	-0.6815	-0.2470
	StD of Errors	4.4219	7.2811	4.8134	5.4386	4.7477	5.4551	5.1754	4.1623
	MSE	19.3010	52.6393	23.3557	29.6179	22.3044	29.8908	26.8971	**17.1579**
	RMSE	4.3933	7.2553	4.8328	5.4422	4.7227	5.4673	5.1862	**4.1422**
Breast Cancer Wisconsin	Mean of Errors	0.0020	0.0089	-0.0160	-0.0031	-0.0530	0.0080	0.0081	-0.0208
	StD of Errors	0.1994	0.1965	0.1863	0.1872	0.1797	0.1991	0.1751	0.1564
	MSE	0.0396	0.0385	0.0348	0.0348	0.0349	0.0395	0.0305	**0.0247**
	RMSE	0.1990	0.1962	0.1865	0.1867	0.1870	0.1988	0.1749	**0.1574**
Chemical Sensor	Mean of Errors	0.7572	-0.2147	0.0332	0.2612	-0.6033	0.1465	0.0416	0.2130
	StD of Errors	43.1220	2.4416	2.7092	2.3574	2.4796	2.5414	2.2799	2.1127
	MSE	1847.60	5.9673	7.2913	5.5885	6.4710	6.4367	5.1649	**4.4788**
	RMSE	42.9837	2.4428	2.7002	2.3640	2.5438	2.5371	2.2726	**2.1163**
House Pricing	Mean of Errors	0.9332	-0.0387	0.0433	-0.3015	0.1935	0.3160	0.2217	0.1374
	StD of Errors	21.3118	4.8672	4.4899	4.6635	5.4785	4.7637	4.6083	3.5604
	MSE	436.5705	23.5353	20.0285	21.6960	29.8537	22.6433	21.1460	**12.6121**
	RMSE	20.8943	4.8513	4.4753	4.6579	5.4639	4.7585	4.5985	**3.5513**
Iris	Mean of Errors	0.0015	0.0232	-0.0125	0.0582	0.0083	-0.0011	0.0054	4.38e-06
	StD of Errors	0.0130	0.1084	0.0291	0.1386	0.1504	0.1481	0.0523	2.35e-05
	MSE	0.0001	0.0120	0.0009	0.0221	0.0222	0.0214	0.0027	**5.60e-10**
	RMSE	0.0129	0.1097	0.0313	0.1489	0.1490	0.1465	0.0520	**2.37e-05**
Wine	Mean of Errors	0.1413	0.0278	0.0115	-0.0021	0.0183	0.0044	-0.0591	-0.0107
	StD of Errors	0.9438	0.1792	0.2133	0.2105	0.2018	0.1956	0.2310	0.1170
	MSE	0.8940	0.0322	0.0447	0.0435	0.0403	0.0375	0.0558	**0.0135**
	RMSE	0.9455	0.1797	0.2115	0.2085	0.2008	0.1938	0.2363	**0.1164**

experiment based on NFC. Also in this type of experiment, CPU time is recorded. In addition, the stop condition is reached 15,000 calls. Given the type of issue, this number seemed appropriate. The goal was for the algorithms to have enough time to solve the problem. This enables them to be better analyzed. In this type of issue, the lower number of fitness function calls in could have made the algorithms fail to perform well.

The NFC based experiment was performed only between the optimization algorithms used in this paper. Also, 70% of the

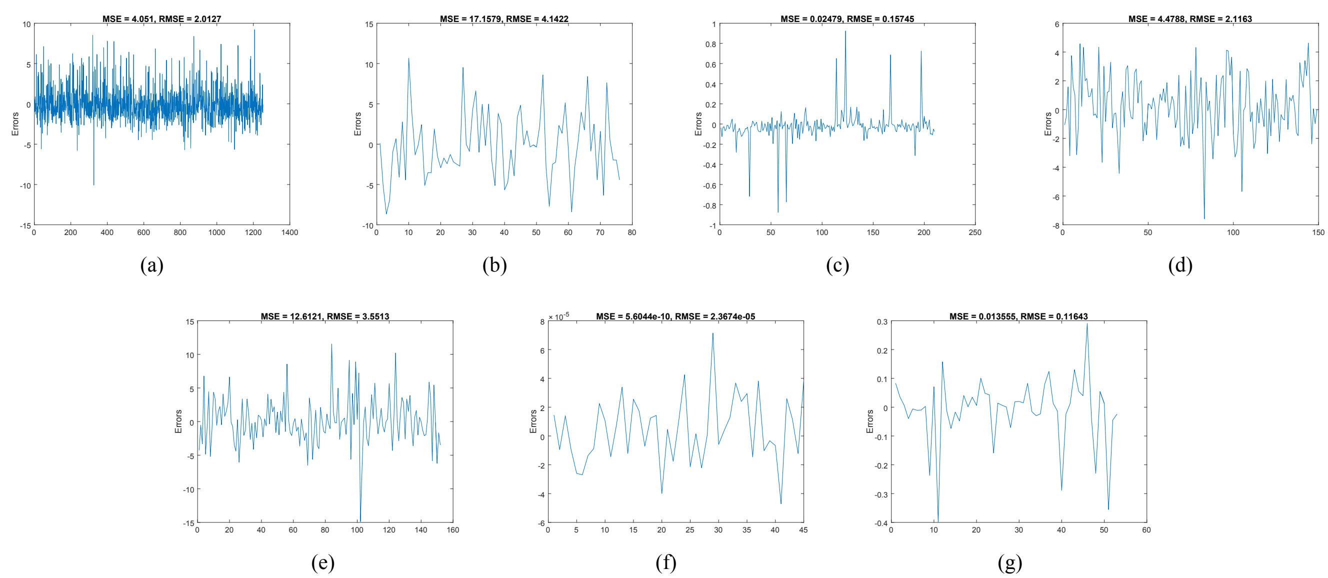

Fig. 8. The Error graph in data testing mode by applying EPC optimization algorithm. (a) is Abalone Shell Rings dataset. (b) is Body Fat Percentage dataset. (c) is Breast Cancer Wisconsin dataset. (d) is Chemical Sensor dataset. (e) is House Pricing dataset. (f) is Iris dataset. (g) is Wine dataset.

