Research Article

Parameter Identification of Multistage Fracturing Horizontal Well Based on PSO-RBF Neural Network

Rongwang Yin,1,2 Qingyu Li,1 Peichao Li,3 and Detang Lu1

1School of Engineering Science, University of Science and Technology of China, Hefei 230026, China
2Department of Basic Teaching and Experiment, Hefei University, Hefei 230601, China
3School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

Correspondence should be addressed to Qingyu Li; liqy@ustc.edu.cn and Detang Lu; dtlu@ustc.edu.cn

Received 30 April 2019; Revised 14 March 2020; Accepted 18 June 2020; Published 3 July 2020

Academic Editor: Cristian Mateos

Copyright © 2020 Rongwang Yin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to more accurately identify multistage fracturing horizontal well (MFHW) parameters and address the heterogeneity of reservoirs and the randomness of well-production data, a new method based on the PSO-RBF neural network model is proposed. First, the GPU parallel program is used to calculate the bottomhole pressure of a multistage fracturing horizontal well. Second, most of the above pressure data are imported into the RBF neural network model for training. In the training process, the optimization function of the global optimal solution of the PSO algorithm is employed to optimize the parameters of the RBF neural network, and eventually, the required PSO-RBF neural network model is established. Third, the resulting neural network is tested using the remaining data. Finally, a field case of a multistage fracturing horizontal well is studied by using the presented PSO-RBF neural network model. The results show that in most cases, the proposed model performs better than other models, with the highest correlation coefficient, the lowest mean, and absolute error. This proves that the PSO-RBF neural network model can be applied effectively to horizontal well parameter identification. The proposed model has great potential to improve the prediction accuracy of reservoir physical parameters.

1. Introduction

Reservoir description is an important means of estimating the physical properties of a reservoir in the petroleum industry. These reservoir physical parameters require better methods to improve the prediction accuracy to further enhance the subsequent success rate of exploration and development. Multiphase flow in oil and gas fields usually refers to the simultaneous flow of multiple fluids in a reservoir, which generally occurs during oil and gas production [1, 2]. For the old oilfields, oil and gas development has experienced stages such as self-spraying, secondary, and tertiary oil recovery, which makes the process of the oil and gas production complex and challenging. In order to deal with this complexity, a lot of studies have been done at home and abroad, and the reservoir numerical simulations have been introduced to estimate the multiphase flow characteristics of oil, natural gas, and water [3, 4].

In the petroleum industry, well-logging data have been commonly used to predict formation parameters such as permeability, formation boundaries, and fracture length. Permeability is defined as a measure of the ability of a porous medium to allow fluid to pass through it. The concept of permeability is important for determining accurate reservoir description, simulation, and management. Therefore, prior to any modeling or calculation, the permeability of porous media must be determined. The earliest method of permeability prediction is the empirical correlation between permeability and other petrophysical properties such as porosity and water saturation. These correlations have made some successes in sandstone reservoirs. However, it is not applicable to heterogeneous formations.

In recent years, the artificial neural network (ANN) has been applied to the reservoir field, solving many highly complex nonlinear problems [5]. The ANN is considered to be a nonlinear tool that can predict complex nonlinear
In the past few decades, researchers have done a lot of work to study reservoir parameter estimation problems, such as permeability, porosity, and fracture length. Taking the reservoir behavior well as an example, there are currently three main methods: core analysis, well test analysis, and artificial intelligence methods.

The most accurate method of permeability prediction is to perform core analysis in the laboratory by applying Darcy’s law. Among the different techniques, the permeability obtained from the core analysis is more efficient than that obtained by other methods, and the permeability from the core analysis can be used to verify other estimation models. However, the core analysis process is time-consuming and difficult to use widely.

Another method of obtaining reservoir permeability is well test analysis. The analytical solution is solved by using the porous flow equation, and some rock properties such as permeability and porosity can be then obtained inversely from the pressure curve. The data available after well testing help petroleum engineers calculate formation permeability, skin factor, and wellbore storage [9]. Due to the cost of testing, well testing is not a good solution for any reservoirs that have been developed.

