Iranian Journal of Oil & Gas Science and Technology, Vol. 3 (2014), No. 3, pp. 11-25
http://ijogst.put.ac.ir

Identifying Flow Units Using an Artificial Neural Network Approach Optimized by the Imperialist Competitive Algorithm

Seyyed Hossein Hosseini Bidgoli[1], Ghasem Zargar[1*], and Mohammad Ali Riahi[2]

[1] Department of Petroleum Exploration Engineering, Petroleum University of Technology, Abadan, Iran
[2] Institute of Geophysics, Tehran University, Tehran, Iran

Received: July 26, 2013; *revised:* November 30, 2013; *accepted:* December 15, 2013

Abstract

The spatial distribution of petrophysical properties within the reservoirs is one of the most important factors in reservoir characterization. Flow units are the continuous body over a specific reservoir volume within which the geological and petrophysical properties are the same. Accordingly, an accurate prediction of flow units is a major task to achieve a reliable petrophysical description of a reservoir. The aim of this paper was core flow unit determination by using a new intelligent method. Flow units were determined and clustered at specific depths of reservoir by using a combination of artificial neural network (ANN) and a metaheuristic optimization algorithm method. At first, artificial neural network (ANN) was used to determine flow units from well log data. Then, imperialist competitive algorithm (ICA) was employed to obtain the optimal contribution of ANN for a better flow unit prediction and clustering. Available routine core and well log data from a well in one of the Iranian oil fields were used for this determination. The data preprocessing was applied for data normalization and data filtering before these approaches. The results showed that imperialist competitive algorithm (ICA), as a useful optimization method for reservoir characterization, had a better performance in flow zone index (FZI) clustering compared with the conventional K-means clustering method. The results also showed that ICA optimized the artificial neural network (ANN) and improved the disadvantages of gradient-based back propagation algorithm for a better flow unit determination.

Keywords: Hydraulic Flow Units, Imperialist Competitive Algorithm, Artificial Neural Network, Core data, Well logging Data

1. Introduction

Flow units are the continuous body over a specific reservoir volume within which the geological and petrophysical properties are the same (EBANKS, 1987). A complete reservoir description is mostly provided through the identification of flow units. An accurate prediction of flow units is essential for a reliable reservoir petrophysical modeling.

Artificial neural network is a powerful computing method and is based on a nonlinear relationship between inputs and output(s). Neural network can extract the hidden patterns in well log and core data. Neural network has many applications in oil and gas industry such as reservoir characterization, the identification of well test interpretation models, rock and fluid properties forecasting, completion

* Corresponding Author:
Email: zargar@put.ac.ir

analysis, reservoir fractures detection, and formation damage prediction (Mohaghegh, 2000).

In this paper, the multilayer perceptron (MLP) as a well-known feed-forward neural network is used for determining the flow units for a well in one oil field located in the southwest of Iran. The data preprocessing is applied for data normalization and removing outliers.

Many evolutionary algorithms such as genetic algorithm (GA), simulated annealing, particle swarm optimization (PSO) are used for neural network optimization and data clustering (Atashpaz-gargari et al., 2007).

Neural network optimization can be considered for different aspects such as weight training, architecture adaptation (the number of hidden layers, the number of hidden neurons, and node activation functions) and learning rules (Mahmoudi et al., 2009). Moreover, data clustering is an unsupervised learning approach for grouping the data into clusters with the same features.

In this study, imperialist competitive algorithm (ICA) as a new population-based evolutionary method is used for this optimization and clustering.

ICA was presented by Atashpaz-Gargari and Lucas in 2007 and is based on socio-political evolution process (Abdechiri et al., 2010). This algorithm has many applications in solving engineering optimization problems such as data clustering (Niknam et al., 2011; Ebrahimzadeh et al., 2012), Nash equilibrium point achievement (Rajabioun et al., 2008), artificial neural network (ANN) training (Khorani et al., 2011; Zhang, 2012), composite structures (Abdi et al., 2011), production management problems (Nazari-Shirkouhi et al., 2010), and oil industry optimization problems (Ahmadi et al., 2012).

Imperialist competitive algorithm (ICA) is employed for multilayer perceptron (MLP) neural network optimization in flow unit determination at specific depth intervals and for FZI clustering for assigning the best flow unit number to these intervals. This approach is applied to petrophysical modeling and predicting the petrophysical parameters in un-cored wells. The optimal weights and biases for multilayer perceptron (MLP) neural network were obtained and based on FZI clustering; the flow units were distributed across the well.

The efficiency of ICA for MLP neural network optimization and FZI clustering is demonstrated according to the results compared with conventional methods such as gradient descent back propagation algorithm for neural network optimization and K-means approach for data clustering.

2. Hydraulic flow unit definition

Flow units are laterally and vertically continuous across reservoir zones, having the same hydraulic and pore throat petrophysical properties. Petrophysical parameters such capillary pressure, relative permeability curves, and all porous media properties related to porosity/permeability correlation are the same in any hydraulic units (Shahvar et al., 2009). By using this concept, the petrophysical parameters can be investigated at homogeneous intervals and thereby making a better estimation for these properties in un-cored intervals.

Among petrophysical techniques such as Winland R_{35}, Winland-Pittman for carbonates and sandstones, Rock-Fabric, Bryant-Finney, stratigraphic modified Lorenz plot (SMLP), and Carmen-Kozeny and stratigraphic flow profile (SFP) for HFU determination (Gunter et al., 2012) the flow zone index (FZI) approach was considered for this determination. FZI method was presented by Amaefule (1993) and is based on Kozeny-Carmen equation (Amaefule et al., 1993). In this method, the porous medium is supposed to consist of a bundle of capillary tubes.

Different geological parameters and pore scale petrophysical properties are used in Kozeny-Carmen equation. The core permeability and porosity are required for this determination.

At first, the reservoir quality index (*RQI*) and normalized porosity (φ_z)is calculated as follows (Amaefule et al., 1993):

$$RQI = 0.0314\sqrt{\frac{K}{\varphi_e}} \tag{1}$$

where, *K* is permeability in mD.

$$\varphi_z = \frac{\varphi_e}{1-\varphi_e} \tag{2}$$

Then the FZI can be determined by this formula:

$$FZI = \frac{RQI}{\varphi_z} \tag{3}$$

By taking a logarithm of each side of Equation 3:

$$\log(RQI) = \log(\varphi_z) + \log(FZI) \tag{4}$$

By plotting the *RQI* versus φ_z on a logarithmic form, samples by the same FZI scatter around a straight line with a unit slope. Samples of each scatter have the same pore throat properties which belong to a specific flow unit. Therefore, a set of parallel lines can be created equal to the number of flow units determined for depth intervals and the intercepts of this lines at φ_z=1 show the FZI values (Shenawi et al., 2007; Abdi et al., 2007).