TABLE IV
THE RESULTS OF FOUR CRITERIA OBTAINED BY APPLYING OPTIMIZATION ALGORITHMS ON BENCHMARK DATASETS FOR TRAINING PHASE BASED ON NFC

Dataset	Criteria	Algorithms						
		GA	PSO	ABC	DE	ACO	IWO	**EPC**
Abalone Shell Rings	Mean of Errors	-0.2408	-9.90e-05	0.0012	0.0132	-3.75e-15	0.0204	0.0085
	StD of Errors	2.1239	2.1451	2.2331	2.2151	2.1538	2.1102	2.0702
	MSE	4.5654	4.5998	4.9852	4.9051	4.6374	4.4517	**4.2842**
	RMSE	2.1367	2.1447	2.2327	2.2147	2.1535	2.1099	**2.0698**
Body Fat Percentage	Mean of Errors	1.72e-14	5.61e-15	9.29e-15	-6.0053e-15	0.4056	0.0056	0.2522
	StD of Errors	4.1001	4.0613	3.9883	4.1481	3.8814	4.1462	3.6472
	MSE	16.715	16.4006	15.8158	17.1089	15.1439	17.0930	**13.2899**
	RMSE	4.0884	4.0498	3.9769	4.1363	3.6915	4.1344	**3.6455**
Breast Cancer Wisconsin	Mean of Errors	-0.0191	-0.0002	-0.0001	-0.0014	-1.64e-15	-1.48e-15	-0.0052
	StD of Errors	0.1826	0.1677	0.1560	0.1843	0.1968	0.1921	0.1254
	MSE	0.0336	0.0280	0.0243	0.0339	0.0386	0.0368	**0.0157**
	RMSE	0.1834	0.1675	0.1559	0.1841	0.1966	0.1919	**0.1254**
Chemical Sensor	Mean of Errors	0.1527	1.61e-13	-4.73e-14	1.62e-15	-2.30e-13	-0.0087	-2.23e-14
	StD of Errors	0.3253	2.3856	2.3832	2.3502	2.4335	2.4088	2.2469
	MSE	5.4151	5.6748	5.6633	5.5074	5.9049	5.7859	**5.0341**
	RMSE	2.3270	2.3822	2.3798	2.3468	2.4300	2.4054	**2.2437**
House Pricing	Mean of Errors	0.0105	-0.0448	2.69e-14	0.1347	2.99e-15	-0.0447	0.1587
	StD of Errors	4.1845	2.9421	4.2503	4.7326	4.8694	3.7607	3.3601
	MSE	17.4607	**8.6335**	18.0139	22.3526	23.6443	14.1046	11.2834
	RMSE	4.1786	**2.9383**	4.2443	4.7279	4.8625	3.7556	3.3591
Iris	Mean of Errors	0.0019	0.0004	4.67e-17	-0.0207	1.14e-16	0.0008	-0.0017
	StD of Errors	0.0903	0.0236	0.1396	0.1272	0.1409	0.0380	0.0183
	MSE	0.0080	0.0005	0.0193	0.0164	0.0196	0.0014	**0.0003**
	RMSE	0.0899	0.0235	0.1389	0.1283	0.1402	0.0378	**0.0183**
Wine	Mean of Errors	0.0003	0.0001	-2.52e-16	0.0028	1.66e-16	-0.0010	0.0015
	StD of Errors	0.1661	0.1355	0.1686	0.1610	0.1709	0.1063	0.1030
	MSE	0.0273	0.0182	0.0282	0.0257	0.0289	0.0112	**0.0105**
	RMSE	0.1654	0.1350	0.1680	0.1604	0.1702	0.1059	**0.1026**

data was used to training the neuro-fuzzy system. The training and testing results are shown in Tables IV and V. In the training phase, as Table IV shows, the lowest RMSE value is related to the EPC algorithm. This algorithm was able to record the best performance in six datasets. Only on the House Pricing dataset, the PSO algorithm is better than other algorithms. However, in this case, the EPC algorithm is better than others after the PSO. GA and IWO algorithms also have acceptable results. It is clear that the ABC, DE and ACO algorithms did not provide good answers. The error rates in these algorithms are high. For the Iris dataset, which is a simple dataset, the performance of the proposed algorithm is very well. This shows that this algorithm performs well in the training phase with respect to NFC. Of course, the results for more complex datasets such as Abalone Shell Rings, Body Fat Percentage and Chemical Sensor are also favorable.

In the testing phase, we also obtained the expected results. In Table V, which shows the testing phase, the best overall performance is related to the EPC algorithm. This algorithm has been successful in six datasets. Only on the Wine dataset, the IWO algorithm is better and after IWO the EPC has good performance. The GA and PSO algorithms also worked mediocrity in the testing phase. According to observations similar to the training phase, the ABC, DE and ACO algorithms do not perform well. It is obvious that they are not suitable for such issues. The EPC algorithm has also been successful in solving complex datasets. In addition to the acceptable results, the CPU consumption of the EPC algorithm is appropriate. Table VI shows the total time consumed by the CPU when calling the fitness function. This time is in seconds. The recorded time of the EPC algorithm in five datasets out of seven datasets is less than the other algorithms. Only for two datasets, the CPU consumption of the GA is better than the other algorithms.