ANNs are increasingly being employed to predict reservoir properties based on geophysical well-logging data [10, 11]. Mohaghegh et al. pointed out that neural networks are powerful tools for identifying the relationship between permeability and well-logging data [12]. In addition, Aminian et al.’s research shows that artificial neural networks can be used to predict formation permeability, even in highly heterogeneous reservoirs [13, 14]. Based on the artificial neural network and particle swarm optimization algorithm, Ahmadi et al. proposed a method for predicting horizontal well productivity in a pseudosteady state, and the experimental results achieved a high fitting accuracy (<0.82%) [15].

Saemi et al. improved the predictive performance of neural networks by using the GA to optimize the network parameters of the ANN [19]. In addition, Kaydani et al.’s findings indicate that the GA and ANN network structures designed by subregion can predict the permeability of a heterogeneous reservoir in Iran [20]. Tahmasebi et al. proposed four methods for predicting permeability of different neural network structures and statistically compared the results obtained. Finally, a new method of the modular neural network (MNN) was obtained [21]. By combining cuckoo optimization algorithm (COA), particle swarm optimization algorithm (PSO), imperialist competitive algorithm (ICA), and Levenberg–Marquardt algorithm (LM), Kaydan et al. proposed a new method to estimate the permeability [22].

Rough set theory can be successfully applied to the prediction of permeability of porous media [23–28]. By establishing a rough set model, Ilkchki et al. successfully predicted the permeability of an offshore gas field in Iran [29].

Support vector regression (SVR) based on the principle of structural risk minimization is a promising machine learning method. Through experimental comparisons, Gholami et al. found that the SVR method has faster speed and higher accuracy in predicting reservoir permeability [30].

3. Problem Formulation

3.1. Optimized Design. After fracturing tight sandstone reservoirs, the physical properties of the reservoirs are characterized by extremely low porosity, a wide range of permeability changes, and complex pore-permeability relationships. Therefore, conventional production prediction models often fail to meet the requirements in terms of prediction accuracy.

Effective algorithms for complex nonlinear problems are essentially optimization problems. The so-called optimization refers to the problem of seeking the maximum or minimum value of the given objective function according to the change of design variables under the condition of...
meeting some certain constraints. It can be described by the following equation:

\[
\min f(x) \text{ or } \max f(x),
\]

s.t. \( g_i(x) = 0, i = 1, 2, \ldots, h_j(x) \geq 0 \text{ or } h_j(x) \leq 0, \quad (1) \)

where \( x \in S \) and \( S \) is the design parameter space that meets the limiting conditions, called the solution space. \( f(x) = f(x_1, x_2, \ldots, x_n) \) is the objective function to be optimized. \( x = (x_1, x_2, \ldots, x_n)^T, \ x_1, x_2, \ldots, x_n \) is the optimization design variable, and \( \min f(x) \) or \( \max f(x) \) is the minimum or maximum value of the objective function. \( g_i(x) = 0, i = 1, 2, \ldots, \) are the equality constraints, and \( h_j(x) \geq 0 \text{ or } h_j(x) \leq 0 \) denotes the inequality constraints.

Taking MFHW as the studied object, the PSO-RBF neural network was used to predict permeability and fracture half-length. The optimization objective function is the minimum RMSE (root mean square error) value of permeability and half-length of fracture. The input variables of the network are the formation pressure, the number of fractures, stratum thickness, the well storage constant, the well storage skin, and so on, and the output variables are the RMSE values.

3.2. Data Set. The data set describes the characteristics and behavior of the reservoir’s input and output parameters and is the training data set for the reservoir model. It is obtained using static and dynamic data such as porosity, permeability, pressure, and reservoir production values. For a given source unit, according to the line source superposition principle, the pressure distribution at any point \( (x_D, y_D, z_D) \) in the formation space shall be

\[
P_{D_{ij}}(x_D, y_D, z_D, t_D) = \int_0^t q_{Dij}(t)G_{x_D}(x_D, \tau_D)G_{y_D}(y_D, \tau_D)G_{z_D}(z_D, \tau_D)d\tau.
\]

(2)

The coefficient matrix is expressed as follows:

\[
\begin{pmatrix}
M_{11} & M_{12} & \cdots & M_{1NM_i} & -1 \\
M_{21} & M_{22} & \cdots & M_{2NM_i} & -1 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
M_{NM_i,1} & M_{NM_i,2} & \cdots & M_{NM_i,NM_j} & 1 \\
1 & 1 & \cdots & 1 & 0
\end{pmatrix}
\begin{pmatrix}
q_{D11} \\
q_{D12} \\
\vdots \\
q_{DNM_i} \\
p_{D_{ij}}
\end{pmatrix}
= \begin{pmatrix}
0 \\
0 \\
\vdots \\
0 \\
1
\end{pmatrix}
\]