3. Artificial neural network

Artificial neural network is a popular computing method for solving problems with a complex hidden structure. This method is applicable to some tasks such as function approximation, time series prediction, pattern recognition, data classification, noise filtering, and control systems. Learning algorithms are used for adjusting the neuron weights and structure parameters and have three main classes, namely supervised learning, reinforcement learning, and unsupervised learning (Kumar, 2012). Some types of neural network are feed-forward, Kohonen self-organizing, and recurrent and radial basis function (RBF) neural networks. In the current study, the most known type of neural network called multilayer perceptron (MLP) was used. MLP is a feed-forward neural network for solving nonlinear problems by using back-propagation algorithm as a gradient-based and supervised training tool. MLP as a multilayer hierarchical structure creates a nonlinear correlation between inputs and output(s). MLP consists of the input, hidden, and output layers. Each layer consists of nodes as processing factors with an activation function and the nodes connect to each other by using right weights (Gentry, 2003).

4. Imperialist competitive algorithm

Imperialist competitive algorithm (ICA) is a new evolutionary computation method based on socio-political competition. This algorithm uses the assimilation policy which the imperialistic countries have reached after the 19th century. In this study, the goal of the optimization is to find the optimal solution for $1\times N_{var}$ variables of neural network weights and biases. Therefore, we have an N_{var}-dimensional optimization problem. At the first step, a country with a $1\times N_{var}$ array is divided into imperialists and colonies based on their cost values. The country and cost function are given as below (Atashpaz-gargari and Lucas, 2007):

$$country = [p_1, p_2, p_3, ..., p_{N_{var}}] \tag{5}$$

$$cost = f(country) = f(p_1, p_2, p_3, ..., p_{N_{var}}) \tag{6}$$

Then, the countries are divided into N_{imp} of imperialists and N_{col} of colonies based on their costs. Therefore, the normalized cost of an imperialist should be determined as reads (Atashpaz-gargari et al., 2007):

$$C_n = c_n - \max_i \{c_i\} \tag{7}$$

c_n and C_n are the cost and normalized cost of nth imperialist respectively. By considering this cost, the normalized power of an imperialist can be defined by (Atashpaz-gargari et al., 2007):

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \tag{8}$$

The initial number of colonies that are possessed by an imperialist (NC_n) is proportional to the power and can be found as given below (Atashpaz-gargari et al., 2007):

$$N.C._n = round\{p_n.N_{col}\} \tag{8}$$

For the second step, the assimilation policy is considered as a movement of colonies toward the imperialists. The distance between the initial position of the colony and an imperialist is defined by d, and the new position of the colony with respect to its initial condition is defined by a random parameter with a uniform distribution (χ) (Figure 1). By considering the assimilation coefficient (β) as a number greater than one, χ can be formulated as (Atashpaz-gargari et al., 2007):

$$x \sim U(0, \beta\times d) \tag{9}$$

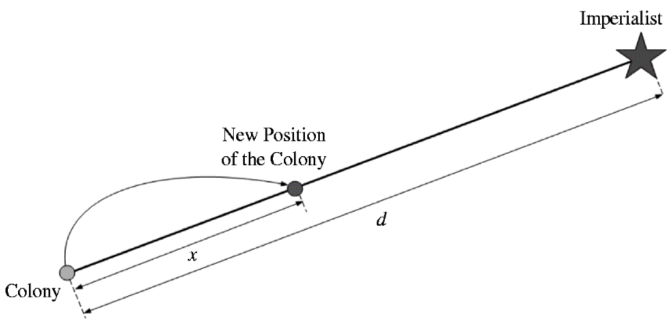


Figure 1
Assimilation process (Atashpaz-gargari et al., 2007).

This movement may be done in all the directions by taking different positions by colonies toward the imperialist. This process is defined by a deviation angle (θ), which is a random number defined as follows (Figure 2) (Atashpaz-gargari et al., 2007) :

$$\theta \sim U(-\gamma, \gamma) \tag{10}$$

γ is the assimilation angle coefficient that adjusts this deviation with respect to the original.

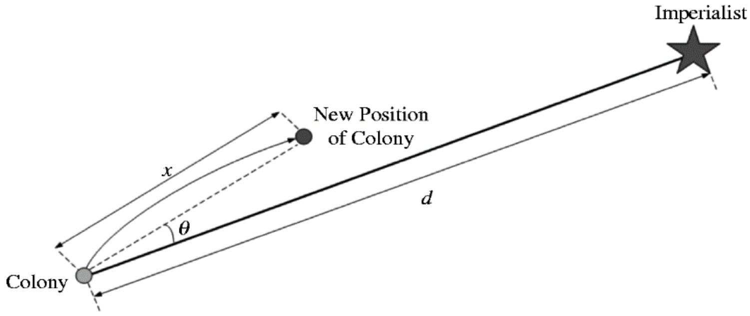


Figure 2
Assimilation process with a deviated movement (Atashpaz-gargari et al., 2007).

For the third step, the algorithm uses a process named revolution. The revolution is a sudden change in the positions of colonies and imperialists, which prevents country from sticking in a local optimum and increases the algorithm exploration. This change in the position is because of a change in the socio-political characteristics of countries and it is defined by a revolution rate in the algorithm.

For the last step, an imperialistic competition takes place between empires. Each empire consists of an individual imperialist with its colonies. At first, the total cost (TC_n) of each empire should be determined as given below (Atashpaz-gargari et al., 2007):

$$T.C._n = Cost(imperialist_n) + \zeta mean\{Cost(colonies\ of\ \ empire_n)\} \tag{11}$$

ζ (zeta) is a positive number less than one.