TABLE VI
CPU CONSUMING BETWEEN OPTIMIZATION ALGORITHMS

Dataset	CPU Time for Algorithms						
	GA	PSO	ABC	DE	ACO	IWO	**EPC**
Abalone Shell Rings	**203.17**	210.69	223.05	234.38	226.04	224.88	205.47
Body Fat Percentage	**81.413**	84.600	85.253	85.355	84.527	83.186	83.803
Breast Cancer	27.455	28.513	27.094	27.832	27.518	27.656	**26.254**
Chemical Sensor	66.892	66.047	66.481	66.895	65.985	68.034	**65.374**
House Pricing	100.27	98.733	99.283	100.14	98.867	99.304	**94.753**
Iris	18.903	18.706	18.963	19.516	18.676	18.622	**18.259**
Wine	33.285	33.155	33.730	33.771	33.249	32.939	**32.236**

TABLE V
THE RESULTS OF FOUR CRITERIA OBTAINED BY APPLYING OPTIMIZATION ALGORITHMS ON BENCHMARK DATASETS FOR TESTING PHASE BASED ON NFC

Dataset	Criteria	Algorithms						
		GA	PSO	ABC	DE	ACO	IWO	**EPC**
Abalone Shell Rings	Mean of Errors	-0.0968	0.1107	-0.0304	0.0420	0.0510	-0.0434	-0.0464
	StD of Errors	2.2033	2.2954	2.2568	2.2093	2.3229	2.2643	2.1685
	MSE	4.8625	5.2767	5.0902	4.8789	5.3942	5.1248	**4.7006**
	RMSE	2.2051	2.2971	2.2561	2.2088	2.3225	2.2638	**2.1681**
Body Fat Percentage	Mean of Errors	1.5767	0.3620	-0.8899	-0.4370	0.0256	0.0653	-0.2249
	StD of Errors	4.7636	4.5130	4.8910	4.4887	4.5166	4.6142	4.4532
	MSE	24.8792	20.2300	24.3994	20.0776	20.1323	21.0154	**19.6204**
	RMSE	4.9879	4.4978	4.9396	4.4805	4.4869	4.5843	**4.4295**
Breast Cancer Wisconsin	Mean of Errors	-0.0332	-0.0001	0.0100	-0.0041	0.0053	-0.0152	-0.0038
	StD of Errors	0.1873	0.2113	0.1743	0.2135	0.1833	0.1946	0.1465
	MSE	0.0360	0.0444	0.0303	0.0454	0.0334	0.0379	**0.0213**
	RMSE	0.1898	0.2108	0.1742	0.2131	0.1830	0.1947	**0.1462**
Chemical Sensor	Mean of Errors	-0.0614	-0.1861	0.1424	-0.7226	-0.3555	0.2371	0.1191
	StD of Errors	2.3426	2.3268	2.6291	2.3286	2.2418	2.2677	2.2145
	MSE	5.4547	5.4121	6.8859	5.9084	5.1185	5.1643	**4.8852**
	RMSE	2.3355	2.3264	2.6241	2.4307	2.2624	2.2725	**2.2102**
House Pricing	Mean of Errors	-0.1218	0.1865	0.2610	0.2030	0.7067	-0.1473	-0.8033
	StD of Errors	5.0514	4.9265	4.9445	4.8004	5.6746	5.3433	4.2855
	MSE	25.3633	24.1457	24.3554	22.9336	32.4889	28.3851	**18.8897**
	RMSE	5.0362	4.9138	4.9351	4.7889	5.6999	5.3278	**4.3462**
Iris	Mean of Errors	-0.0155	-0.0084	-0.0045	-0.0341	0.0010	-0.0012	0.0026
	StD of Errors	0.0938	0.0388	0.1563	0.1552	0.1538	0.0355	0.0206
	MSE	0.0088	0.0015	0.0239	0.0247	0.0231	0.0012	**0.0004**
	RMSE	0.0941	0.0392	0.1547	0.1572	0.1521	0.0351	**0.0206**
Wine	Mean of Errors	0.0031	0.0078	-0.0724	0.0161	-0.0060	0.0267	-0.0161
	StD of Errors	0.1858	0.1776	0.1935	0.2042	0.1945	0.1298	0.1495
	MSE	0.0338	0.0310	0.0419	0.0411	0.0371	**0.0172**	0.0221
	RMSE	0.1841	0.1761	0.2049	0.2029	0.1927	**0.1313**	0.1489

Somewhat, by observing the results table for the experiments, we can guess that the EPC algorithm is better than others. But without statistical analysis, it is impossible to accurately distinguish between them. In the next section, we used the statistical analysis to determine the difference between the algorithms.

V. STATISTICAL ANALYSIS

To find significant differences between the results obtained by algorithms, statistical analysis is used. This section is divided into two parts. The first part (Part A) is statistical analysis related to the results obtained from the iteration based experiment. The second part (Part B) is the NFC based experiment statistical analysis.

A. Statistical analysis for iteration based experiment

To detect significant differences in the results, Friedman and Iman-Davenport tests are employed [32, 33]. Table VII shows the ranking of optimization algorithms based on the RMSE criterion using the Friedman test in the iteration based experiment. As expected, the EPC algorithm is first in the ranking, then the IWO algorithm is located. In the next ranks, the algorithms are PSO, ANFIS technique, GA, DE, ACO, and ABC, respectively.

