SPE 112214

The Use of Neuro-Fuzzy Proxy in Well Placement Optimization

Faraj Zarei, SPE, Ayoub Daliri, and Naser Alizadeh, SPE, Amirkabir University of Technology

This paper was prepared for presentation at the 2008 SPE Intelligent Energy Conference and Exhibition held in Amsterdam, The Netherlands, 25–27 February 2008.

Abstract

Optimal well placement is crucial step in oil filed development but it is a very sophisticated process on account of different engineering and geological variables affect reservoir performance and they are often nonlinearly correlated. This study presents an approach where a hybrid optimization technique based on genetic algorithm (GA) and a Neuro-Fuzzy system as proxy was created and used to determine the optimal well locations regarding net present value (NPV) maximization as the objective. Neuro-Fuzzy system was used as proxy to decrease the numbers of costly and time consuming-simulations. Such a system has supplanted a conventional technology in some scientific applications and engineering systems, especially in modeling nonlinear systems. Neuro-Fuzzy modeling is a flexible framework, in which different paradigms can be combined, providing, on the one hand, a transparent interface with the designer and, on the other hand, a tool for accurate nonlinear modeling. The rule-based character of Neuro-Fuzzy models allows for the analysis and interpretation of the result.

Within Hybrid Genetic Algorithm (HGA), a database of the completed simulations is made. This database is used to construct of Neuro-Fuzzy network. Then this network is used to estimate the fitness function at points that no simulations have not been done. This proxy is also able to get better during the optimization each time a new point is verified and visited points database is updated.

A synthetic reservoir was tested and comparisons made among HGA, simple GA and non-proxy using approaches. Results showed that Neuro-Fuzzy system is very reliable proxy to estimate fitness function so the HGA will have a good chance to obtain the optimal place for the well in minimum possible duration.

Introduction

The main task of reservoir engineering team is to develop a plan to recover as much hydrocarbon as possible within ecumenical and physical limits. This task involves optimal well placement and production scheduling. Most of the time, a slightly better decisions in this stage may lead to significant increase in project value. However optimal decision making is not an easy task, because different variables affecting reservoir performance are uncertain and they are often are often nonlinearly correlated.

Numerical simulations used in oil industry provide precise approach to predict reservoir manner and assess the value of opportunities in filed. Those models are derived from complex studies involving a nonlinearly equations with high degree of uncertainty and a large amount of parameters that could be either independent or dependent on each other. The relation between the computed simulations result and the input data is generally highly nonlinear.[1] Hence reservoir simulations normally require large computational effort and considerable time consumption; in a parallel manner the activities connected with reservoir simulators suffer severe limitations that make it difficult with the vigorous development. Also most of the time, the large number of possibilities, constraints on computational resources and the size of the simulation models limit the number of possible scenarios that may be considered.[2] Analysis tools encoded in computer, programs can spend hours or date for processing a single run, depending on their sophistication and features. Moreover, it can be costly to prepare the input data if many hypotheses are going to be considered.[3]

Nowadays numerical simulation is widely used to place new wells. Even with these models, current practices are still the ad-hoc, single-well-configuration-at-a-time approach when infill prospects are sought. In each trial, well configuration is selected based on the intuition of reservoir engineer. For a single-well case, this one-well-at-a-time approach may lead to suboptimal decisions. The problem definitely compounds when multiple producers and injectors are involved in a field development scenario.[4]

In such cases automated optimization is an option. It provides a systematic way to explore a broader set of scenarios and aim at finding very good or optimal ones for some given conditions. In conjunction with specialists, these algorithms provide a powerful mean to reduce the risk in decision-making. Nevertheless, the major drawback in using optimization algorithms is the cost of repeatedly evaluating different exploitation scenarios by numerical simulation.[5]