Therefore, the normalized total cost (NTC_n) and the possession probability (P_{P_n})of each empire are calculated by (Atashpaz-gargari et al., 2007):

$$N.T.C._n = T.C._n - \max_i\{T.C._i\} \tag{12}$$

$$p_{p_n} = \left| \frac{N.T.C._n}{\sum_{i=1}^{N_{imp}} N.T.C._i} \right| \tag{13}$$

Then a vector (P) of these powers is also defined (Atashpaz-gargari and Lucas, 2007):

$$P = [p_{p_1}, p_{p_2}, p_{p_3}, ..., p_{p_{N_{imp}}}] \tag{14}$$

For a random distribution of powers between empires, a set of random numbers (R) is subtracted from vector (P). Thus, the desired randomly distributed powers of empires (D) are generated (Atashpaz-gargari and Lucas, 2007):

$$R = [r_1, r_2, r_3, ..., r_{N_{imp}}] \tag{15}$$

$$r_1, r_2, r_3, ..., r_{N_{imp}} \sim U(0,1) \quad D = P - R = [D_1, D_2, D_3, ..., D_{N_{imp}}] = [p_{p_1} - r_1, p_{p_2} - r_2, p_{p_3} - r_3, ..., p_{p_{N_{imp}}} - r_{N_{imp}}] \tag{16}$$

By considering these randomly distributed powers and the imperialist competition between them, the weaker colonies take apart from the weaker empire and join to an empire by a lower total cost. Therefore, the weakest empire loses all its colonies and its imperialist turns into a colony moving toward the stronger empire. At the end, one empire will remain. The imperialist and colonies of this empire have the same power and there is no difference between them (Figure 3).

ICA as a new evolutionary-based algorithm uses its factors such as assimilation coefficient (β) for exploration and exploitation in the result space or deviation angle (θ) for more diversity in the results. ICA is a proper algorithm for good global convergence performance and it has a good performance in convergence rate and global optimization.

According to this process for imperialist competitive algorithm, the optimal results will be obtained for network training. By defining a proper cost function and by considering the same steps for the algorithm, ICA can be used for data clustering (Khorani et al., 2011).

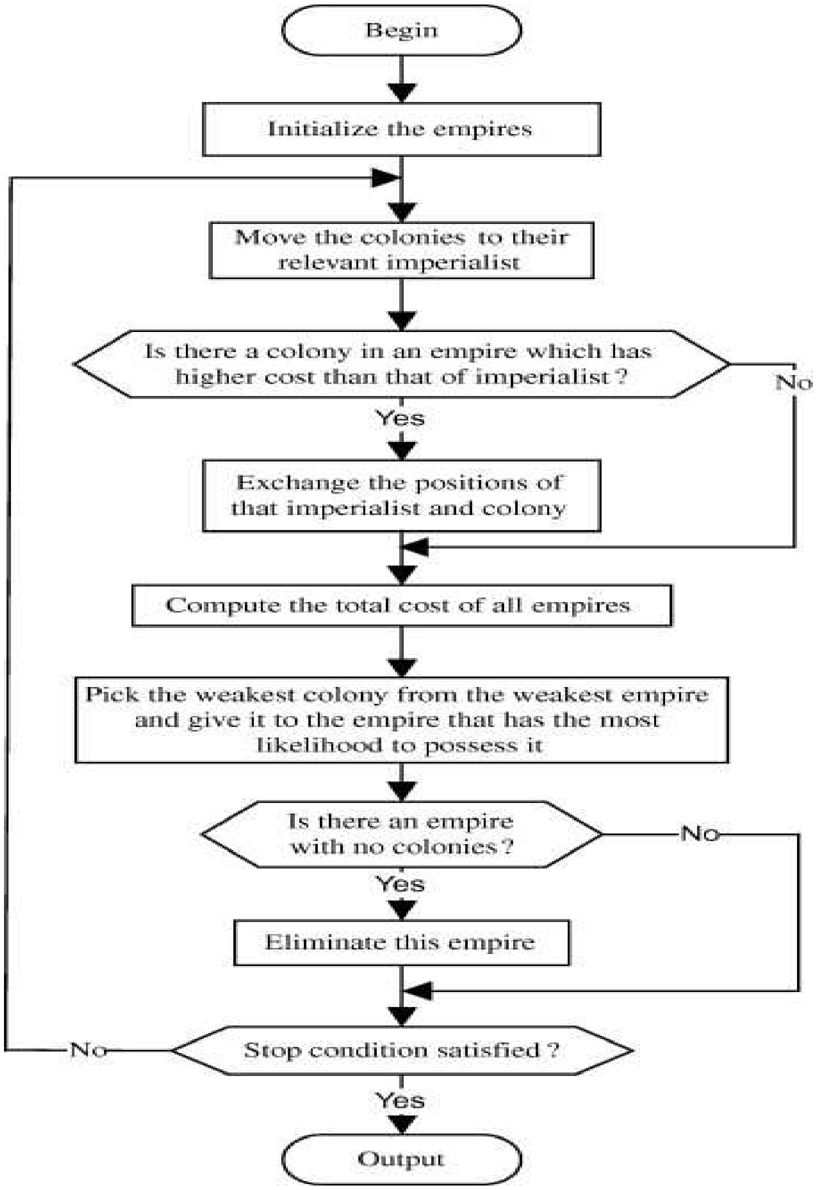


Figure 3
Flowchart of imperialist competitive algorithm (ICA) (Duan et al., 2010).

5. Case study: an oil field of Iran

The oil field of this study is located in the southwest of Iran. It has a trend along the northwest-southwest direction. The structure of this oil field is asymmetrical anticlinal fold. According to WOC, GOC levels, and petrophysical properties, this oil field was divided to four sectors. Well-13 has been chosen for this study. This well is located in the middle sector named sector 3-west. This oilfield consists of Bangestan and Asmari reservoirs. All the depth intervals of Asmari reservoir can be observed in this well and the top of this reservoir is at a depth of 2203 meter along the well. In this reservoir, most of the rocks characterized are limestone and dolomite, and dolomitic limestone is abundant. The porosity of dolomite beds is between 15 to 20%. Fracturing within the Asmari units has limited vertical communication with other units and the horizontal permeability is better than or equal to the vertical permeability.

Wireline logs and the conventional core data were available for flow unit determination. Wireline logs data include log porosity, sonic travel time [DT], deep resistivity [LLD], neutron porosity [NPHI], bulk density [RHOB], standard gamma ray [SGR], and water saturation [Sw].

6. Results and discussion

6.1. Data preprocessing and normalization

Because of reservoir heterogeneity and the existence of errors in data, data preprocessing is an essential procedure before neural network training. In this study, data preprocessing consists of normalization, scaling, removing outliers, and the best dependent inputs selection (Kotsiantis et al., 2006). Many statistical transformations are applied to data normalization such as square root, logarithmic, inverse, and arcsine and Box-Cox transformations.

In the present work, Box-Cox method was used for data normalization. This method has a range of power transformations and high efficiency performance in the normalization of the positively-and negatively-skewed variables (Osborne, 2010).

This transformation is defined as follows:

$$y(\lambda)=\begin{cases}\dfrac{y^{\lambda}-1}{\lambda} & (\lambda\neq 0);\quad y>0\\ Ln(y) & (\lambda=0);\quad y>0\end{cases} \tag{17}$$

In this formula, y, is the amount of data and lambda (λ) is a real number.