Table VIII shows the iteration based results of the Friedman and Iman-Davenport tests. In this table, there is the Chi-Square value with seven degrees of freedom, and also there is the asymptotic significance of the test (p-value) with very close to zero value. Being close to the zero value of the asymptotic significance, the hypothesis is rejected. Therefore, it can be concluded that there is a significant difference in the performance of algorithms.

TABLE VII
RANKING OF THE ALGORITHMS BASED ON THE RMSE FOR TRAINING PHASE IN ITERATION BASED EXPERIMENT

	Algorithms							
	ANFIS	GA	PSO	ABC	DE	ACO	IWO	**EPC**
Ranking	3.57	4.57	3.57	6.86	6.43	6.57	3.43	**1.00**

TABLE VIII
RESULTS OF FRIEDMAN'S AND IMAN–DAVENPORT'S TESTS BASED ON RMSE FOR TRAINING PHASE IN ITERATION BASED EXPERIMENT

Test method	Chi-Square	Degrees of freedom (DF)	*p*-Value	Hypothesis
Friedman	33.4761	7	2.16e-05	Rejected
Iman–Davenport	12.9390	7	9.92e-09	Rejected

Because a significant difference has been observed, Holm's method is used as a post-hoc test to obtain the results of best performance compared to other performances. In this method, according to Friedman's rank, the best rank is the EPC algorithm which is considered as a control algorithm and the confidence interval is 95% ($\alpha = 0.05$). The results are shown in Table IX. In the IWO, PSO, and ANFIS technique cases, there is no significant difference based on the Holm's method results. However, the training results reported in Table II show that the proposed EPC approach outperforms IWO, PSO, and ANFIS technique in seven out of seven datasets.

TABLE IX
RESULTS OF THE HOLM'S METHOD BASED ON THE RMSE FOR TRAINING PHASE IN ITERATION BASED EXPERIMENT (EPC IS THE CONTROL ALGORITHM)

Algorithm	*j*	α/*j*	*z*-Score	*p*-Value	Hypothesis
IWO	1	0.0500	1.855943	0.063461	Not Rejected
PSO	2	0.0250	1.962869	0.049661	Not Rejected
ANFIS	3	0.0166	1.962869	0.049661	Not Rejected
GA	4	0.0125	2.726632	0.006398	Rejected
DE	5	0.0100	4.147231	0.000033	Rejected
ACO	6	0.0083	4.254157	2.100e-05	Rejected
ABC	7	0.0071	4.475648	7.600e-06	Rejected

A similar procedure was performed to prove significant differences in RMSE obtained in the testing phase. Table X shows the ranking of optimization algorithms based on the RMSE criterion (the RMSE results of Table III) using the Friedman test. In Table X, the best rank is related to the EPC algorithm. Table XI shows the results of the Friedman and Iman-Davenport tests based on the RMSE criterion in the iteration based experiment. In this table, the hypothesis is also rejected according to the p-value. There is a significant difference in the error rates. Also, Table XII shows the results of Holm's method regarding the RMSE obtained in the testing phase. The results obtained by Friedman's test indicate that the EPC algorithm is ranked first and there are significant differences in the results of the algorithms. Moreover, from the results of the Holm's method in Table XII, it could be concluded that the control algorithm (EPC) in the testing phase performs significantly better regarding the RMSE than the remaining algorithms, with a significant level of 0.05.

TABLE X
RANKING OF THE ALGORITHMS BASED ON THE RMSE FOR TESTING PHASE IN ITERATION BASED EXPERIMENT

	Algorithms							
	ANFIS	GA	PSO	ABC	DE	ACO	IWO	**EPC**
Ranking	6.29	4.86	4.14	5.00	5.43	5.71	3.57	**1.00**

TABLE XI
RESULTS OF FRIEDMAN'S AND IMAN–DAVENPORT'S TESTS BASED ON RMSE FOR TESTING PHASE IN ITERATION BASED EXPERIMENT

Test method	Chi-Square	Degrees of freedom (DF)	*p*-Value	Hypothesis
Friedman	22.3333	7	0.0022	Rejected
Iman–Davenport	5.0250	7	0.0003	Rejected

TABLE XII
RESULTS OF THE HOLM'S METHOD BASED ON THE RMSE FOR TESTING PHASE IN ITERATION BASED EXPERIMENT (EPC IS THE CONTROL ALGORITHM)

Algorithm	*j*	α/*j*	*z*-Score	*p*-Value	Hypothesis
IWO	1	0.0500	1.962869	0.049661	Rejected
PSO	2	0.0250	2.398214	0.016475	Rejected
GA	3	0.0166	2.948123	0.003197	Rejected
ABC	4	0.0125	3.055050	0.002250	Rejected
DE	5	0.0100	3.383468	0.000715	Rejected
ACO	6	0.0083	3.597321	3.215e-04	Rejected
ANFIS	7	0.0071	4.040304	5.340e-05	Rejected

B. Statistical analysis for NFC based experiment

Once again, the same procedure was performed for NFC based experiment. Similar to the iteration based experiment to find significant differences between algorithms, the Friedman and Iman-Davenport tests are used. Table XIII shows the rankings obtained through the Friedman test. This ranking is calculated based on the RMSE criterion that exists in Table IV. In this ranking, the EPC algorithm is located in the first rank and after EPC, the PSO algorithm is placed. In the next ranks, the algorithms are GA, IWO, ABC, DE, ACO, respectively. Just like the iteration based experiment, the ABC, DE and ACO algorithms are at the last ranks. Table XIV shows the results of the Friedman and Iman-Davenport tests in NFC based experiment. In this table, there is the Chi-Square value with 6 degrees of freedom, and also there is the asymptotic significance of the test (p-value) with very close to zero value. So the hypothesis is rejected. Therefore, it can be concluded that there is a significant difference in the performance of algorithms. The existence of a significant difference cause that Holm's method is also used for this part. Holm's method is performed based on Friedman's ranking. Here, the best rank (the EPC algorithm) is chosen as the control algorithm. Also, like before, confidence interval is 95% (α = 0.05).

