Replacement of numerical simulations with machine learning in the inverse problem of two-phase flow in porous medium

Yu A Goncharova¹ and I M Indrupskiy²,¹
¹National University of Oil and Gas «Gubkin University», Moscow, Russia
²Oil and Gas Research Institute of the Russian Academy of Sciences (OGRI RAS), Moscow, Russia

goncharova97st@mail.ru

Abstract. A possibility to replace full-physics numerical simulations with machine-learning-based algorithms in inverse problems of multiphase flow in porous media is studied on an example of specialized oil-well tests. The performance function is computed at each iteration of the inverse problem solution using an artificial neural network (ANN) instead of forward numerical simulations. Proper ANN structure, training set, and learning algorithm are established and implemented. Achieved approximation quality for the performance function on a test sample is well suitable for the inverse problem solution. Further improvements and alternative ANN formulations are proposed for practical applications.

1. Introduction

Inverse problems of flow in porous media arise in many popular practical applications. The most methodically and computationally difficult are formulations with significant numbers of identified parameters and numerically solved forward problems. In particular, this applies to the important practical problem of history matching of flow simulation models of oil and gas reservoirs and modern numerical models of oil-well tests.

The forward problem consists in determining unknown distributions of phase pressures and saturations, well bottomhole pressures and flow rates at different points in time, based on known properties of the reservoir and fluids. It is based on a mathematical model which in the general case considers a 3D-, multiphase flow in a reservoir with complex geometry, heterogeneous and anisotropic properties. The forward problem is usually solved by numerical methods.

The inverse problem corresponds to the adjustment of the forward problem input data (reservoir and fluid properties) according to the real measurements at wells during reservoir development. The need of inverse problem solution originates from the inherent lack of completeness, accuracy and reliability in the input data for the forward problem: point-wise nature and scale inconsistency of core and log data, indirectness and insufficient resolution of seismic data, etc.

In the case of numerical solution of the forward problem, the inverse problem is formulated and solved in an optimization statement. This means that an objective function is to be minimized which characterizes the misfit between the forward problem solution and the well measurements by adjusting the input data of the forward problem.

A wide range of optimization algorithms is used for solution of inverse problems [1–3]. Despite their significant difference, all of them require solving a series of forward problems with different sets of input data and determining the corresponding values of the objective function within an iterative (or
‘ensemble’, on a set of realizations) procedure of inverse problem solution. Depending on complexity of the mathematical model, chosen numerical method and optimization procedure, the repeated solution of the forward problem may require different computational times and resources. However, in any case, it is the most computationally expensive component of the inverse problem solution. Thus, the ability of fast calculation of the objective function is of extreme theoretical and practical importance.

A certain step in this direction is provided by the proxy models which are implemented in specialized software and additional options of flow simulators for solution of the history matching problems and uncertainty estimation (Schlumberger MEPO®, Roxar Enable®, RFD tNavigator®, etc.). However, the quality of proxy models essentially depends on the number of reference solutions of the forward problem and deteriorates greatly in case of a large number of different input data, as well as a strongly non-linear dependence of solution on them. From this point of view, it is interesting to use an alternative approach, based on machine learning methods, to accelerate calculation of the objective function for given set of model input parameters instead of classical solution of the forward problem using numerical methods.

As an example, the forward and inverse problems were considered of specialized two-phase well tests with water injection into the reservoir to evaluate relative permeability (RP) functions [4, 5]. Despite the relative simplicity of the forward problem formulation compared to the problems of 3D models’ history matching, it reflects the main features of this class of problems: diversity of identified parameters, essential nonlinearity, necessity to involve numerical methods in the solution of forward and inverse problems, and use of weighted least squares objective function for a set of calculated and measured well data.

2. Classical problem formulation
In contrast to conventional problems of well tests based on single-phase flow models, this study considers complex well tests with forced creation of two-phase flows (oil + water) in the reservoir with large variations of water saturation near the wellbore [4, 5]. The set of parameters identified through the solution of the inverse problem includes the RP functions for oil and water. Thus, the forward problem is formulated on the basis of a two-phase flow model. Figure 1 illustrates schematically the difference between the considered problem and the traditional problems of well testing.