(3)

As for the definitions of various symbols appearing in equations (2) and (3), the reader can refer to our previous paper [31]. Note that, the formation pressure data of MFHW are obtained by the GPU parallel calculation method proposed in the reference [31]. All training data sets are divided into two parts before the PSO-RBF training begins. One is the input data set, and the other is the output data set. The input and output data sets are normalized within a specific range. Before the training begins, a standard normalization function is used to limit the range of the input and output data sets between \(-1\) and \(1\), and the mathematical function is given in the following equation:

\[
y = \frac{e^{2x} - 1}{e^{2x} + 1}
\]

(4)

During PSO-RBF training, the training data set is divided into three parts: training, validation, and testing. Training data are used in the training, and validation data are also used during training. However, it should be mentioned that validation data are used to check the network learning instead of training the network.

4. PSO-RBF Model

The advantage of ANNs over other conventional techniques is the ability to perform complex and highly nonlinear tasks accurately and swiftly. In most previous works related to reservoir models, the researchers used the BPNN to construct reservoir models. However, the BPNN has a problem of falling into local minimum during training time. In most cases, the network does not reach the global minimum to find the minimum error value. The global optimization ability of the PSO algorithm makes the radial basis neural network model based on the PSO algorithm have no local minimum problem.

4.1. PSO Algorithm. Inspired by the movement of the birds, the PSO algorithm was developed by Kennedy [32]. In this approach, every possible solution is considered as a particle. Each particle is characterized by its position and velocity. The position of the particle is defined in the hyperspace, and its dimension is equal to the number of nonoptimized parameters. Several particles are initially defined in hyperspace, which iteratively change their positions to determine the best position. The fitness of a particle is determined by a fitness function such as RMSE. This algorithm is similar to how birds search for food in a wide area.

During the iterative execution of the algorithm, the values of \( p_{best} \) and \( g_{best} \) are constantly updated. \( p_{best} \) is defined as the optimal position of the particle in the hyperspace, determined by the fitness value. \( g_{best} \) is the overall best position for all particles. In each iteration step, the speed is updated first and then the location is updated. The particles are accelerated to \( p_{best} \) and \( g_{best} \) by updating the speed:

\[
v_{k+1} = w_k \cdot v_k + c_1 \cdot r_1 \cdot (p_{kbest} - p_k) + c_2 \cdot r_2 \cdot (g_{best} - p_k),
\]

(5)

where \( v_{k+1} \) is the speed of the next iteration, \( w_k \) is the inertia weight, \( v_k \) is the current speed, \( r_1 \) and \( r_2 \) are random numbers, and \( p_k \) is the current position of the particle. Update the position of the particle using the following equation:

\[
p_{k+1} = p_k + v_{k+1} \cdot \Delta t,
\]

(6)

The initial position and velocity of each particle are randomly distributed. After initializing the position and velocity of all particles, the fitness is calculated. In the subsequent steps, the position and velocity are iteratively
updated by the local best parameters and the global best parameters, as shown in Figure 1.

The entire flowchart can be divided into four parts, namely initialization, fitness calculation, status check, and updating of speed and position.

4.2. Radial Basis Function Neural Network. RBFNN has a small number of hidden layer neurons and are used to develop networks with good generalization capabilities [33]. RBFNN is considered to be a special type of artificial neural network because its architecture requires only a hidden layer, which allows the input space to be represented in new spaces of different hidden layer neurons. The structure is shown in Figure 2. In the training process of RBFNN, it appears as a linear model because all hidden neuron centers and calculations are fixed. The RBFNN hidden layer neurons perform a nonlinear transformation and map all inputs into the new input space. The output layer is considered as a linear converter and is applied to the new input space. The performance of RBFNN can be determined by adjusting the center (width) of the hidden layer neurons, and there is no specific formula to choose the width of the radial basis function [34]. RBFNN has been widely used in system prediction, pattern recognition, speech recognition, and adaptive control. It has also been used to solve the problems of oil and gas fields, e.g., oil and gas ratios of reservoirs, electromagnetics, resistivity, log data inversion, log data prediction, seismic properties, and nonlinear relationships between reservoir properties and seismic properties [35–37].