TABLE XIII
RANKING OF THE ALGORITHMS BASED ON THE RMSE FOR TRAINING PHASE IN NFC BASED EXPERIMENT

	Algorithms						
	GA	PSO	ABC	DE	ACO	IWO	**EPC**
Ranking	3.86	3.14	4.71	5.14	6.00	4.00	**1.14**

TABLE XIV
RESULTS OF FRIEDMAN'S AND IMAN–DAVENPORT'S TESTS BASED ON RMSE FOR TRAINING PHASE IN NFC BASED EXPERIMENT

Test method	Chi-Square	Degrees of freedom (DF)	*p*-Value	Hypothesis
Friedman	22.1020	6	0.00116	Rejected
Iman–Davenport	6.6646	6	8.36e-05	Rejected

The results are shown in Table XV. The results show that there is a significant difference between the EPC algorithm and the others except the PSO. In the PSO case, there is no significant difference based on Holm's method results.

However, we observed that the EPC in much more better than PSO based on the results of Table IV. GA and IWO are mediocrity based on NFC in the training phase.

TABLE XV
RESULTS OF THE HOLM'S METHOD BASED ON THE RMSE FOR TRAINING PHASE IN NFC BASED EXPERIMENT (EPC IS THE CONTROL ALGORITHM)

Algorithm	j	α/j	z-Score	p-Value	Hypothesis
PSO	1	0.0500	1.732050	0.083274	Not Rejected
GA	2	0.0250	2.355589	0.018498	Rejected
IWO	3	0.0166	2.476832	0.013257	Rejected
ABC	4	0.0125	3.091710	0.00199	Rejected
DE	5	0.0100	3.464101	0.000532	Rejected
ACO	6	0.0083	4.208883	2.60e-05	Rejected

Table XVI shows the ranking of optimization algorithms based on the RMSE criterion (the RMSE results of Table V) using the Friedman test. In Table XVI, the best rank is again related to the EPC algorithm. The IWO algorithm is in the next rank. Table XVII shows the results of the Friedman and Iman-Davenport tests based on the RMSE criterion. In this table, the hypothesis is also rejected according to the p-value. There is a significant difference in the error rates. Also, Table XVIII shows the results of Holm's method. The results of Holm's method show that there is a significant difference between EPC and the others in the testing phase related to NFC based experiment.

TABLE XVI
RANKING OF THE ALGORITHMS BASED ON THE RMSE FOR TESTING PHASE IN NFC BASED EXPERIMENT

	Algorithms						
	GA	PSO	ABC	DE	ACO	IWO	**EPC**
Ranking	4.43	4.14	5.14	4.71	4.57	3.86	**1.14**

TABLE XVII
RESULTS OF FRIEDMAN'S AND IMAN–DAVENPORT'S TESTS BASED ON RMSE FOR TESTING PHASE IN NFC BASED EXPERIMENT

Test method	Chi-Square	Degrees of freedom (DF)	p-Value	Hypothesis
Friedman	15.7959	6	0.01489	Rejected
Iman–Davenport	3.6168	6	0.00655	Rejected

TABLE XVIII
RESULTS OF THE HOLM'S METHOD BASED ON THE RMSE FOR TESTING PHASE IN NFC BASED EXPERIMENT (EPC IS THE CONTROL ALGORITHM)

Algorithm	j	α/j	z-Score	p-Value	Hypothesis
IWO	1	0.0500	2.355589	0.018498	Rejected
PSO	2	0.0250	2.598076	0.009377	Rejected
GA	3	0.0166	2.849223	0.004383	Rejected
ACO	4	0.0125	2.970467	0.002974	Rejected
DE	5	0.0100	3.091710	0.00199	Rejected
ABC	6	0.0083	3.464101	5.32e-04	Rejected

VI. SOLVING A CLASSIC ENGINEERING PROBLEM

An inverted pendulum is a pendulum that has its center of mass above its pivot point. It is unstable and without additional help will fall over. A pendulum with its bob hanging directly below the support pivot is at a stable equilibrium point. There is no torque on the pendulum so it will remain motionless, and if displaced from this position will experience a restoring torque that returns it toward the equilibrium position. Fig 9 shows a schematic drawing of the inverted pendulum on a cart.

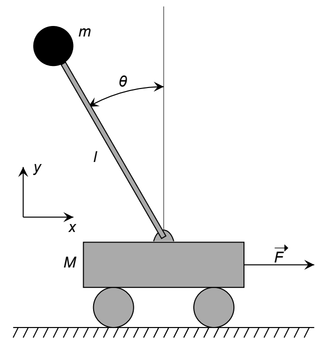

Fig. 9. A schematic drawing of the inverted pendulum on a cart.