Researchers have looked into the optimization of well placement using numerical simulation. Beckner and Song1 formulated the problem as a traveling-salesman problem and used simulated annealing to optimize well location and drilling schedule. Bittencourt and Horne[3] investigated optimization of well placement using a hybrid of the genetic algorithms (GA) and the polytope method. Aanonsen et al.[6] coupled a CPU-efficient reservoir simulator with an optimization algorithm and made use of a kriging proxy to find optimum well locations. Pan and Horne[7] also used kriging to decrease the necessary number of simulations required to optimize well location. Rogers and Dowla[8] and Centilmen et al.[9] used neural networks as a substitute for numerical simulation. From an optimization standpoint, most algorithms employed so far are either stochastic or heuristic approaches; in particular, this includes simulated annealing (SA) [10] and GA[11,12]. Some of them have also been combined with deterministic approaches to provide a fast convergence close to the solution; for instance, GA with polytope and tabu search[3] and GA with neural networks.[9,11] In all cases, the authors point out that all these algorithms are still computationally demanding for large scale applications. [5]

For the purposes of well placement, GAs are very appropriate choices. They are able to handled both discrete and continuous parameters. GAs can work with various data structures simultaneously and easily modified for different problems. Also they are also suitable for hybridization with other algorithms.[14] However three advantages off GAs are very important for industrialists. The algorithm returns multiple solutions for further consideration, this is important when the model dos not capture all of the known behavior, that the algorithm is very robust, this is important if it can not be guaranteed that the objective function can always be evaluated successfully, that it is possible to easily parallelized the process, this is attractive as many organizations have many computers doing nothing over night. [15]

In view of the fact that exact objective faction evaluation for each well pattern involves using expensive numerical simulators, methods to predict simulation results with minimum computational effort are very attractive. The most applicable techniques to reduce these problems are using proxy methods such as Spline, Kriging and Artificial Neural Networks. Although these techniques to be useful to solve the problem but they could have inconveniences including high computation efforts needing to calculate high order derivatives and hardship to capture the nonlinear characteristics. Neuro-Fuzzy modeling is a new flexible framework that combination of fuzzy logic and neural net technology is called Neuro-Fuzzy and combined the advantages of the two technologies. In addition, a Neuro-Fuzzy system is a neural network system that is self-training, but uses fuzzy logic for knowledge representation, the rules for behavior of the system, and for training the system.[15] Neuro-Fuzzy techniques can combine different paradigms, on the one hand, a transparent interface with the designer and, on the other hand, a tool for accurate nonlinear modeling. The rule-based character of Neuro-Fuzzy models allows for the analysis and interpretation of the result.

According to advantages of Genetic Algorithm and Neuro-Fuzzy systems, we integrated them in new technique. This Hybrid Genetic Algorithm (HGA) used to in optimization of well pattern in a synthetic reservoir. GA is the principal optimization method and Neuro-Fuzzy technique is a proxy to estimate the response of simulator for different well patterns. Different sets of vertical wells placed through different locations and proxy used to predict simulation results.

Initially a series of simulation with different well patterns is done. Since overall objective was maximizing of Net Present Value (NPV), simulator output data used to NPV calculations. These data used for proxy construction and validation. After training the proxy, that can be used instead of simulator to estimate NPV new well patterns.

Experiments repeated in three configurations. In first case all of the well patterns needed by GA evaluated by proxy. In the second case three of the best individuals in each generation validated with commercial simulator. We used simple GA (without proxy) in third case. Results show that as we respected simple GA returned the best pattern but we had very more run time for the same optimization setting. In contrast HGA was very fast running. Case one had problems to achieve the best answer but proxy validation step (case two) can easily fix the problem with minimum requiring time. In this way GA can search more thoroughly in search area with minimum CPU time. However exhaustive search in many cases is still impossible.

Reservoir Simulations

In this study, we use a 3D black-oil synthetic model. This heterogonous reservoir consists in a $20 \times 40 \times 12$ grid. The existence of 100% water-saturated region in deeper sector and another region with more than 85% of gas saturation characterize this reservoir model. **Fig. 1** illustrates the model in Ternary saturation. We assume development team decided to place four vertical producer wells in this reservoir.

Eclipse 100 commercial simulator[16] was used and linking with Matlab[17] was made through files as following: First of all, necessary data about the possible places for vertical wells (places with at least on layer to complete) and layers to be completed at these points should be obtained. The required information can obtain from MODEL.PRT output file. This file includes initial oil saturation throughout reservoir. Cells with minimum 30% of oil saturation selected as valid completion points. Running simulator we obtained initial oil saturations then screening criteria applied. 494 wellhead points remained out of 800 initial wellhead points. These were points with al least one suitable layer.