The outliers were plotted by Boxplot tool in Matlab Software (Figure 4) and 3.2171% of data were filtered by removing these outliers from the dataset (Figure 5).

Scaling is standardizing the data into the finite range of [-1,1]. In other words, by tuning the network parameters for a given range, the importance of variables can be equalized and therefore continuum weights are created in a predictable range for a better network training (Tang et al., 2011).

The effect of this transformation and data filtering on the correlation between the inputs and output can be investigated by a comparison between the results in Table 1 and Table 2.

By considering the correlation coefficients in Table 2, the best well logs chosen as the inputs are deep resistivity (LLD), sonic travel time (DT), bulk density (RHOB), water saturation (Sw), and neutron porosity (NPHI).

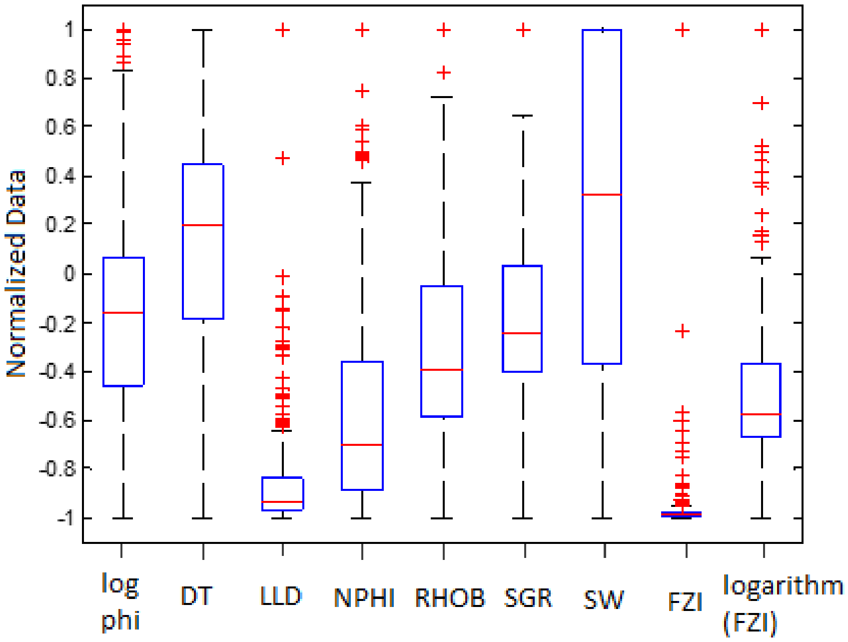


Figure 4
Boxplot of all the well log data, FZI, and logarithm (FZI).

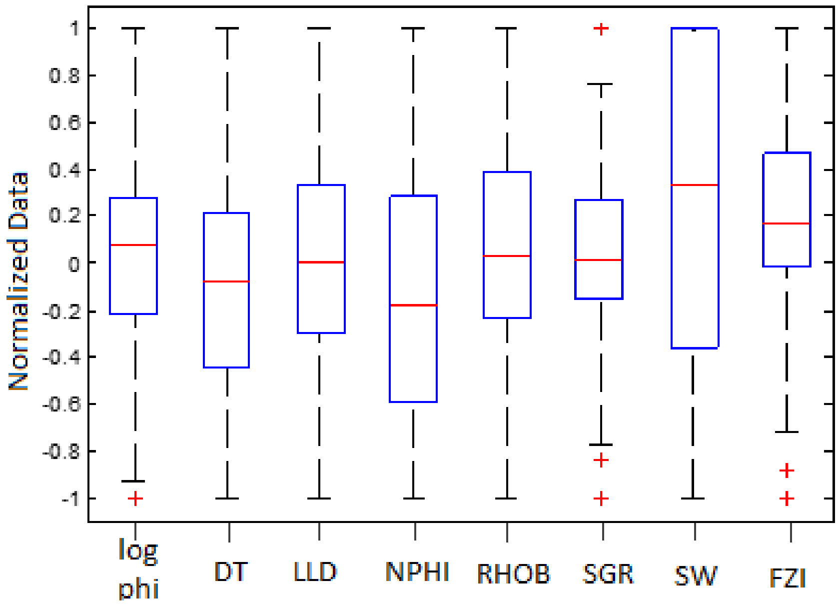


Figure 5
Boxplot of well log data and FZI after Box-Cox transformation and data filtering.

Table 1
Correlation coefficient for all the data (373 rows of data).

	log phi	DT	LLD	NPHI	RHOB	SGR	Sw
FZI	-0.03170	-0.13984	0.12306	0.03544	0.09965	-0.02089	-0.05119
log(FZI)	-0.02588	-0.31383	0.42428	0.22152	0.33117	0.02591	-0.21418

Table 2
Correlation coefficient after Box-Cox transformation and data filtering for normalization
(361 rows of data and 3.2171% of data were filtered).

	log phi	DT	LLD	NPHI	RHOB	SGR	Sw
FZI	-0.06059	-0.35422	0.64304	0.20812	0.34973	0.03593	-0.25867

6.2. FZI clustering using ICA

Clustering is an unsupervised learning approach for partitioning a set of data. The methods of this approach can be considered as hierarchical and partitional methods. K-means is a simple and very popular partitional clustering method that is based on Lloyd algorithm (Lloyd, 1982). However, this algorithm is very dependent on the initial condition of cluster centers and may be stuck at a local optimum. For this reason, evolutionary algorithms can be applied to this approach. In this work, imperialist competitive algorithm (ICA) as a global optimization method was used for well data clustering. This algorithm has good performance in training for better results and in the convergency of clustering approach compared to most evolutionary methods such as genetic algorithm (GA), simulated annealing (SA), and Tabu search (TS) algorithms (Niknam et al., 2011). The sum of minimum distance of points to parallel lines with a unit slope in a logarithmic scale was considered as the cost function for this ICA clustering method.

By using the cumulative distribution function (CDF) histogram for log (FZI) and the number of broken lines assigned to this curve, eight flow units were considered as an input to the algorithm. The ICA method can fit the best line into samples by using an assimilation policy and a competition process, and then the best FZI can be assigned to each cluster (Figure 7). By using the competition feature of ICA, samples located between flow unit lines can be distinguished and clustered in a proper group. The distribution of these flow units is illustrated in Figure 8. The FZI intervals and number of data points related to each flow unit are listed in Table 3.

Table 3
HFU groups determined by ICA.