The state variables of this problem are: the movement with acceleration in the direction of the x-axis, that is, the amount of movement (x_1); Derivative of movement or velocity (x_2); The angle (x_3) of the inverted pendulum from the y-axis; Angular acceleration of the pendulum (x_4). Equation 7 specifies these four state variables.

$$x_1 = x, \qquad x_2 = \dot{x}, \qquad x_3 = \theta, \qquad x_4 = \dot{\theta} \tag{7}$$

In fact, this is a quadratic system with four state variables. Equation 8 shows the dynamical equation of this system,

$$\begin{gathered} \dot{x}_1 = \dot{x} = x_2 \\ \dot{x}_2 = \ddot{x} \\ = \frac{-mg\sin x_3 \cos x_3 + mlx_4^2 \sin x_3 + f_\theta m x_4 \cos x_3 + F}{M + (1-\cos^2 x_3)m} \\ \dot{x}_3 = \dot{\theta} = x_4 \\ \dot{x}_4 \\ = \frac{(M+m)(g\sin x_3 - f_\theta x_4) - (lmx_4^2 \sin x_3 + F)\cos x_3}{l(M + (1-\cos^2 x_3)m)} \end{gathered} \tag{8}$$

where M is the inverted pendulum mass. l is the pendulum length. f_θ is the amount of friction in the pendulum movement. F it is the normal force applied to the cart. M is the mass of cart. In this problem, if the angle is not zero, which means there is an error. Problem inputs are error (θ) and derivative of error. The output is the force F. In this specific problem, determining the number of rules is different. In order to enable us to fine-tune this particular system, for each input, we have five membership functions, and in total, there are 25 rules. The fuzzy rules is based on Table XIX. To begin the optimization, the amount of 0.2 was determined for θ. Fig 10 shows the results obtained by applying EPC and other optimization algorithms to this fuzzy problem.

TABLE XIX
FUZZY RULES OF INVERTED PENDULUM

$\delta\theta$	θ				
	NB	N	Z	P	PB
NB	NB	NM	NS	NS	ZE
N	NM	NS	NS	ZE	PS
ZE	NS	NS	ZE	PS	PS
P	NS	ZE	PS	PS	PM
PB	ZE	PS	PS	PM	PB

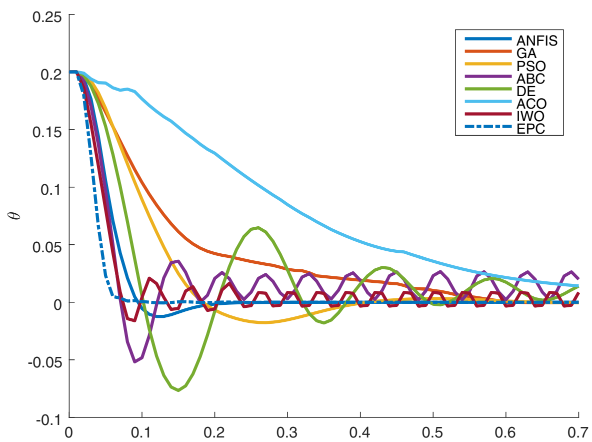

Fig. 10. Results obtained by applying EPC and other optimization algorithm to inverted pendulum problem.

Fig 10 shows that the performance obtained by the EPC algorithm is better than other optimization algorithms. The EPC algorithm quickly shifts the angle θ to zero and determines the force F faster than other algorithms.

VII. Discussion

The combination of neural networks and fuzzy logic models in ANFIS form has many advantages. These advantages include solving complex problems and solving nonlinear problems. Because of the acceptable results, ANFIS is used in a wide range of applications such as classification or rule-based process control, and so on. Also, due to the existence of fuzzy logic and neural network, this model has strong computational complexity. ANFIS uses hybrid learning techniques. As mentioned, these techniques are a combination of backpropagation gradient descent and least-squares estimation. Due to the use of gradient descent, it may get caught in the local trap. To deal with this problem, one of the best way is to use metaheuristic algorithms. Metaheuristic algorithms have been used to solve many real-world problems. Also, these algorithms have been very successful in solving problems. These algorithms can be used to better training the ANFIS. Using a population-based algorithm to optimize the fuzzy system parameters could give an improvement to fuzzy system accuracy.

As shown in the experiments, this paper used GA, PSO, ABC, DE, ACO, IWO and EPC optimization algorithms for ANFIS training. In different researches, GA and PSO based methods have been used more frequently. However, there are other algorithms that perform better than GA and PSO. There are many metaheuristic algorithms and all of them cannot be compared in one specific research. So this issue is a research gap and comparing different algorithms which have been less investigated can be the subject of further researches. This paper clearly shows that the EPC algorithm performs very well. The less commonly used IWO algorithm also has acceptable performance. Even better in some cases than GA and PSO. The GA and PSO algorithms have always been successful in solving various problems because of their improvements during the time. However, it is better to do more research on new algorithms like EPC, because they also have the potential to improve.

Occasionally with slight iterations and sometimes with much iterations we may reach an appropriate threshold for training. But the purpose of this paper was to use a certain number of iterations to compare the algorithms. So, experimenting with the exact number of iterations can be the subject of further researches. Reducing the error rate helps the system to be well trained. This means that, in the testing phase, the robust results can be achieved. Sometimes some algorithms cannot replace the optimal parameters. This means that the algorithm is not capable of dealing with the problem. From the selected algorithms in this paper, the ABC, DE and ACO algorithms are incapable of solving the problem. The PSO and GA algorithms, which are widely used in research, are capable of solving the designed system, but they did not surprise us much. They were very mediocrity. In order to get better results from these two algorithms, it may be necessary to increase the share of their training data. The IWO algorithm performed better than GA and PSO. We knew that the proposed algorithm, namely EPC, had many potentials. We also obtained the expected results. This algorithm was quite successful in optimizing the fuzzy system parameters. This shows that the spiral-like movement used in this algorithm was quite effective. The EPC algorithm was more successful in the training and testing phases than the other algorithms. This is because the EPC algorithm has memory and knowledge of good solutions is maintained by all penguins. So the population always share their knowledge with others. When we talking about ANFIS training based on metaheuristic techniques, it is important to use flexible techniques in global optimal search. Using information sharing among the population makes the EPC algorithm converged at a very high speed in finding a global optimal solution.