A table with 14 columns was made. Two first columns are I, J of the wellhead and the other 12 columns indicate valid completion layers. To simplify the problem, we assume same control data for all layers production. However it is not hard to define specific control data for each completion layer at every well.

Data file structure

Data file and its included files contain all essential data and configurations used by simulator (grid size, grid type, cell dimensions, active cells, geologic parameters, initial conditions). Matlab modifies the adjust.txt file which is included in the *.Data Eclipse input file. This is done to make simpler the modifications and it allows changing the well pattern of the Eclipse input file without having to change the whole Data file. The adjust.txt file is the schedule section of the data file and consists mainly in well data (placement, completion, control parameters) and the time steps. The program modifies I, J parameters in the Eclipse, WELSPECS and COMPDAT keywords. WCONPROD have to be modified as well, together with WELSPECS and COMPDAT.

Database Preparation

After autonomous preparation of Data file, Matlab oblige Eclipse to simulate the case. When the simulation is done, the program reads the RSM (simulation output) file and obtains the oil, gas and water flow for the field during time steps for ten years and then Net Present Value (NPV) of pattern is calculated.

Random number generator of Matlab produces 500 sets containing four members each between 1 and 494. These numbers represent a possible wellhead place for new vertical wells. Layers to be completed can get from prepared completion table as well. After revising adjust.txt, program runs simulator and reads its .RSM output file and calculates case NPV. In this regard needing database for proxy construction could be provided.

Objective Function

An optimization procedure requires the characterization of the function to be optimized (minimized or maximized), known as the objective function, as well as the choice of an appropriate optimizing method. The complexity of predicting hydrocarbon production profiles requires the use of reservoir simulators. So, the simulator must be part of the evaluation of the objective function.[3] In this study we use Net Present value (NPV) as objective function.

NPV is a method used to evaluate the positive and negative cash flow of an investment alternative using present worth calculations that requires an analytical approach of systematically and quantitatively evaluating all of the economic considerations that affect the economic potential of the investment. The NPV of an investment alternative is determined by calculating the present worth of all the future net cash flows and summing them. It is based on the economic equivalence concepts presented and highly dependent on the interest rate (commonly referred to as the discount rate) chosen to determine the time value of money. [18]

The corresponding NPV’s from each well pattern calculated from simulation outputs. Calculation parameters are given in **Table 1**. Unfortunately due to lake of commercial figures for cost of oilfield services, current version of the program assumes similar completion and production costs all wells.

Neuro-Fuzzy Systems

Nowadays researchers are using artificial intelligence ways such as neural network and fuzzy logic to model complex system.

Artificial Neural Networks (ANNs) imitate biological information processing systems. They are typically designed to carry out a nonlinear mapping from a set of inputs to a set of outputs and use a dense interconnection of simple processing elements analogous to biological neurons[19]. They are non-programmed adaptive information processing systems that can autonomously develop operational capabilities in response to an information environment[19,20]. Neural networks are self-learning model-less systems that learn from the underlying relationships of data and no need to know data relationships[19]. They modify their behavior in response to the environment, and are ideal in cases where the required mapping algorithm is not known and tolerance to faulty input information is required. [21] The essential reason for using an artificial neural network in first choice to other likely methods of solution is that there is an expectation that it will be able to provide a rapid solution to a significant problem. Depending on the type of problem being considered, there are often satisfactory alternative proven methods capable of providing a fast assessment of the situation. [21] Artificial Neural Networks are not universal solutions to all problems. They are really just an alternative mathematical device for rapidly processing information and data that applicable to model various systems. [19] Despite these advantages they are some limitation. One of these limitations is that they unable to handle linguistic information. Another is inability to manage imprecise or vague information and resolve conflicts. Difficulties to reach global minimum is one of the most common problems in using neural networks[20].