The growth and overall structure of RBFNN is affected by the RBF. The RBFNN input is directly connected to each of the basic functions and produces an output. That is,

$$
\phi_i = \exp \left( -\frac{\|x - u\|^2}{\sigma^2} \right),
$$

where $x$ represents the input data of the network, $u$ is the center of the radial basis function, and $\sigma$ is the width of the RBF ($\sigma > 0$). First, the hidden layer neurons are calculated based on the radius of the radial basis function and then transmitted to the output layer. Here, the sum of the
products between the hidden layer neurons and the weight vectors is calculated to obtain the final network output \( y_n \); that is,

\[
y_n = \sum_{m=1}^{m} w_i \cdot \phi_i.
\]

(8)

4.3. PSO-RBF Neural Network Method. The prediction accuracy of the RBFNN mainly depends on the central vector \( u \) of the radial basis kernel function, the radius \( \sigma \) of the radial basis, and the connection weight \( w \) between the output layer and the hidden layer. The traditional RBF neural network uses local information based on parameter space to set parameters, which results in the values of \( u, \sigma \), and \( w \) being local optimal solutions rather than global optimal solutions. In view of these defects in the RBFNN, this work employs the PSO algorithm to optimize the traditional RBFNN when identifying MFHW parameters. The RBFNN parameters optimized by the PSO algorithm are global optimal parameters, thus avoiding the problem of low reliability of RBFNN learning. The specific flowchart of the PSO-RBF algorithm is shown in Figure 3.

PSO optimization RBFNN parameters are divided into two steps: the first step is to determine the center value and width of the basis function, and the second step is to determine the connection weight between the hidden layer and the input layer. In the optimization process, the data obtained by GPU parallel computing are used to train and verify the network.

The algorithm flow of the first step is as follows.

(1) Collection of reservoir model data samples; (2) cluster analysis of samples to generate center \( u \) and width \( \sigma \) of basis functions; (3) initialization of the particle swarm algorithm using \( u \) and \( \sigma \); (4) calculation of each particle root mean square error (RMSE); (5) updating local optimal solution and global optimal solution; (6) updating particle position and velocity; (7) repeating steps (4–6) until the accuracy requirement or number of iterations is reached; and (8) obtaining the center \( u \) and width \( \sigma \) of the basis function.

The algorithm flow of the second step is as follows.

(1) Calculation of the output of the hidden layer; (2) initialization of the weight \( w \) and reinitialization of the particle group; (3) calculating the cumulative error of each particle; (4) updating the local optimal solution and the global optimal solution; (5) adjusting the position and velocity of the particles; (6) repeating steps (3–5); and (7) obtaining the parameter \( w \) of the RBF neural network.

5. A Field Case Study

5.1. Computation. The bottomhole flow pressure is analyzed and calculated by using the measured data of multistage fracturing horizontal wells in an oilfield in western China. The three-dimensional PEBI grid of fracturing of underground horizontal wells is shown in Figure 4. The reservoir stratum is homogeneous, the fluid flow satisfies Darcy’s law, the horizontal and vertical permeability of the fracture are different, and the horizontal well is fractured in a closed rectangular reservoir. Using the GPU-based MFHW bottomhole pressure calculation method proposed in our previous work [31], the bottomhole flow pressure of the horizontal well is calculated, and the data set is obtained.

5.2. Training Model. The various parameter settings in the particle swarm algorithm are described in detail in the sensitivity study section. The RBFNN consists of 5 input nodes and 1 output node. The PSO optimization steps presented in the previous section are used to optimize the training of hidden layer neurons 4, 8, 12, 16, 20, 24, and 28, respectively. The test results are shown in Figure 5. As can be seen from the figure, when the number of hidden layer neurons is 16, the error is the lowest. Therefore, an RBN network structure of 5-16-1 is obtained.
5.3. Prediction. After obtaining the RBF neural network parameters through the PSO algorithm, the network needs to be tested. To this end, the remaining data are imported into the optimized RBF neural network to obtain the predicted values of the bottomhole permeability and the fracture length of MFHW.