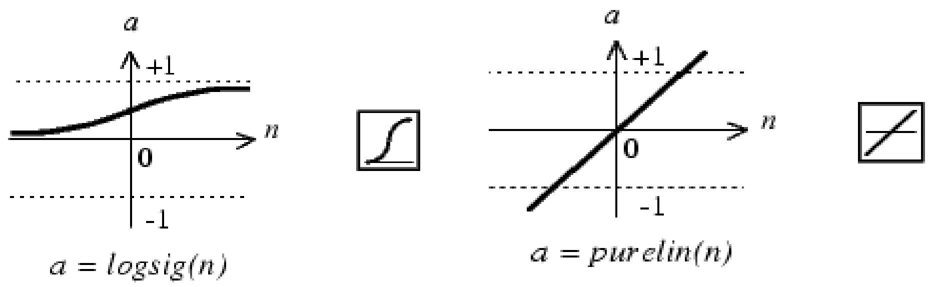
HFU No.	log(FZI) intervals	Number of data	log(FZI)
HFU 1	[-0.6805 , -0.3346]	86	-0.384007585
HFU 2	[-0.3305 , -0.2099]	90	-0.279294165
HFU 3	[-0.2061 , -0.0737]	42	0.137889131
HFU 4	[-0.0647 , 0.0746]	50	0.003756828
HFU 5	[0.0914 , 0.2427]	40	0.157518957
HFU 6	[0.2632 , 0.4223]	23	0.343655364
HFU 7	[0.4401 , 0.8420]	20	0.523070737
HFU 8	[0.9463 , 1.9965]	10	1.228028598

6.2. Multilayer perceptron (MLP) neural network optimization by using ICA (ICA-ANN method)

In the process of ANN optimization by using ICA, the goal of this algorithm is reducing the network training error by assigning the optimal weights to each node connection.

After the data preprocessing stage, the best relevant well logs data (LLD, DT, RHOB, Sw, and NPHI) as the inputs and flow zone index (FZI) as the output were adjusted.

One hidden layer with four neurons was considered for the network structure. Log-sigmoid and linear transfer functions were adjusted for the hidden and output layers (Figure 6).



Log-Sigmoid Transfer Function Linear Transfer Function

Figure 6
Transfer functions for the hidden and output layers.

At first, the network was trained based on a feed-forward neural network with a back propagation gradient descent method using Levenberg-Marquardt (Trainlm) algorithm (Figure 9). The disadvantages of this method are sticking at a local optimum, low speed in convergency, and overfitting. All this disadvantages depend on the initial network weights and biases. For this reason, the neural network was optimized by imperialist competitive algorithm as a global searching method (Ahmadi et al., 2012). This optimization problem has N_{var} unknown parameters equal to the number of weights and biases. Therefore, we optimize the network as an N_{var}-dimension optimization problem. All weights and biases are considered as countries for the ICA. A matrix for the population of countries and a set of training data are used as the inputs to the cost function and the network is evaluated by minimizing the sum of squared errors (SSE) between the real targets (y_r) and the network outputs (y_n):

$$SSE = \frac{1}{N}\sum_{i=1}^{N}(y_r - y_n)^2 = (N-1)\left(S_{y_r}{}^2 - \frac{S_{y_n y_r}{}^2}{S_{y_n}{}^2}\right) \tag{18}$$

In this formula, N is the number of samples and S_{y_r}, S_{y_n}, and $S_{y_n y_r}$ denote the variance of the real targets, the variance of the network outputs, and the covariance between them respectively. Adjusted parameters for this algorithm are listed in Table 4. Using this optimization process, the optimal weights and biases are fitted for a better correlation in the results (Figure 10).

By considering the correlation coefficient (R), mean square error (MSE) parameters (Table 5), and a comparison with another industrial project (Fathollahi, 2007) based on ANN method, the efficiency of ICA for MLP neural network optimization and HFU determination is concluded.

Table 4
Adjusted parameters for ICA.

Number of imperialists	Number of colonies	Number of decades	Revolution rate	Assimilation coefficient	Assimilation angle coefficient	Zeta	Damping ratio	Uniting threshold
7	173	60	0.4	2	0.6	0.03	0.99	0.03

Table 5
Comparison between the results of ANN and ICA optimized ANN methods.

	MSE for training data	Correlation coefficient(*R*) for training data	MSE for testing data	Correlation coefficient(*R*) for testing data
ANN	0.0471	0.77216	0.088	0.6195
ICA optimized ANN	0.0605	0.7204	0.061	0.70063