Fuzzy logic is another way of artificial intelligence. Fuzzy logic was first developed by Zadeh for representing uncertain and imprecise knowledge[20]. It provides an approximate but effective means of describing the behavior of systems that are too complex, ill-defined, or not easily analyzed mathematically. It involves fuzzification, fuzzy inference, and defuzzification. The fuzzification process converts a crisp input value to a fuzzy value. The fuzzy inference is responsible for drawing conclusions from the knowledge base.[22] The defuzzification process converts the fuzzy control actions into a crisp control action. Fuzzy logic uses graded statements rather than ones that are strictly true or false. It attempts to mimic human decision making to handle vague concepts. [21] Thus, fuzzy logic provides an approximate but effective way of describing the behavior of systems that are not easy to describe precisely. Fuzzy logic has an ability to deal with imprecise or imperfect information and resolving conflicts by collaboration, propagation and aggregation[20]. Based on the capability of natural language processing and programming improve knowledge representation and uncertainty reasoning. Thus it is sufficient for modeling of complex, non-linear. [21]

Neural networks, fuzzy logic have shown capability on many problems, but have not yet been able to solve the really complex problems completely. It is useful to combine neural networks, fuzzy systems for compensate the demerits of one technique by the merits of another technique. Neuro-Fuzzy refers to the combination of fuzzy set theory and neural networks that mimic human decision making process with the advantages of both that can handle any kind of information (numeric, linguistic, logical, etc.) and can manage imprecise, partial, vague or imperfect information to resolve conflicts by collaboration and aggregation[20,21]. Neural networks and Fuzzy logic have some common features such as distributed representation of knowledge, model-free estimation, ability to handle data with uncertainty and imprecision etc. Fuzzy logic has tolerance for imprecision of data, while neural networks have tolerance for noisy data. A neural network learning capability provides a good way to adjust expert knowledge and it automatically generates additional fuzzy rules and membership functions to meet certain specifications.[23] This reduces the design time and cost. On the other hand, the fuzzy logic approach possibly enhances the generalization capability of a neural network by providing more reliable output when extrapolation is needed beyond the limits of the training data. [24]

Mathematical Background

The Neuro-Fuzzy system consists of the components of a conventional fuzzy system except that computations at each stage is performed by a layer of hidden neurons and the neural network learning capacity is provided to enhance the system knowledge (**Fig. 2**). Without loss of generality, a linguistic fuzzy logic model described by a base of *M* fuzzy rules may be given by the following generic rule[20]:

$Rule\,1: IF\ x_1\ is\ A_{11}\ and\ x_2\ is\ A_{21}\ ...\ and\ x_m\ is\ A_{m1}\ THEN\ y\ is\ B_1,$

$Rule\,2: IF\ x_1\ is\ A_{12}\ and\ x_2\ is\ A_{22}\ ...\ and\ x_m\ is\ A_{m2}\ THEN\ y\ is\ B_2,$

…

$Rule\ n: IF\ x_1\ is\ A_{1n}\ and\ x_2\ is\ A_{2n}\ ...\ and\ x_m\ is\ A_{mn}\ THEN\ y\ is\ B_n,$

where A_{ji} (j=1, 2, …, m; i=1, 2, …, n) and B_i are fuzzy subsets of x_j and y, respectively and the subscript i corresponds to the ith fuzzy rule.

When an observation ($x_1, x_2, ... x_m$) is given, a fuzzy inference consequence y can be obtained by using the product-sum-gravity fuzzy reasoning method as follows[20, 21]:

$$h_i = A_{1i}(x_1)A_{2i}(x_2)...A_{mi}(x_m) = \prod_{j=1}^{m} A_{ji}(x_j) \qquad (1)$$

$$h_i = A_{1i}(x_1)A_{2i}(x_2)...A_{mi}(x_m) = \prod_{j=1}^{m} A_{ji}(x_j) \qquad (2)$$

$$y = \frac{\left(\sum_{i=1}^{n} h_i s_i y_i\right)}{\left(\sum_{i=1}^{n} h_i s_i\right)} \qquad (3)$$

Where h_i (i=1, 2, …, n) is the agreement of the antecedent of the ith fuzzy rule, s_i is the area of B_i, and y_i is the center of B_i .