![Figure 1. Schematization of the traditional and considered approaches to oil well tests.](image)

A typical study consists of the 4 main stages [5]. At the first stage, until the time point T1, the well produces the reservoir fluid (oil, if no mobile water is present in the reservoir) with a specified (possibly changing) flow rate. For the period from T1 to T2, the well is closed for pressure build-up. From T2 to T3, the well has the injection function: water is pumped into the reservoir with a specified,
Generally changing, flow rate. From \( T_1 \) to \( T_2 \), fluid is produced from the reservoir with a specified (also possibly changing) flow rate: first, almost pure water; then water with an increasing share of oil; and so until the water content in the produced fluid reduces back to initial or almost initial values.

It was shown in previous studies [5–9] that implementation of such a well test with the appropriate hydrodynamic and logging measurements makes it possible to identify the following reservoir parameters through the solution of the inverse problem:

- porosity \( \varphi \);
- permeability \( k \);
- permeability of the ‘skin’ (near-wellbore) zone \( k_c \);
- residual/critical oil saturation \( S_{o}^* \) and connate/critical water saturation \( S_{w}^* \);
- parameters of RPs for oil and water as functions of oil (water) saturation.

Relative phase permeabilities for oil and water are set as parametric dependencies of a certain class, for example, as a power dependence:

\[
\begin{align*}
    k_\alpha & = 0 & S_\alpha & < S_\alpha^* \\
    k_\alpha & = C_\alpha (S_\alpha - S_\alpha^*)^{n_\alpha} & S_\alpha^* & \leq S_\alpha \leq 1 - S_\lambda^*
\end{align*}
\]

where \( k_\alpha \) is the RP, \( S_\alpha \) is the saturation, and \( S_\alpha^* \) is the critical saturation for the phase \( \alpha \); \( S_\lambda^* \) is the critical saturation for another phase (i.e. \( \lambda = o \), if \( \alpha = w \); and \( \lambda = w \), if \( \alpha = o \)). \( n_\alpha \) and \( C_\alpha \) are unknown coefficients, which, together with the critical saturations and the reservoir properties, are to be identified through interpretation of the well test data. The subscripts \( \alpha \) and \( \lambda \) denote the phase: ‘o’ for oil, ‘w’ for water.

In the forward problem, the parameters listed above, along with other parameters of the reservoir and fluids, initial and boundary conditions, are used as input data for calculation of well test data dynamics during the well testing process. Dynamics of bottomhole pressure, near-wellbore water saturation and oil production rate are considered as the controlled well test data. Since some stages of the well test imply the presence of essentially two-phase fluid flows, the forward problem in the simplest (one-dimensional axisymmetric) case is reduced to the following system of nonlinear partial differential equations [5]:

\[
\frac{\partial}{\partial u} \left( \frac{k k_a \rho_a \partial p_a}{\mu_a} \right) = R_o^2 e^{2u} \frac{\partial}{\partial t} (\varphi \rho_a S_a), \quad \alpha = o, w.
\]

Initial conditions are given by

\[
p_o(u)|_{t=0} = p_0, \quad S_w(u)|_{t=0} = S_w^*.
\]

Boundary conditions at the outer contour of the reservoir are

\[
\frac{dp_a}{dt}|_{u=0} = 0, \quad \alpha = o, w.
\]

Here \( p_a \) is the pressure in the phase \( \alpha \); \( \rho_a \) and \( \mu_a \) are the density and viscosity of the phase \( \alpha \); \( \varphi \) and \( k \) are the porosity and permeability of the reservoir; \( u = \ln \frac{r}{R_o} \) and \( t \) are the logarithmic spatial and temporal coordinates; \( p_0 \) is the initial reservoir pressure; \( R_o \) is the radius of the outer boundary of the reservoir.

The influence of the skin effect is taken into account by setting different permeability values for the skin zone and the main part of the reservoir, that is, for \( R_w \leq r \leq R_s \), the value of \( k \) changes to \( k_c \). The boundary conditions at the wellbore correspond to the specified liquid flow rate or water flow rate, depending on the stage of the well test, and are expressed in the form of Neumann boundary conditions for pressure [5].

In the framework of software implementation developed earlier at OGRI RAS, numerical algorithm is used to solve the forward problem based on the finite difference method [5]. The set of difference equations is solved for the variables \([p_o, S_w]\). In the general case, a non-uniform block-centered grid with hundreds of blocks is used along the logarithmic spatial coordinate \( u \) if accurate saturation front
tracking is necessary. To account for vertical inhomogeneity of the reservoir, an axisymmetric 2D formulation in cylindrical \( r-z \) coordinates [5] or a 3D formulation can also be used, which leads to a further increase in the dimension of the forward problem.