In order to further detect the performance of the network, the PSO-RBF network structure proposed in the paper is compared with the PSO-BPNN and the SVR algorithm. The result is shown in Figure 6. The permeability prediction results and errors of the three algorithms are shown in Table 1. As can be obtained from Table 1, the RMSE of the PSO-BPNN is 0.332, the RMSE of the SVR algorithm is 0.308, and the RMSE of the PSO-RBF is 0.178. The fracture length is predicted by three algorithms, and the result is shown in Figure 7. The fracture length prediction results and errors of the three algorithms are shown in Table 2. As can be obtained from Table 2, the RMSE of the PSO-BPNN is 3.776, the RMSE of the SVR algorithm is 3.319, and the RMSE of the PSO-RBF is 2.250. It can be seen from the comparison results that the RMSE of the PSO-RBF algorithm is much lower than that of the other two algorithms, showing that the algorithm has higher prediction accuracy and better performance.

6. Sensitivity Study

In the particle swarm optimization algorithm, the population number, particle speed, inertia weight, learning factor, and random number all have certain effects on the performance of the algorithm. According to the studies at home and abroad, the improvement of the PSO algorithm mainly focuses on the inertia weights and learning factors [38, 39]. The inertia weight plays a major role in the convergence of the particle swarm algorithm. The larger the value of \( \omega \), the stronger the global search ability and, conversely, the stronger the local search capability. \( c_1 \) and \( c_2 \) are the maximum step sizes of the particles to adjust to the individual optimal or group optimal direction,
Table 1: Permeability prediction results of different algorithms.

| Fracture number | Measured value | PSO-BPNN algorithm | SVR algorithm | PSO-RBN algorithm |
|-----------------|----------------|--------------------|---------------|-------------------|
| 1               | 1.549          | 1.463              | 1.294         | 1.663             |
| 2               | 0.011          | 0.085              | 0.094         | 0.056             |
| 3               | 2.837          | 2.668              | 2.762         | 2.793             |
| 4               | 4.671          | 5.172              | 4.854         | 4.701             |
| 5               | 1.437          | 1.660              | 1.687         | 1.532             |
| 6               | 1.452          | 1.887              | 1.854         | 1.564             |
| 7               | 1.818          | 2.024              | 1.687         | 1.900             |
| 8               | 1.756          | 2.034              | 2.114         | 1.902             |
| 9               | 19.500         | 19.060             | 18.965        | 19.225            |
| 10              | 17.539         | 17.837             | 17.365        | 17.385            |
| 11              | 12.632         | 13.092             | 12.394        | 13.065            |
| 12              | 11.966         | 11.674             | 12.321        | 12.211            |
| 13              | 13.389         | 13.157             | 13.256        | 13.395            |
| 14              | 5.250          | 5.638              | 5.054         | 5.341             |
| 15              | 5.633          | 5.159              | 6.254         | 5.406             |

Figure 7: Prediction of fracture length in different algorithms.

Table 2: Prediction results of fracture lengths for different algorithms.

| Fracture number | Measured value | PSO-BPNN algorithm | SVR algorithm | PSO-RBN algorithm |
|-----------------|----------------|--------------------|---------------|-------------------|
| 1               | 234            | 238                | 236           | 232               |
| 2               | 378            | 369                | 382           | 376               |
| 3               | 286            | 293                | 288           | 289               |
| 4               | 306            | 304                | 302           | 304               |
| 5               | 276            | 280                | 272           | 279               |
| 6               | 233            | 236                | 235           | 230               |
| 7               | 257            | 254                | 261           | 253               |
| 8               | 268            | 267                | 271           | 269               |
| 9               | 254            | 252                | 251           | 252               |
| 10              | 167            | 169                | 169           | 166               |
| 11              | 149            | 151                | 147           | 148               |
| 12              | 241            | 239                | 237           | 243               |
| 13              | 239            | 241                | 242           | 241               |
| 14              | 248            | 250                | 251           | 246               |
| 15              | 305            | 307                | 301           | 303               |
respectively. When the learning factor is small, the particles may linger away from the target area, while the particles can quickly move toward the target area or even exceed the target area when the learning factor is large.