In order to optimize parameters in a fuzzy system, gradient-descent training algorithms known from the area of neural networks can be employed. [21]

Typically, smooth antecedent membership functions are used, such as the Gaussian functions[20]:

$$A_{ji}(x_j) = \exp\left(\frac{-(x_j - a_{ji})^2}{\sigma_{ji}}\right) \qquad (4)$$

And

$$B_i(y) = \exp\left(\frac{-(y - y_i)^2}{2\sigma_i^2}\right) \qquad (5)$$

Where a_{ji} and σ_{ji} (j=1, 2, … m; i=1, 2, … n) are the center and width of A_{ji}, respectively, as shown in **Fig. 3**, y_i and σ_i (i=1, 2, … n) are the center and width of B_i , respectively.

When the training input-output data ($x_1, x_2, ... x_m ; y^*$) are given for a fuzzy system model, it is well-known to use the following objective function E for evaluating an error between y^* and y, which can be regarded as an optimum problem[20, 21]:

$$E = \frac{(y^* - y)^2}{2} \tag{6}$$

Where y^*,y are desired output value and corresponding fuzzy inference result receptively. In order to minimize the objective function E, gradient descent method used for tuning the parameters of the fuzzy rule as follows: [2,19]

In (1) the fuzzy subsets A_{ji}(j=1, 2, …, m; i=1, 2, …, n) on the antecedent parts and B_i on the consequent parts are of Gaussian type as (4) and (5), respectively. Also, in (5), let $\sigma_i = \sigma$ be a constant. Then, (3) can be rewritten as[20,21]:

$$y = \frac{\left(\sum_{i=1}^{n} h_i y_i\right)}{\left(\sum_{i=1}^{n} h_i\right)} \tag{7}$$

Where y_i (i=1, 2, …, n) stands for the center of fuzzy subsets B_i .

By using the back-propagation algorithm, Wang and Mendel formulate the following Neuro-Fuzzy learning algorithm for updating the center a_{ji} and width σ_i of A_{ji} (j=1, 2, …, m; i=1, 2, …, n) and the center y_i of B_i [20]:

$$\begin{aligned} a_{ji}(t+1) &= a_{ji}(t) - \alpha \frac{\partial E}{\partial a_{ji}(t)} \\ &= a_{ji}(t) - \alpha \frac{(y^* - y)(y_i - y)h_i(x_j - a_{ji})}{\sigma^2{}_{ji} \sum_{i=1}^{n} h_i} \end{aligned} \tag{8}$$

$$\begin{aligned} \sigma_{ji}(t+1) &= \sigma_{ji}(t) - \beta \frac{\partial E}{\partial \sigma_{ji}(t)} \\ &= \sigma_{ji}(t) - \beta \frac{(y^* - y)(y_i - y)h_i(x_j - a_{ji})}{\sigma^3{}_{ji} \sum_{i=1}^{n} h_i} \end{aligned} \tag{9}$$

$$\begin{aligned} y_i(t+1) &= y_i(t) - \gamma \frac{\partial E}{\partial y_i(t)} \\ &= y_i(t) - \gamma \frac{(y^* - y)h_i}{\sum_{i=1}^{n} h_i} \end{aligned} \tag{10}$$

Where α , β and γ are the learning rates which are regarded as the constants in the learning process, and t means the learning iteration.

Proxy Construction

500 of visited patterns which simulator had did numerical simulations, used to proxy construction. Four hundred of them, used for training the Neuro-Fuzzy proxy. As to be seen in the **Fig. 4** output of Neuro-Fuzzy model can trace the output of commercial simulator. Therefore we can use this Neuro-Fuzzy model to generate the new data. Fig. 5 shows the percent of error of each data as to be seen maximum error is less than two percent.

After training the Neuro-Fuzzy we use 100 samples for test the Neuro-Fuzzy model. Fig. 6 shows the real data and prediction of Neuro-Fuzzy model. Fig. 7 shows the error of each of pattern. As to be seen the maximum error is less than 5%. Because of the Neuro-Fuzzy model was not trained with this new data the error of perdition for this newer data increased but the maximum error yet less than 5%. This error can be lesser with more points in proxy database and this is happened whenever a call is made to the numerical simulator during optimization process.