The approach of optimizing the Neuro-Fuzzy System based on metaheuristics discussed in this paper was a systematic approach. This approach can be used to design any type of fuzzy system with slight modifications. The approach discussed in this paper was actually methods for training Neuro-Fuzzy System and determining the optimal values of their parameters using intelligent optimization algorithms. In fact, how to train the Neuro-Fuzzy System is expressed as an optimization problem and then solved using intelligent optimization algorithms.

VIII. Conclusion

In this paper, a neuro-fuzzy system based on an Adaptive Neuro-Fuzzy Inference System (ANFIS) was presented. The parameters of this system were optimized using the nature-inspired Emperor Penguins Colony (EPC) algorithm. The purpose was to use the EPC algorithm as an application to optimize the neuro-fuzzy system. The presented neuro-fuzzy system was compared with other neuro-fuzzy systems. Statistical analysis was used to reveal significant differences. The proposed approach was also used to solving the classical inverted pendulum problem. The results showed that the proposed approach had better performance and fewer errors.

REFERENCES

[1] Zheng, Yu-Jun, Haifeng Ling, Shengyong Chen, and Jin-Yun Xue. "A Hybrid Neuro-Fuzzy Network Based on Differential Biogeography-Based Optimization for Online Population Classification in Earthquakes." IEEE Trans. Fuzzy Systems 23, no. 4 (2015): 1070-1083.

[2] Srivastava, Vivek, Bipin K. Tripathi, Vinay K. Pathak, and Anand Tiwari. "On the synergism of evolutionary neuro-fuzzy system." In Neural Networks (IJCNN), 2016 International Joint Conference on, pp. 4827-4834. IEEE, 2016.

[3] Abraham, Ajith. "Neuro fuzzy systems: State-of-the-art modeling techniques." In International Work-Conference on Artificial Neural Networks, pp. 269-276. Springer, Berlin, Heidelberg, 2001.

[4] Taghavifar, Hamid, Asad Modarres Motlagh, Aref Mardani, Ali Hassanpour, Ashkan Haji Hosseinloo, Leyla Taghavifar, and Chongfeng Wei. "Appraisal of Takagi–sugeno type neuro-fuzzy network system with a modified differential evolution method to predict nonlinear wheel dynamics caused by road irregularities." Transport 31, no. 2 (2016): 211-220.

[5] Toosi, Adel Nadjaran, and Mohsen Kahani. "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers." Computer communications 30, no. 10 (2007): 2201-2212.

[6] Obo, Takenori, Chu Kiong Loo, Manjeevan Seera, and Naoyuki Kubota. "Hybrid evolutionary neuro-fuzzy approach based on mutual adaptation for human gesture recognition." Applied Soft Computing 42 (2016): 377-389.

[7] Chen, Yuehui, Bo Yang, Ajith Abraham, and Lizhi Peng. "Automatic design of hierarchical Takagi–Sugeno type fuzzy systems using evolutionary algorithms." IEEE Transactions on Fuzzy Systems 15, no. 3 (2007): 385-397.

[8] Juang, Chia-Feng, and Po-Han Chang. "Designing fuzzy-rule-based systems using continuous ant-colony optimization." IEEE Transactions on Fuzzy Systems 18, no. 1 (2010): 138-149.

[9] Pandiarajan, K., and C. K. Babulal. "Fuzzy harmony search algorithm based optimal power flow for power system security enhancement." International Journal of Electrical Power & Energy Systems 78 (2016): 72-79.

[10] Precup, Radu-Emil, Radu-Codrut David, and Emil M. Petriu. "Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity." IEEE Transactions on Industrial Electronics 64, no. 1 (2017): 527-534.

[11] Shi, Yuhui, Russell Eberhart, and Yaobin Chen. "Implementation of evolutionary fuzzy systems." IEEE Transactions on fuzzy systems 7, no. 2 (1999): 109-119.

[12] Hancer, Emrah, Bing Xue, Mengjie Zhang, Dervis Karaboga, and Bahriye Akay. "A multi-objective artificial bee colony approach to feature selection using fuzzy mutual information." In Evolutionary Computation (CEC), 2015 IEEE Congress on, pp. 2420-2427. IEEE, 2015.

[13] Chatterjee, Amitava, Koliya Pulasinghe, Keigo Watanabe, and Kiyotaka Izumi. "A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems." IEEE Transactions on Industrial Electronics 52, no. 6 (2005): 1478-1489.

[14] Araujo, Ernesto, and Leandro dos S. Coelho. "Particle swarm approaches using Lozi map chaotic sequences to fuzzy modelling of an experimental thermal-vacuum system." Applied Soft Computing 8, no. 4 (2008): 1354-1364.

[15] Sharma, Kaushik Das, Amitava Chatterjee, and Anjan Rakshit. "A hybrid approach for design of stable adaptive fuzzy controllers employing Lyapunov theory and particle swarm optimization." IEEE Transactions on Fuzzy Systems 17, no. 2 (2009): 329-342.