The inverse problem consists in finding such values of the identified parameters that ensure the best match of the forward problem solution to the real measurements of the well data dynamics. The minimized objective function is a weighted least squares one with the following form [5]:

\[
J(\vec{u}) = \sum_{j=1}^{n} \left[ C_p (p_j^c - p_j^m)^2 + C_s (S_j^c - S_j^m)^2 + C_Q (Q_j^c - Q_j^m)^2 \right].
\] (5)

Here \( \vec{u} \) is the vector of identified (control) parameters listed above; \( j \) is the sequence number of the well measurement, \( N \) is the number of measurements (in time); \( p_j^c \), \( S_j^c \), and \( Q_j^c \) are respectively the bottomhole pressure, near-wellbore water saturation and oil flow rate at the time of the \( j \)-th measurement. Subscript ‘\( c \)’ corresponds to the values calculated by the forward problem solution, ‘\( m \)’ – to the actually measured data at the well. The weights \( C_p \), \( C_s \), and \( C_Q \) are used to normalize values with different scale, take into account measurement errors and provide control over the influence of different measurement types on the value of the objective function (5).

Regardless of the choice of specific algorithm for minimizing the objective function (5), it is required to determine the value of the objective function \( J(\vec{u}) \) at each iteration of the inverse problem solution for the current values of identified parameters. In the classical approach this corresponds to a separate run of the numerical solver for the forward problem.

Thus, the question is how to omit the computationally expensive numerical solution of the forward problem at each inverse problem iteration. In this study, we use the methods of machine learning for this purpose, namely, the artificial neural network (ANN).

3. Construction of the ANN

ANN is a mathematical model that represents a system of connected and interacting artificial neurons. The structure of a neural network consists of neurons and synapses. A neuron is a computational unit that receives information, performs simple calculations on it, and passes it on. Neurons are interconnected by synapses, which, when transmitting a signal from one neuron to another, change this signal with the help of a single parameter – weight. Thus, a neuron receives a total signal from all previous neurons and processes it using the activation function [10], which normalizes the signal to an acceptable value. More information about neural networks can be found in specialized literature [11]. Figure 2, as an example, shows the structure of the simplest neural network – the perceptron.

The problem we consider in this study is as follows. For known actually measured well operation data and given combination of identified parameter values \( \vec{u}_i \), it is necessary to obtain the value of the objective function (5) using an approximation of the \( J(\vec{u}) \) by a neural network:

\[
J_i = J(\vec{u}_i)
\] (6)

In general, the procedure for construction of the approximating ANN is as follows. First, an optimal ANN structure and appropriate learning algorithm are selected based on specifics of the problem. Then a training sample set is generated using experiment design methods. The training set is a set of different sample pairs (vector of input parameter values \( \vec{u}_i \) – value of the objective function \( J(\vec{u}_i) \)). Next step is the ANN training, i.e. optimization of synapses’ weights on the training set. To assess the quality of approximation for the trained ANN, a separate validation sample set is used made up of pairs (\( \vec{u}_i \), \( J(\vec{u}_i) \)) that were not included in the training, but lie within the admissible range of parameter values.
Let us describe in more detail the parameters for the ANN construction in the problem considered. The input is a vector of the control parameters values, and the output is a single value of the objective function. In this study, reservoir parameters are normalized in accordance with the effective pore space (EPS) concept [5]. Each input vector consists of the values of 7 different parameters: effective porosity φᵢ, effective permeability kᵢ, skin zone permeability kₛᵢ, critical oil saturation Sₒᵢ* (normalized by effective pore volume), coefficient Cₒᵢ of the RP function for water, exponents nᵦw and nₒᵢ for the RP functions for water and oil, respectively. According to the EPS normalization of saturations and RP functions, Sₒᵢ*=0 and Cₒᵢ=1 by definition.

To construct the training and validation sample sets, the following admissible ranges for parameter values were provided:

- φ∈[0.05; 0.3], frac.;
- k∈[0.1; 0.7], Darcy;
- kₛ∈[0.005; 0.05], Darcy;
- Sₒᵢ*∈[0.05; 0.6], frac.;
- Cₒᵢ∈[0.1; 1];
- nᵦw∈[1; 3];
- nₒᵢ∈[1; 3].