Genetic Algorithm Model

One of the most effective techniques for oilfield optimization problems is the genetic algorithm (*GA*). It is a successful symbiosis of stochastic and gradient methods. [25] With reference to a problem in view for the same time of calculations, the *GA* (in comparison with other methods of optimization) gives solution closer to a global optimum owing to its property to develop and improve the optimization problem successful solutions. [26] Genetic Algorithms are search algorithms based on the mechanics of natural genetics and natural selection. They used “survival of the fittest” concept analogous to natural evolutionary mechanisms, combined with a structured information exchange. [27]

For the purposes of well placement, GAs are very appropriate choices. They are able to handled both discrete and continuous parameters. GAs can work with various data structures simultaneously and easily modified for different problems. Also they are also suitable for hybridization with other algorithms.[14] However three advantages off GAs are very important for industrialists. The algorithm returns multiple solutions for further consideration, this is important when the model dos not capture all of the known behavior, that the algorithm is very robust, this is important if it can not be guaranteed that the objective function can always be evaluated successfully, that it is possible to easily parallelized the process, this is attractive as many organizations have many computers doing nothing over night.
GAs are robust search algorithms based on the mechanics of natural selection. Survival of the fittest among a population of individuals with a structured yet stochastic exchange of information is the basic idea of method. Selection criteria, and reproduction approaches replicate natural life and used as operatives in this artificial environment. The mechanism is obtained from population of individuals (solutions) represented by chromosomes where each one is related to a fitness value (objective function value). These chromosomes are presented to an evaluation procedure with selection, reproduction, crossover and mutation in several iterative sequences. At the end of the evaluation procedure, the best chromosome of population supplies the optimized solution.
Following operations is done in a cycle until the stopping criteria are reached .[28]

1. Evaluation

In the evaluation step all the chromosomes are evaluated. This evaluation can be as complex as desired, and can incorporate technical and economical functions. This evaluation ranks all the chromosomes from best to worst.

2. Selection

Depending on each chromosome fitness, select the ones to build the reproducing set in the population.

3. Reproduction

In this step the new generation of chromosomes is created from the parent chromosomes. The mating and survival of the chromosomes is based upon their evaluation.

4. Replacement

Replace some or all of the original population with new chromosomes. *Selection* can use different selection Schemes
Further information about genetic algorithm can be found in the References (4,27,28).

Optimization production wells

The optimization problem consists in search to best positioning the wells to be drilled into petroleum field. The system model in this work consists in three main modules;
A Neuro-Fuzzy proxy to estimate simulation results (A) which constructed and tuned in above section, Optimization module containing the genetic algorithm (B) and the objective function module composted by a reservoir simulator and the economical net present value(NPV) model (C). The iterative loop process is performed as following:

- Genetic algorithm generates the population where each individual of population is a proposal alternative to be evaluated by NPV computing.
- To perform the evaluation for an individual, this is submitted to the reservoir simulator Neuro-Fuzzy proxy to obtain NPV of the alternative.
- Three best individuals validated with the commercial reservoir simulator.
- To close the loop the already evaluated NPV returns to optimization module, as the objective function for alternative.

Fig. 8 shows the framework of the optimization system and the purposed iterative process. General settings for GA are listed in **Table 2**.
The configuration four wells was optimized with the HGA. In these cases, it is not feasible to carry out exhaustive runs because the search-space size is very large, and a very large number of simulations would be necessary. Runs were iterated for 30 generations before termination. Results are given in terms NPV.
We used above configuration to optimize four production wells in a synthetic oilfield. This work is done through three different approaches:

Without using proxy (simple GA)

We do not use any proxy in this approach and all the individuals directly submitted to simulator. This means that we have to evaluate about 480 numerical simulations. However, due to construction visited points database, the number of such evaluations reduced to 327 in this situation. Using a Celeron(R) 2.67GHZ and 512 megabyte of RAM, each simulation took about 150 seconds. These runs took about 13.6 hours. Final NPV was 78.1 million U.S $. **Fig. 9** shows the results.

Without verification

In this method all individuals submitted to the proxy. Verification for the results was not done. Genetic Algorithm, in this way answered the problem almost instantly. **Fig. 10** shows results in this case. Best pattern proposed by algorithm, presented 75.1 million U.S $. In comparison with simple GA, the NPV in this case, is 3.8% less.