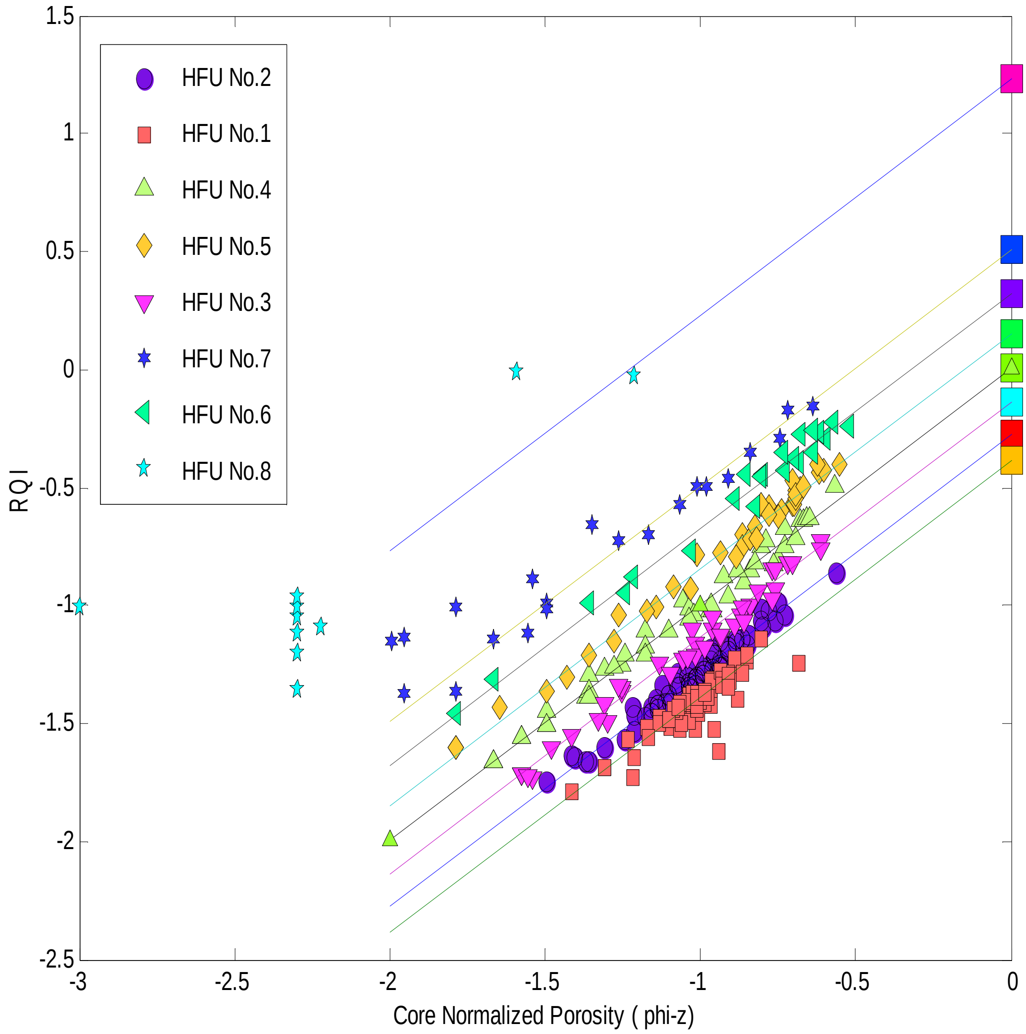


Figure 7
FZI clustering by using ICA.

core depth (m)	HFU No.	core depth (m)	HFU No.	core depth (m)	HFU No.	core depth (m)	HFU No.	core depth (m)	HFU No.	core depth (m)	HFU No.
-2442.67	4	-2464.003	7	-2485.34	7	-2505.15	4	-2535.33	1	-2538.98	2
-2442.97	3	-2464.308	5	-2485.64	3	-2505.46	6	-2535.63	1	-2539.29	1
-2443.28	5	-2465.832	4	-2485.95	3	-2505.76	3	-2535.94	2	-2539.59	2
-2443.58	4	-2466.4416	4	-2486.25	1	-2506.07	6	-2536.24	2	-2539.9	1
-2443.89	7	-2466.7464	5	-2486.86	6	-2506.37	6	-2536.55	1	-2540.2	1
-2444.19	5	-2467.0512	6	-2487.17	6	-2506.68	4	-2536.85	2	-2540.51	3
-2444.5	5	-2467.356	7	-2487.47	5	-2506.98	4	-2537.16	1	-2540.81	2
-2444.8	4	-2467.6608	7	-2487.78	6	-2507.28	7	-2537.46	1	-2541.12	1
-2445.11	6	-2468.2704	5	-2488.08	6	-2507.59	4	-2537.76	2	-2541.42	2
-2445.41	5	-2468.5752	6	-2488.39	6	-2507.89	3	-2538.07	1	-2541.73	1
-2445.72	3	-2468.88	3	-2488.69	5	-2508.2	8	-2538.37	2	-2542.03	2
-2446.02	3	-2469.1848	2	-2489	4	-2508.5	2	-2538.68	1	-2542.34	2
-2446.32	4	-2469.4896	3	-2489.3	6	-2508.81	1	-2538.98	2	-2542.64	2
-2446.63	3	-2469.7944	3	-2489.61	2	-2509.11	1	-2539.29	1	-2542.95	2
-2446.93	2	-2470.0992	3	-2489.91	4	-2509.42	1	-2539.59	2	-2543.25	2
-2447.24	2	-2470.404	7	-2490.22	8	-2516.12	1	-2539.9	1	-2543.56	2
-2447.54	5	-2470.7088	4	-2490.52	1	-2516.43	1	-2540.2	1	-2543.86	1
-2447.85	4	-2471.0136	4	-2491.44	4	-2516.73	1	-2540.51	3	-2544.17	2
-2448.15	4	-2471.3184	3	-2491.74	5	-2517.04	1	-2540.81	2	-2544.47	2
-2448.46	4	-2471.6232	3	-2492.04	4	-2517.34	1	-2541.12	1	-2544.78	2
-2448.76	4	-2471.928	6	-2492.35	8	-2517.65	1	-2541.42	2	-2545.38	1
-2449.07	5	-2472.2328	6	-2492.65	3	-2517.95	1	-2541.73	1	-2545.69	1
-2449.37	6	-2472.5376	5	-2492.96	2	-2518.26	1	-2542.03	2	-2545.99	1
-2449.68	5	-2472.8424	6	-2493.26	2	-2518.56	1	-2542.34	2	-2546.3	1
-2449.98	4	-2473.1472	5	-2493.57	5	-2518.87	3	-2542.64	2	-2546.6	1
-2450.29	1	-2473.452	5	-2493.87	8	-2519.17	3	-2542.95	2	-2546.9	1
-2450.59	8	-2473.7568	3	-2494.18	5	-2519.48	7	-2543.25	2	-2547.21	1
-2450.9	3	-2474.0616	5	-2494.48	7	-2519.78	3	-2543.56	2	-2562.45	3
-2451.2	5	-2474.3664	4	-2494.79	7	-2520.09	7	-2543.86	1	-2562.68	2
-2452.42	3	-2474.6712	7	-2495.09	7	-2520.39	2	-2544.17	2	-2563.06	3
-2452.73	2	-2474.976	5	-2495.4	4	-2520.7	3	-2544.47	2	-2563.37	2
-2455.16	2	-2475.2808	5	-2495.7	4	-2521	1	-2544.78	2	-2563.67	3
-2455.47	2	-2475.5856	8	-2496.01	2	-2521.31	2	-2545.38	1	-2563.98	2
-2455.77	1	-2475.8904	3	-2496.31	1	-2521.61	1	-2545.69	1	-2564.28	1
-2456.08	2	-2476.1952	3	-2496.62	1	-2521.92	1	-2545.99	1	-2564.59	2
-2456.38	2	-2476.5	5	-2496.92	1	-2522.22	5	-2546.3	1	-2564.89	3
-2456.69	1	-2476.8048	5	-2497.23	1	-2522.52	1	-2546.6	1	-2565.2	5
-2456.99	1	-2477.1096	6	-2497.53	2	-2522.83	2	-2546.9	1	-2565.5	2
-2457.3	1	-2477.4144	4	-2497.84	2	-2523.44	2	-2547.21	1	-2565.81	1
-2457.6	4	-2478.024	6	-2498.14	3	-2523.74	1	-2532.58	2	-2566.42	1
-2457.91	3	-2478.3288	6	-2498.45	4	-2524.05	1	-2532.89	1	-2566.72	2
-2458.21	3	-2478.6336	4	-2498.75	5	-2524.35	1	-2533.19	2	-2567.03	2
-2458.52	5	-2478.9384	2	-2499.06	2	-2524.66	6	-2533.5	2	-2567.33	2
-2458.82	5	-2479.2432	3	-2499.36	1	-2530.14	1	-2533.8	2	-2567.64	2
-2459.13	4	-2479.548	3	-2499.66	3	-2530.45	1	-2534.11	2	-2567.94	4
-2459.43	1	-2479.8528	4	-2500.27	1	-2530.75	1	-2534.41	2	-2568.24	4
-2459.74	1	-2480.1576	2	-2500.58	4	-2531.06	1	-2534.72	2	-2568.55	4
-2460.04	4	-2480.4624	2	-2501.19	2	-2531.36	2	-2535.02	1	-2568.85	2
-2460.35	7	-2481.3768	4	-2501.49	2	-2531.67	2	-2535.33	1	-2569.16	4
-2460.65	6	-2481.6816	5	-2501.8	5	-2531.97	2	-2535.63	1	-2569.46	3
-2460.96	4	-2481.9864	4	-2502.1	5	-2532.28	1	-2535.94	2	-2569.77	4
-2461.26	7	-2482.2912	5	-2502.41	4	-2532.58	2	-2536.24	2	-2570.07	4
-2461.56	8	-2482.9008	5	-2502.71	6	-2532.89	1	-2536.55	1	-2570.38	4
-2461.87	4	-2483.2056	2	-2503.02	8	-2533.19	2	-2536.85	2	-2570.68	4
-2462.17	5	-2483.5104	4	-2503.32	7	-2533.5	2	-2537.16	1	-2570.99	3
-2462.48	4	-2483.8152	7	-2503.63	8	-2533.8	2	-2537.46	1	-2571.3	3
-2462.78	3	-2484.12	4	-2503.93	7	-2534.11	2	-2537.76	2	-2571.6	4
-2463.09	8	-2484.4248	2	-2504.24	7	-2534.41	2	-2538.07	1	-2571.9	5
-2463.39	6	-2484.7296	3	-2504.54	5	-2534.72	2	-2538.37	2	-2572.21	4
-2463.7	7	-2485.0344	5	-2504.85	5	-2535.02	1	-2538.68	1	-2572.51	3
										-2572.82	5