For each vector \( \vec{u}_i \) of the training and validation sets, the forward problem was solved using the OGRI RAS computational software implementing the classical numerical algorithm, and the corresponding value of the objective function was obtained using equation (5):

\[
J_i = J(\phi_i, k_i, k_s, S_{o_i}^*, C_{o_i}, n_{w_i}, n_{o_i}).
\]  

(7)

For the test calculations performed in this study, the actually measured well data in the objective function (5) were also replaced by the solution of the forward problem, but with fixed values of the control parameters \( \vec{u}_i \) considered as their ‘true’ values. Figures 3–6 show the graphs of the objective function with respect to individual identified parameters, whereas the values of the other parameters are set to the ‘true’ ones. All the dependencies are smooth, but have different shapes, greatly differ by the range of the objective function values, and are asymmetric with respect to the ‘true’ point (\( J=0 \)).

4. Preparation of the training sample set

Appropriate selection of the training set plays an important role in the quality of function approximation within the admissible range of parameter values. If the approximated function is close to linear by all arguments, a good training set can be constructed as a two-level full factorial experiment (FFE) [12], that is, composed of all possible combinations of maximum and minimum admissible values of the parameters.

In our case, 7 different parameters enter the ANN input; therefore, there are 7 factors. A two-level FFE corresponds to the sample size of \( 2^7 = 128 \) training pairs (\( \vec{u}_i - J(\vec{u}_i) \)). However, taking into account the essentially non-monotonic nature of the objective function dependence on the identified
parameters (see figures 3–6), it is necessary to involve additional combinations (training pairs) using intermediate values of each parameter for adequate approximation of the function. But even a three-level FFE already involves more than two thousand training pairs, which makes it unacceptable for the problem considered.

Based on the problem specifics, we proposed to compose a training set of the two two-level FFEs. In the first and second FFE, respectively, one level is set close to the left or right border of the admissible range of parameter values. And the other level – to the minimum point of the objective function with respect to this parameter (see figures 3-6). Thus, the training set consists of 256 training pairs \((\bar{u}_i - f(\bar{u}_j))\). Figure 7 shows the values of the objective function versus training pair numbers.

When the training set is ready, the parameters must be normalized. For our problem, the natural normalization \(\bar{x}_{ij} = (x_{ij} - x_{\text{min},j})/(x_{\text{max},j} - x_{\text{min},j})\) was used, where \(i\) is the number of the sample vector, \(j\) is the parameter number in the vector, and subscripts ‘min’ and ‘max’ designate minimum and maximum admissible values for the parameter, respectively.

**Figure 3.** Dependence of the objective function on porosity.

**Figure 4.** Dependence of the objective function on critical oil saturation.
5. Determining the ANN structure

The problem considered is characterized by 7 input parameters and 1 unknown value of the objective function that needs to be determined. Therefore, the ANN should have 7 neurons in the input layer and 1 neuron in the output layer.

For better approximation of the function and to avoid learning problems due to the use of unbiased activation function, a bias neuron was additionally added to each layer of the network except for the output layer [11]. Thus, the input layer contains 8 neurons.

Using the ANN theory, the required number of hidden (intermediate) layers and the number of neurons in them was chosen. In our case, one hidden layer is enough to approximate the function. The recommended number of neurons in the hidden layer is calculated by \( k = \sqrt{nm} \), where \( k \) is the number of neurons in the hidden layer, \( n \) is the number of neurons in the input layer; \( m \) is the number of neurons in the output layer, which in our case gives

\[
k = \sqrt{8 \times 1} \approx 3.
\]

By adding the bias neuron to the hidden layer, we obtain the total number of 4 hidden neurons.
Figure 7. The values of the objective function on the training set

The number of synapses (weights) is 32 from the input layer to the hidden one, and 4 from the hidden layer to the output one, 36 total.

As an activation function, the sigmoid function [10] was empirically found to be optimal for the problem considered.

6. ANN training

Batch mode was selected for training of the ANN, which corresponds to the adjustment of the synapses’ weights to minimize the sum of squared approximation errors over all elements of the training set [10]. The target optimization function for our ANN is as follows:

$$\Phi(\vec{w}_1, \vec{w}_2) = \sum_{i=1}^{N}(f(\sum_{j=1}^{4}x_i^j * w_{1j}^i) * w_{2k}^i - J_i)^2 \rightarrow \min,$$

(8)

where

$$f(z) = \frac{1}{1+\exp(-z)}$$

is the sigmoid activation function; $\vec{w}_1$ and $\vec{w}_2$ are the adjusted vectors of weights from the input layer to the hidden one and from the hidden layer to the output one, respectively; $j$ is the index of the neuron in the input layer, $k$ is the index of the neuron in the hidden layer, $i$ is the index of the training vector, $x_i^j$ is the $j^{th}$ input parameter of the $i^{th}$ training vector, $N$ is the number of elements of the training set.