The values of the inertia weight and learning factor in the particle swarm algorithm can change during the optimization process. During the search process, the inertia weight is set to change with the number of iterations. The formula of inertia weight is expressed by the following equation:
\[
\omega_i = \omega_{\text{max}} - i \times \frac{\omega_{\text{max}} - \omega_{\text{min}}}{i_{\text{max}}}
\]  
(9)

where \(\omega_{\text{max}}\) is the maximum value of inertia weight, \(\omega_{\text{min}}\) is the minimum value of inertia weight, \(i\) is the current number of iterations, and \(i_{\text{max}}\) is the maximum number of iterations. The calculation formula of the learning factor is shown in the following equation:

\[
c_1 = c_{1,\text{start}} + \frac{i \times (c_{1,\text{end}} - c_{1,\text{start}})}{i_{\text{max}}}
\]

(10)

\[
c_2 = c_{2,\text{start}} + \frac{i \times (c_{2,\text{end}} - c_{2,\text{start}})}{i_{\text{max}}}
\]

where \(c_{1,\text{start}}\) and \(c_{1,\text{end}}\) are the initial and final values of the learning factor \(c_1\); \(c_{2,\text{start}}\) and \(c_{2,\text{end}}\) are the initial and final values of the learning factor \(c_2\); \(i\) is the current number of iterations; and \(i_{\text{max}}\) is the maximum number of iterations.

In order to illustrate the effect of parameter settings of inertial weight and learning factor on the prediction results of the PSO-RBF model, a classic particle swarm algorithm with fixed parameter values and an improved particle swarm algorithm with parameter value changing were used to predict the permeability and the half-length of the fracture, respectively. The basic parameters of the two algorithms are shown in Table 3. The permeability prediction data and fracture length prediction data of the two algorithms are shown in Tables 4 and 5, respectively. The prediction results of the two algorithms are shown in Figures 8–11, respectively. From Table 4, it can be concluded that when making permeability predictions, the RMSE of the classic algorithm is 0.294 and the RMSE of the improved algorithm is 0.178. From Table 5, it can be concluded that when predicting the
fracture length, the RMSE of the classic algorithm is 3.823 and the RMSE of the improved algorithm is 2.25. Therefore, the improved particle swarm algorithm is applied to the neural network training process in this paper.

7. Conclusions

In this study, a neural network model of PSO-RBN was proposed to predict the permeability and fracture length in a MFHW in western China. A comparison of the performance of the PSO-RBN algorithm and core permeability measurements shows that the presented model can adequately estimate reservoir permeability and fracture length despite the highly nonlinear relationship between reservoir parameters. In addition, the PSO-RBN algorithm and other algorithms are statistically compared in the prediction of permeability and fracture length. The results depict that the presented model has some advantages over the mentioned algorithms. Therefore, the proposed PSO-RBN model can provide more accurate and efficient predictions to reservoir physical parameters.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of the article.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant no. 41672114) and National Science and Technology Major Project (grant no. 2017ZX05009005-002).

References

[1] P. Pourafshary, A. Varavei, K. Sepehrnoori, and A. Podio, “A compositional wellbore/reservoir simulator to model multiphase flow and temperature distribution,” *Journal of Petroleum Science & Engineering*, vol. 69, no. 1-2, pp. 40–52, 2009.

[2] A. Amini and A. J. Schleiss, “Numerical modeling of oil-water multiphase flow contained by an oil spill barrier,” *Engineering Applications of Computational Fluid Mechanics*, vol. 3, no. 2, pp. 207–219, 2009.

[3] Y. Jiang, *Techniques for Modeling Complex Reservoirs and Advanced wells*, Stanford University, Stanford, CA, USA, 2008.

[4] A. Roux, J. Corteville, and M. Bernicot, “Wellsim and pepete: accurate models of multiphase flow in oil wells and risers,” in *Proceedings of the International Meeting on Petroleum Engineering*, Tianjin, China, November 1988.

[5] S. Mohaghegh, A. Popa, and S. Ameri, “Intelligent systems can design optimum fracturing jobs,” in *Proceedings of the SPE Eastern Regional Conference and Exhibition*, Pittsburgh, PA, USA, November 1999.

[6] N. V. Queipo, S. Pinto, N. Rincón, N. Contreras, and J. Colmenares, “Surrogate modeling-based optimization for the integration of static and dynamic data into a reservoir description,” *Journal of Petroleum Science and Engineering*, vol. 35, no. 3-4, pp. 167–181, 2002.