Figure 8
HFU's distribution across core depth intervals.

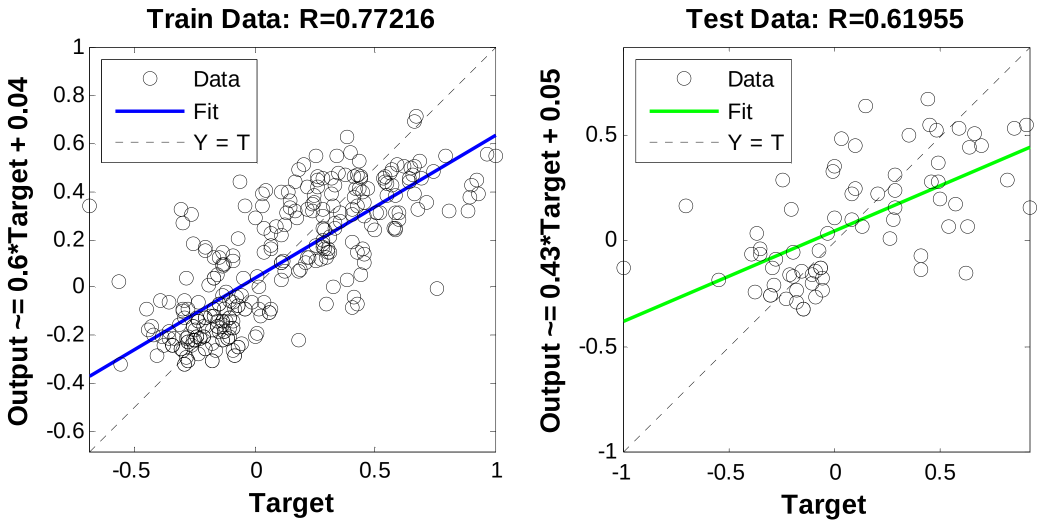


Figure 9
Correlation coefficients for ANN.

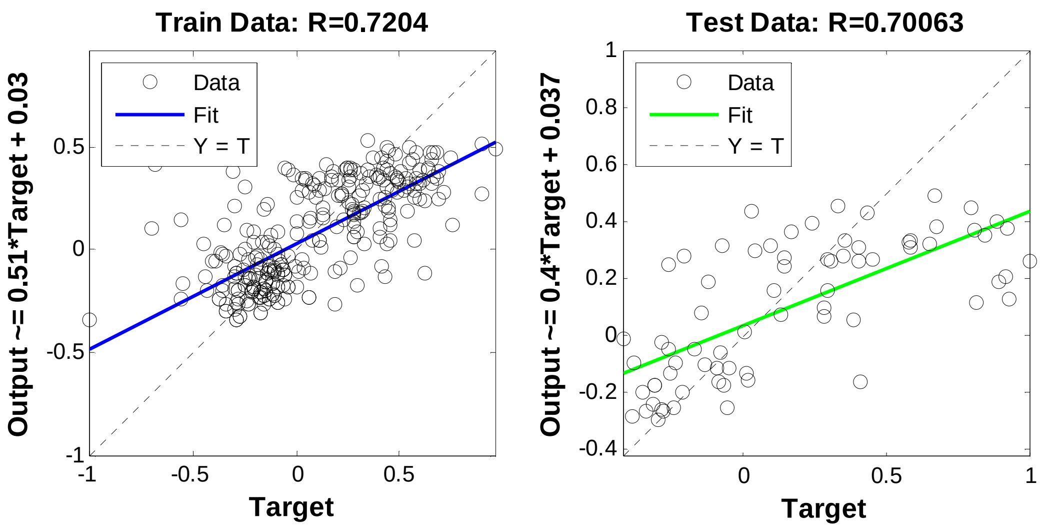


Figure 10
Correlation coefficients for ICA optimized ANN.

7. Conclusions

1. In this paper, imperialist competitive algorithm (ICA) was applied to multilayer perceptron (MLP) neural network and flow units clustering;
2. Data preprocessing of real well log and core data improved the correlations between the inputs and output by data normalization and filtering;
3. ICA as a global searching evolutionary method had better performance in flow zone index (FZI) clustering compared with conventional K-means method, because of the dependency of K-means method on the initial condition of clustering and sticking at a local optimum. The results showed that ICA determined the optimal weights and biases for MLP and improved the disadvantages of the gradient-based back propagation algorithm such as over fitting and low convergence speed;

4. For future work, the architecture adaption and learning rules of neural network can be optimized by improved ICA methods such as evolving ICA and new optimization algorithms or ICA combined with chaos theory.