[16] Wai, Rong-Jong, Yu-Chih Huang, and Yi-Chang Chen. "Intelligent daily load forecasting with fuzzy neural network and particle swarm optimization." In Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on, pp. 1-6. IEEE, 2012.

[17] Li, Mengshan, Xingyuan Huang, Hesheng Liu, Bingxiang Liu, Yan Wu, and Xiaozhen Deng. "Solubility prediction of gases in polymers using fuzzy neural network based on particle swarm optimization algorithm and clustering method." Journal of Applied Polymer Science 129, no. 6 (2013): 3297-3303.

[18] Osório, G. J., J. C. O. Matias, and J. P. S. Catalão. "Short-term wind power forecasting using adaptive neuro-fuzzy inference system combined with evolutionary particle swarm optimization, wavelet transform and mutual information." Renewable Energy 75 (2015): 301-307.

[19] Chen, Wei, Mahdi Panahi, and Hamid Reza Pourghasemi. "Performance evaluation of GIS-based new ensemble data mining techniques of adaptive neuro-fuzzy inference system (ANFIS) with genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO) for landslide spatial modelling." Catena 157 (2017): 310-324.

[20] Karaboga, Dervis, and Ebubekir Kaya. "Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey." Artificial Intelligence Review (2018): 1-31.

[21] Harifi, Sasan, Madjid Khalilian, Javad Mohammadzadeh, and Sadoullah Ebrahimnejad. "Emperor Penguins Colony: a New Metaheuristic Algorithm for Optimization." Evolutionary Intelligence (2019).

[22] H. T. Hammel. "Infrared emissivities of some arctic fauna." Journal of Mammalogy 37, no. 3 (1956): 375-378.

[23] Jang, J-SR. "ANFIS: adaptive-network-based fuzzy inference system." IEEE transactions on systems, man, and cybernetics 23, no. 3 (1993): 665-685.

[24] Bezdek, James C., Robert Ehrlich, and William Full. "FCM: The fuzzy c-means clustering algorithm." Computers & Geosciences 10, no. 2-3 (1984): 191-203.

[25] S. N. Sivanandam, and S. N. Deepa. Introduction to genetic algorithms. Springer Science & Business Media, 2007.

[26] James Kennedy. "Particle swarm optimization." In Encyclopedia of machine learning, pp. 760-766. Springer US, 2011.

[27] Dervis Karaboga, and Bahriye Basturk. "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm." Journal of global optimization 39, no. 3 (2007): 459-471.

[28] Rainer Storn, and Kenneth Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces." Journal of global optimization 11, no. 4 (1997): 341-359.

[29] Dorigo, Marco, Mauro Birattari, and Thomas Stutzle. "Ant colony optimization." IEEE computational intelligence magazine 1, no. 4 (2006): 28-39.

[30] [30] Mehrabian, Ali Reza, and Caro Lucas. "A novel numerical optimization algorithm inspired from weed colonization." Ecological informatics 1, no. 4 (2006): 355-366.

[31] Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[32] Derrac, Joaquín, Salvador García, Daniel Molina, and Francisco Herrera. "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms." Swarm and Evolutionary Computation 1, no. 1 (2011): 3-18.

[33] Mendenhall, William, Robert J. Beaver, and Barbara M. Beaver. Introduction to probability and statistics. Cengage Learning, 2012.

Sasan Harifi is currently Ph.D candidate of software systems at the Islamic Azad University of Karaj, Iran. He received his B.Sc degree in computer engineering from the Islamic Azad University of Karaj, Iran in 2011. He received his M.Sc degree in software engineering at Faculty of Electrical, Computer and IT Engineering, Islamic Azad University, Qazvin Branch, Iran in 2015. His research interests include optimization algorithms, metaheuristic algorithms, swarm intelligence, data mining, and big data.

Madjid Khalilian received B.Sc degree in Computer Software Engineering from Islamic Azad University, Iran, in 1997, and M.Sc degree in Computer Software Engineering from Islamic Azad University, Tehran in 2000. Since 2002 he has served as a full-time faculty member of the department of software engineering at Islamic Azad University of Karaj. He has been doing research several projects such as expert systems for classification and identification of plants. In 2008 he joined the faculty of computer science and information technology, department of computer science University Putra Malaysia (UPM) doing his Ph.D. He received his Ph.D degree in 2012. During his study, he investigated the Data Stream Clustering. He also served as a member of the Intrusion Detection System research project. His field of study is intelligent computing and his interests are soft computing, data mining, advanced algorithm, intrusion detection systems, discrete mathematics with application in computer science, semantic web and information retrieval.

Javad Mohammadzadeh is an assistant professor of the software engineering department at Islamic Azad University, Karaj Branch. He received his B.Sc degree in computer science from Shahid Bahonar University of Kerman, Iran in 2004. He received his M.Sc degree in computer science from the University of Tehran, in 2007, and Ph.D degree in bioinformatics from the University of Tehran, in 2014. His research interests include swarm intelligence algorithms, bioinformatics algorithms, complex dynamical networks, and parallel computing.

Sadoullah Ebrahimnejad is an associate professor of operation research in the Islamic Azad University, Karaj Branch. He received his Ph.D in Operation Research and Operation Management from Science and Research Branch of the Islamic Azad University (SRBIAU), M.Sc from Amirkabir University of Technology (Tehran Polytechnic), and B.Sc from Iran University of Science and Technology. His research interests include risk management, construction projects selection, fuzzy MADM, scheduling, mathematical programming, shortest path networks, and supply chain management.