[7] N. V. Queipo, J. V. Goicochea, and S. Pintos, “Surrogate modeling-based optimization of SAGD processes,” *Journal of Petroleum Science and Engineering*, vol. 35, no. 1-2, pp. 83–93, 2002.

[8] S. D. Mohaghegh, H. H. Hafez, R. Gaskari, M. Haajizadeh, and M. Kenawy, “Uncertainty analysis of a giant oil field in the middle east using surrogate reservoir model,” in *Proceedings of the Abu Dhabi International Petroleum Exhibition and Conference*, Abu Dhabi, UAE, November 2006.

[9] A. Mohabbi, R. Kamalpour, K. Keyvanloo, and A. Sarrafzadeh, “The prediction of permeability from well logging data based on reservoir zoning, using artificial neural networks in one of an Iranian heterogeneous oil reservoir,” *Petroleum Science and Technology*, vol. 30, no. 19, pp. 1998–2007, 2012.

[10] M. Jamialahmadi and F. G. Javadpour, “Relationship of permeability, porosity and depth using an artificial neural network,” *Journal of Petroleum Science and Engineering*, vol. 26, no. 1–4, pp. 235–239, 2000.

[11] F. K. Boadu, “Rock properties and seismic attenuation: neural network analysis,” *Pure and Applied Geophysics*, vol. 149, no. 3, pp. 507–524, 1997.

[12] S. Mohaghegh, B. Balan, and S. Ameri, “State-of-the-art in permeability determination from well log data: Part 2-verifiable, accurate permeability predictions, the touch-stone of all models,” in *Proceedings of the SPE Eastern Regional Meeting*, Morgantown, West Virginia, September 1995.

[13] K. Aminian, H. I. Bilgesu, S. Ameri, and E. Gil, “Improving the simulation of waterflooding process with the use of neural networks,” in *Proceedings of the SPE Eastern Regional Meeting*, Calgary, Alberta, Canada, April 2000.

[14] K. Aminian, B. Thomas, and H. I. Bilgesu, “Permeability Distribution prediction,” in *Proceedings of the SPE Eastern Regional Conference*, Canton, Ohio, October 2001.

[15] M. A. Ahmadi, R. Soleimani, M. Lee, T. Kashiwao, and A. Rahadori, “Determination of oil well production performance using artificial neural network (ANN) linked to the particle swarm optimization (PSO) tool,” *Petroleum*, vol. 1, no. 2, pp. 118–132, 2015.

[16] J.-R. Zhang, J. Zhang, T.-M. Lok, and M. R. Lyu, “A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training,” *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1026–1037, 2007.

[17] X. Chen, “Deformation measurement of the large flexible surface by improved RBFNN algorithm and BPNN algorithm,” in *Proceedings of the International Symposium on Neural Networks*, pp. 41–48, Berlin, Heidelberg, 2007.

[18] B. Yang, H. Lu, and L. Chen, “BPNN and RBFNN based modeling analysis and comparison for cement calcination process,” in *Proceedings of the Third International Workshop on Advanced Computational Intelligence*, pp. 101–106, Suzhou, China, August 2010.

[19] M. Saemi, M. Ahmadi, and A. Y. Varjani, “Design of neural networks using genetic algorithm for the permeability estimation of the reservoir,” *Journal of Petroleum Science and Engineering*, vol. 59, no. 1-2, pp. 97–105, 2007.

[20] H. Kaydani, A. Mohebbi, and A. Baghaie, “Permeability prediction based on reservoir zonation by a hybrid neural genetic algorithm in one of the Iranian heterogeneous oil reservoirs,” *Journal of Petroleum Science and Engineering*, vol. 78, no. 2, pp. 497–504, 2011.
[21] P. Tahmasebi and A. Hezarkhani, "A fast and independent architecture of artificial neural network for permeability prediction," Journal of Petroleum Science and Engineering, vol. 86-87, pp. 118–126, 2012.

[22] H. Kaydani and A. Mohebbi, "A comparison study of using optimization algorithms and artificial neural networks for predicting permeability," Journal of Petroleum Science and Engineering, vol. 112, pp. 17–23, 2013.