Nomenclature

DT	: Sonic travel time
FZI	: Flow zone index
GA	: Genetic algorithm
ICA	: Imperialist competitive algorithm
LLD	: Deep resistivity
MLP	: Multilayer perceptron
NPHI	: Neutron porosity
PSO	: Particle swarm optimization
RHOB	: Bulk density
RQI	: Reservoir quality index
SA	: Simulated annealing
TS	: Tabu search

References

Abdechiri, M., Faez, K., and Bahrami, H., Adaptive Imperialist Competitive Algorithm (AICA), Cognitive Informatics (ICCI), 9th IEEE International Conference, 2010.

Abdi, B., Mozafari, H., Ayob, A., and Kohandel, R., Imperialist Competitive Algorithm and Its Application in Optimization of Laminated Composite Structures, European Journal of Scientific Research Vol. 55, No. 2, p. 174-187, 2011.

Abdi, Y., Ghane, M., and Haghighi, A., Integrated Reservoir Characterization and Modeling of One Iranian Naturally Fractured Reservoir Using Laboratory and Field Data, SPE/EAGE Reservoir Characterization and Simulation Conference, 2007.

Ahmadi, M. A., Ahmadi M. R., and Shadizadeh, S. R., Evolving Artificial Neural Network and Imperialist Competitive Algorithm for Prediction Permeability of the Reservoir, Neural Computing and Applications, p. 1-9, 2012.

Amaefule, J. O., Altunbay, M., Tiab, D., Kersey, D. J. and Keelan, D. K., Enhanced Reservoir Description: Using Core and Log Data to Identify Hydraulic (Flow) Units and Predict Permeability in Uncored Intervals/wells, SPE Annual Technical Conference and Exhibition, 1993.

Atashpaz-Gargari, E. and Lucas, C., Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition, In Evolutionary Computation, CEC, IEEE Congress on, p. 4661-4667, IEEE, 2007.

Duan, H., Xu, C., Liu, S., and Shao, S., Template Matching using Chaotic Imperialist Competitive Algorithm, Pattern Recognition Letters 31, No. 13, p. 1868-1875, 2010.

Ebrahimzadeh, A., Addeh, J., and Rahmani, Z., Control Chart Pattern Recognition Using K-MICA Clustering and Neural Networks, ISA Transactions, Vol. 51, No. 1, p. 20111-119, 2012.

Fathollahi, S., Permeability Prediction by Using Artificial Neural Networks and Hydraulic Flow Units in a Reservoir in Southwest of Iran, M.S thesis, Petroleum University of Technology, Iran, June 2007.

Gentry, M. D., Applications of Artificial Neural Networks in the Identification of Flow Units, Happy Spraberry Field, Garza County, M.S. thesis, Texas A&M University, 2003.

Khorani, V., Forouzideh, N., and Motie Nasrabadi, A., Artificial Neural Network Weights Optimization Using ICA, GA, ICA-GA and R-ICA-GA: Comparing performances, Hybrid Intelligent Models and Applications (HIMA), IEEE Workshop on IEEE, 2011.

Kotsiantis, S. B., Kanellopoulos, D., and Pintelas, P. E., Data preprocessing for Supervised Leaning, International Journal of Computer Science, Vol. 1, No. 2, p.111-117, 2006.

Kumar, A., Artificial Neural Network as a Tool for Reservoir Characterization and its Application in the Petroleum Engineering, Offshore Technology Conference, 2012.

Lloyd, S., Least Squares Quantization in PCM, Information Theory, IEEE Transactions, Vol. 28, No. 2, p. 129-137, 1982.

Mahmoudi, Tayefeh, M., Forouzideh, N., Lucas, C., and Taghiyareh, F., Artificial Neural Network Weights Optimization Based on Imperialist Competitive Algorithm, In 7th International Conference on Computer Science and Information Technologies (CSIT'09), Yerevan, p. 244-247, 2009.

Mohaghegh, S., Virtual-intelligence Applications in Petroleum Engineering: Part I-Artificial Neural Networks. Journal of Petroleum Technology. Vol. 52, No. 9, p. 64-73, 2000.

Nazari-Shirkouhi, S., Eivazy, H., Ghodsi, R., Rezaie, K., and Atashpaz-Gargari, E., Solving the Integrated Product Mix-outsourcing Problem Using the Imperialist Competitive Algorithm, Expert Systems with Applications, Vol. 37, No. 12. p. 7615-7626. 2010.

Niknam, T., Taherian Fard, E., Ehrampoosh, S., and Rousta. A., A New Hybrid Imperialist Competitive Algorithm on Data Clustering, Sadhana, Vol. 36, No. 3, p.293-315, 2011.

Niknam, T., Taherian Fard, E., Pourjafarian, N., and Rousta, A., An Efficient Hybrid Algorithm Based on Modified Imperialist Competitive Algorithm and K-means for Data Clustering, Engineering Applications of Artificial Intelligence, Vol. 24, No. 2, p. 306-317, 2011.

Osborne, J. W., Improving your Data Transformations: Applying the Box-Cox Transformation, Practical Assessment, Research & Evaluation, Vol. 15, No. 12, p. 1-9, 2010.

Rajabioun, R., Atashpaz-Gargari, E., and Lucas, C., Colonial Competitive Algorithm as a Tool for Nash Equilibrium Point Achievement, Computational Science and its Applications, ICCSA, Springer Berlin Heidelberg, p. 680-695, 2008.

Shahvar, M., Kharrat, R., and Mahdavi, R., Incorporating Fuzzy Logic and Artificial Neural Networks for Building a Hydraulic Unit-based Model for Permeability Prediction of a Heterogeneous Carbonate Reservoir, International Petroleum Technology Conference, 2009.

Shenawi, S. H., White, J. P., Ahmed Elrafie, E., and El-Kilany, K. A., Permeability and Water Saturation Distribution by Lithologic Facies and Hydraulic Units: A Reservoir Simulation Case Study, In SPE Middle East Oil and Gas Show and Conference, Society of Petroleum Engineers, 2007.

Tang, H., Meddaugh, W., and Toomey, N., Using an Artificial Neural Network Method to Predict Carbonate Well Log Facies Successfully, SPE Reservoir Evaluation & Engineering, Vol.14, No.1, p. 35-44, 2011.

Zhang, X., The Application of Imperialist Competitive Algorithm Based on Chaos Theory in Perceptron Neural Network, Physics Procedia, Vol. 25, p. 536-542, 2012.