Since the objective function (8) with a large number of adjusted weights can have many stationary points, it is preferable to use stochastic learning (optimization) algorithms. In this study, a genetic algorithm is applied [13]. At the beginning of the ANN training, it is necessary to generate genes (values of neuron weights) for all individuals (different realizations of $\vec{w}_1$ and $\vec{w}_2$) of the initial population [14]. For sufficient diversity of the population (significant discrepancy between the vectors $\vec{w}_1$ and $\vec{w}_2$ of different individuals), generation of weights should have stochastic nature and provide a good approximation to the uniform distribution. The optimal population size is chosen empirically and turned out to be 50 individuals for our problem. For crossing-over (creation of individuals at a new iteration from existing members of the population), the conventional operation of combining 2 different parts of parent individuals in a stochastic order was used [13]. The choice of parent individuals was carried out by the panmixia method, which to the greatest extent guarantees preservation of genetic diversity of the population [13]. As a selection method (choice of the best parent individuals in terms of the objective function optimization), the roulette method was used, which works well on minimization problems and is quite simple to implement [14]. The optimal share of mutants (randomly generated individuals at each iteration [14]) was determined empirically as 5%.
7. Results
Since the training procedure is resource-intensive, the ANN learning in this study was carried out not on the entire training set composed of the two FFEs, but on its subset of 20 randomly chosen elements. The validation set was composed of 11 elements that were not included in the training set. Table 1 gives the values of the objective function on the elements of the training set: calculated with the trained ANN – in the left column, and actual (based on the numerical solution of the forward problem) – in the right column. A similar comparison is given in Table 2 for the validation set. The results are illustrated graphically in figures 8–10: for the training set (figure 8), for the validation set (figure 9), and the joint graph (figure 10).

| J_{ANN} | J_{actual} |
|---------|------------|
| 782.564 | 786.976    |
| 343.514 | 458.428    |
| 645.333 | 788.73     |
| 2360.83 | 2236.83    |
| 979.402 | 994.748    |
| 1309.22 | 1336.78    |
| 1931.85 | 1839.62    |
| 1239.34 | 1224.37    |
| 287.337 | 414.321    |
| 930.721 | 997.023    |
| 1181.71 | 1134.34    |
| 2083.16 | 2137.51    |
| 1341.73 | 1259.32    |
| 416.881 | 452.332    |
| 452.939 | 502.205    |
| 862.189 | 788.397    |
| 1496.97 | 1505.63    |
| 2211.54 | 2145.33    |
| 2324.83 | 2310.63    |

Table 2. Values of the objective function on the validation set

| J_{ANN} | J_{actual} |
|---------|------------|
| 789.366 | 831.587    |
| 813.414 | 709.53     |
| 3378.16 | 3506.3     |
| 3295.74 | 2767.8     |
| 3775.66 | 3506.22    |
| 791.191 | 830.85     |
| 3542.46 | 3807.57    |
| 2080.29 | 2083.16    |
| 1041.67 | 1341.73    |
| 412.003 | 416.881    |
| 592.152 | 452.939    |
From figures 8–10 one can see that, both on the training and validation sets, the behavior of the approximated function is well reproduced. It is natural that on the validation set not all the points show as good approximation as on the training one. However, the relative ordering of the values is correct, which is the most important for inverse problem solution. Note also that the trained ANN adequately performs on the validation set even though the range of the objective function values in it is wider than that in the training set.

8. Conclusions
The results show that machine learning methods can successfully replace numerical solution of forward problems for calculating objective function with its approximation by ANN in inverse problems of multiphase flow. In the case considered, sufficient approximation quality was achieved
even on a training set of a very limited size. As for quantitative estimation, more accurate results can be obtained by using more iterations of the training (genetic) algorithm and increasing the training set to the proposed double FFE. Also speed-up of the algorithm and improvement of learning accuracy may be achieved by changing the selection method in the genetic algorithm, adding a genetic shakeup, or adding more neurons to the network structure in the hidden layer. Thus, there are enough ways to further improve ANN training, and from the dynamics of training one can see that the chosen ANN structure and method of learning provide approximation of the objective function with acceptable accuracy. At the same time, broad prospects for ANN application to multiphase flow in porous media include not only inverse problems, but also uncertainty estimation with multivariate solution of direct problems. However, it is necessary to remember that, in the formulation considered, the ANN requires re-generation of the training set and re-training on each new set of actually measured dynamic well data in the objective function (5). This specifics does not allow current ANN formulation to show greater efficiency in the actual solution of inverse problems than, for example, application of the optimal control (adjoint) methods [2, 5]. Therefore, it is reasonable to dedicate further research to machine learning problems of multiphase flow in more practical formulations. As well, some other interesting potential applications of machine learning at OGRI RAS include parameter estimation for hydrocarbon adsorption models [15], phase behavior studies [16], etc.