[23] S. J. Cuddy, "Litho-facies and permeability prediction from electrical logs using fuzzy logic," SPE Reservoir Evaluation & Engineering, vol. 3, no. 4, pp. 319–324, 2000.

[24] N. Hambalek and R. González, "Fuzzy logic applied to lithofacies and permeability forecasting case study: sandstone of narical formation, El furrial field, eastern Venezuela basin," in Proceedings of the SPE Latin American and Caribbean Petroleum Engineering Conference, Port-of-Spain, Trinidad and Tobago, April 2003.

[25] A. A. Taghavi, "Improved permeability estimation through use of fuzzy logic in a carbonate reservoir from southwest, Iran," in Proceedings of the SPE Middle East Oil and Gas Show and Conference, Kingdom of Bahrain, March 2005.

[26] J. S. Lim, "Reservoir properties determination using fuzzy logic and neural networks from well data in offshore Korea," Journal of Petroleum Science and Engineering, vol. 49, no. 3-4, pp. 182–192, 2005.

[27] I. S. Nashawi and A. Malallah, "Permeability prediction from wireline well logs using fuzzy logic and discriminant analysis," in Proceedings of the SPE Asia Pacific Oil and Gas Conference and Exhibition, Brisbane, Australia, October 2010.

[28] S. O. Olatunji, A. Selamat, and A. Abdulraheem, "Modeling the permeability of carbonate reservoir using type-2 fuzzy logic systems," Computers in Industry, vol. 62, no. 2, pp. 147–163, 2011.

[29] A. K. Ilkhchi, M. Rezaee, and S. A. Moallemi, "A fuzzy logic approach for estimation of permeability and rock type from conventional well log data: an example from the Kangan reservoir in the Iran offshore gas field," Journal of Geophysics and Engineering, vol. 3, no. 4, pp. 356–369, 2006.

[30] R. Gholami, A. R. Shahraki, and M. Jamali Paghaleh, "Prediction of hydrocarbon reservoirs permeability using support vector machine," Mathematical Problems in Engineering, vol. 2012, pp. 1–18, 2012.

[31] R. Yin, Q. Li, P. Li, Y. Guo, Y. An, and D. Lu, “GPU-based computation of formation pressure for multistage hydraulically fractured horizontal wells in tight oil and gas reservoirs,” Mathematical Problems in Engineering, vol. 2018, pp. 1–10, 2018.

[32] J. Kennedy, "Particle swarm optimization," Encyclopedia of Machine Learning and Data Mining, Springer, Berlin, Germany, pp. 760–766, 2010.

[33] S. D. Mohaghegh, J. S. Liu, R. Gaskari et al., "Application of surrogate reservoir models (SRM) to an onshore green field in Saudi Arabia, case study," in Proceedings of the North Africa Technical Conference and Exhibition, Cairo, Egypt, February 2012.

[34] S. Mohaghegh, "Virtual-intelligence applications in petroleum engineering: Part 1—artificial neural networks," Journal of Petroleum Technology, vol. 52, no. 09, pp. 64–73, 2000.

[35] S. Mohaghegh, R. Arefi, S. Ameri, K. Amini, and R. Nutter, "Petroleum reservoir characterization with the aid of artificial neural networks," Journal of Petroleum Science and Engineering, vol. 16, no. 4, pp. 263–274, 1996.

[36] E. A. El-Sebakhy, O. Asparouhov, A.-A. Abdulraheem et al., "Functional networks as a new data mining predictive paradigm to predict permeability in a carbonate reservoir," Expert Systems with Applications, vol. 39, no. 12, pp. 10359–10375, 2012.

[37] M. Mohammadpoor, K. Shahbazi, F. Torabi, and A. R. Qazvini Firouz, "A new methodology for prediction of bottomhole flowing pressure in vertical multiphase flow in Iranian oil fields using artificial neural networks (ANNs)," in Proceedings of the SPE Latin American and Caribbean Petroleum Engineering Conference, Lima, Peru, December 2010.

[38] A. A. A. Esmin, R. A. Coelho, and S. Matwin, "A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data," Artificial Intelligence Review, vol. 44, no. 1, pp. 23–45, 2015.

[39] F. Van Den Bergh, An Analysis of Particle Swarm optimizers, Ph.D. thesis, University of Pretoria, Pretoria, South Africa, 2007.