With verification

Three of the best individuals submitted to numerical simulation to verify the objective function. Run time for this case is a little more than the first case. Because the procedure needs to perform number of numerical simulations in each generations. At 30 generations the results was worse than simple GA.

But In this case, we can easily increase number of generations without fear of long run time. Also we can frequently repeat optimization processes with different settings to assure about merit of results. We continued optimization to 40 generations. Due to described database, number of such evaluations was 107 instead of 120 in this situation. This means 4.4 CPU time. Final NPV in this case was 78.3 million U.S $. Notice that, all the optimizations done with same GA setting and we just increased generation numbers here. **Fig. 11** shows the results.

Conclusions

Results Showed that Neuro-Fuzzy Systems as proxy models can be a powerful tool to save time in analysis and calculation of reasonable options. However, the advantages and accuracy of application of Neuro-Fuzzy as proxy model has to be assessed for the particular reservoir.

Such systems are not general substitution tool for simulation models. The purpose of proxy model is to find merit candidates to submit for numerical simulation. This can reduce the need for time consuming processes. Particularly regarding exhaustive applications, such as well placement optimization, the time saving due to proxy models can be very important for the success of optimization algorithm.

Case study results proved that Neuro-Fuzzy system is very reliable proxy to estimate fitness function so the HGA will have a good chance to obtain the optimal place for the well in minimum possible duration.

Acknowledgment

We would like to thank Petroleum University of Technology at Tehran for allowing us to use their ECLIPSE.

References

1. Zangl, G., Garf, T., Al-Kinani, A.: "*Proxy Modeling in Production Optimization,*" paper SPE 100131, presented at 2006 SPE Europe/ EAGE Annual Conference and Exhibition, Vienna, Austria, June 12-15.
2. Silva, C.P. *et al*.: "*Use of Neuro-Simulation techniques as proxy to reservoir simulator: Application in Production history matching*", *J of Petrol. Sci. Eng.* (June 2007), 273.
3. Bittencourt, A. C., Horne, R. N.: "*Reservoir Development and Design Optimization,*" paper SPE 38895, presented at 1997 SPE Annual Technical Conference and Exhibition, San Antonio, Texas, 5-8 October.
4. Badru, O., Kabir, C. S., : "*Well Placement Optimization in Field Development,*" paper SPE 84191, presented at 2003 SPE Annual Technical Conference and Exhibition, Denver, Colorado, 5-8 October.
5. Bangerth, W. *et al*.: "*On optimization algorithms for the reservoir oil well placement problem*", Journal of Computational Geosciences. (September 2006), 303.
6. Aanonsen, S.I. *et al.: "Optimizing Reservoir Performance Under Uncertainty With Application to Well Location,"* paper SPE 30710 presented at the 1995 SPE Annual Technical Conference and Exhibition, Dallas, 22–25 October.
7. Pan, Y. and Horne, R.N.: "*Improved Methods for Multivariate Optimization of Field Development Scheduling and Well Placement Design,*" paper SPE 49055 prepared for presentation at the 1998 SPE Annual Technical Conference and Exhibition, New Orleans, 27–30 September.
8. Rogers, L.L. and Dowla, F.U.: "*Optimal Groundwater Remediation Using Artificial Neural Networks with Parallel Solute Transport,*" *Water Resources Research* (1994) 30, No. 2, 458.
9. Centilmen, A. *et al*.: "*Applications of Neural Networks in Multiwell Field Development,*" paper SPE 56433, presented at the 1999 SPE Annual Technical Conference and Exhibition, Houston, 3–6 October.
10. Becker, B. L. and Song, X.: "*Field development planning using simulated annealingoptimal economic well scheduling and placement ,*" paper SPE 30650, presented at the 1995 SPE Annual Technical Conference and Exhibition, Dallas, Texas, October 22-25.
11. Güyagüler, B. and Horne, R. N.: "*Uncertainty assessment of well placement optimization*" Paper SPE 71625. presented at the 2001 SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, 30 September-3 October 2001.
13. Yeten, B. *et al.: "Optimization of nonconventional well type, location, and trajectory."* Paper SPE 86880, SPE Journal (September 2003) 200.
14. Güyagüler, B.: "*Optimization of Well Placement and Assessment of Uncertainty,*" Ph.D. Thesis, Stanford University, Stanford, California, (2002).
15. Nikravesh, M., Aminzadeh, F., Zadeh L.A.: "*Soft computing & intelligent data analysis in oil exploration, (Volume in the developments in petroleum science series, Vol. 51)*" ,1 edition, Elsevier Science, (2003) 52.
16. Eclipse2004a *Reference Manual*. Schlumberger 2004.
17. Matlab *User's Guide*. MathWorks, 2006.
18. Ahmed, T. and McKinney, P.*: "ADVANCED RESERVOIR ENGINEERING"* , 1 edition, Gulf Professional Publishing (2004) 372,
19. Higgins, C.M., Goodman, R.M.: "*Learning Fuzzy Rule-Based Neural Networks for Function Approximation,*" *Proc. of IJCNN, Vol. 1* (1992) 251.