Acknowledgements
The results presented in the paper were obtained under the OGRI RAS state research contract.

![Figure 10](image-url)  
**Figure 10.** A joint graph of approximation quality of the ANN on the objects of training and validation sets.

References
[1] Oliver D S, Reynolds A C and Liu N 2008 *Inverse theory for petroleum reservoir characterization and history matching* (New York: Cambridge University Press)
[2] Zakirov E S 2001 *Three-dimensional multiphase problems of forecast, analysis and regulation of oil and gas fields development* (Moscow: Graal) p 302
[3] Oliver D S and Chen Y 2011 Recent progress on reservoir history matching: a review. *Computational Geosciences* 15 185-221
[4] Zakirov S N, Indrupskiy I M, Zakirov E S and Anikeev D P 2002 New approach toward wells
and formations surveys Neftyanoe khozyaystvo – Oil Industry 2002 6 p 113–115

[5] Zakirov S N et al 2009 New principles and technologies for oil and gas fields development. Part 2 (Moscow–Izhevsk: Institute of Computer Research; Regular and Chaotic Dynamics) p 484

[6] Indrupskiy I M, Zakirov E S, Anikeev D P, Ipatov A I, Fakhretdinov R N, Gulyaev D N and Klochan I P 2008 In-situ relative permeability evaluation Neftyanoe khozyaystvo – Oil Industry 2008 5 p 38-42

[7] Zakirov S N, Indrupskiy I M, Zakirov E S, Anikeev D P, Vasilyev I V, Severinov E V, Gaidukov L A, Rodionov A E, Lachugin D S and Tsagan-Mandzhiev T N 2012 Well test for in-situ determination of oil and water relative permeabilities Proc. SPE Russian Petroleum Technical Conference (Moscow) SPE-162011-MS

[8] Zakirov E S, Indrupskiy I M, Vasilyev I V, Anikeev D P, Tsagan-Mandzhiev T N, Rodionov A E, Lachugin D S, Afanasyev V S, Afanasyev S V and Antonovich A A 2016 Complex well test study to evaluate relative permeability functions to oil and water and displacement efficiency in conditions of abnormally low reservoir injectivity (part 1) Neftyanoe khozyaystvo – Oil Industry 2016 9 p 56-60

[9] Zakirov E S, Indrupskiy I M, Vasilyev I V, Anikeev D P, Tsagan-Mandzhiev T N, Rodionov A E, Lachugin D S, Afanasyev V S, Afanasyev S V and Antonovich A A 2017 Complex well test study to evaluate relative permeability functions to oil and water and displacement efficiency in conditions of abnormally low reservoir injectivity (part 2) Neftyanoe khozyaystvo – Oil Industry 2017 10 p 90-93

[10] Dyakonov A G 2010 Data analysis, learning by examples, logic games; WEKA, RapidMiner and MatLab systems (Moscow: Moscow State University Press) p 278

[11] Wasserman P D 1989 Neural computing: theory and practice (New York: Van Nostrand Reinhold)

[12] Akhnazarova S L and Kafarov V V 1985 Experiment optimization methods in chemical technology (Moscow: Higher School of Economics) p 327

[13] Mitchell M 1996 An introduction to genetic algorithms (Boston: MIT Press)

[14] Panchenko T V 2007 Genetic algorithms (Astrakhan: Astrakhan University) p 87

[15] Lobanova O A, Indrupskiy I M and Scherbyak M V 2018 Phase behavior simulation with dynamic multicomponent adsorption Proc. 16th European Conf. Math. of Oil Recovery EAGE ECMOR XVI (Barcelona) P056

[16] Belyakov M Y, Voronov V P, Gorodetskii E E, Grigorev B A, Deshabo V A, Kosov V I, Kulikov V D, Kuryakov V N, Yudin I K and Yudin D I 2012 Thermodynamics of multicomponent mixtures near the liquid-vapor critical point High Temperature 50 4 p 479–87