20. Shia, Y., Mizumoto, M.: *"Some considerations on conventional neuro-fuzzy learning algorithms by gradient descent method," J. of Fuzzy Sets and Systems* (2000), 51.
21. Shia, Y., Mizumoto, M.: *"A new approach of neuro-fuzzy learning algorithm for tuning fuzzy rules," J. of Fuzzy Sets and Systems* (2000) , 99.
22. Cerrada, M., Aguilar, J., Colina E., Titli, A..: *"Dynamical membership functions: an approach for adaptive fuzzy modeling. J. of Fuzzy Sets and Systems* (2005), 51.
23. Jain, L.C., (Editor) .: *"Sft Computing Techniques in Knowledge-Based Intelligent Engineering Systems"*, Springer-Verlag, Heidelberg (1997).
24. Aguilar, J., Cerrada, M.: *"Fuzzy classifier system and genetic programming on system identification," Proc. of the Genetic and Evolutionary Computation Conference GECCO*, (2001), 1245.
25. Larionov, A. *et al.: "Methodology of Optimal Well Pattern, Location and Paths in Productive Formations During Oil and Gas Fields Development Planning"* paper SPE 104326, presented at the 2006 SPE Russian Oil and Gas Technical Conference and Exhibition, October 3-6, Moscow, Russia.
26. Voronovsky, G. K., *et al.: "Genetic Algorithms, Artificial Neural Nets and Virtual Reality Problems"*, OSNOVA, Kharkov, Ukraine (1997).
27. Goldberg, D.: *"Genetic Algorithms in Search, Optimization and Machine Learning,"* Addison-Wesley, (1989) 1.
28. Rawlins, Gregory J.E.: *Foundations of Genetic Algorithms*. Morgan Kaufmann Publishers, Inc., San Mateo, USA (1994).
29. Goldberg, D.E. and Deb, K.: "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms", *Foundations of Genetic Algorithms*, pages 69-93. Morgan Kaufmann Publishers, Inc., San Mateo, USA (1994).

Tables

Table 1: Parameters for income NPV calculations.

parameter	value
discount rate	10
oil price, U.S. $/bbl	70
Gas price, U.S. $/MSCF	5
Water-handling Cost U.S. $/bbl	3
Operation Cost U.S. $/day	60,000
Well cost U.S. $/Well	3,000,000

Table 2: Genetic Algorithm setting

GA Parameters	
Population Size	16
Data strucrute	Integer
Crossover probability	0.8
Crossover type	Double Point
Mutation type	Uniform
Selection method	Rank-Based
Number of elitists	1

Figures

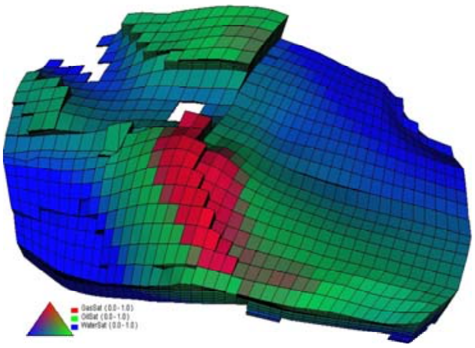

Fig. 1. Synthetic reservoir model

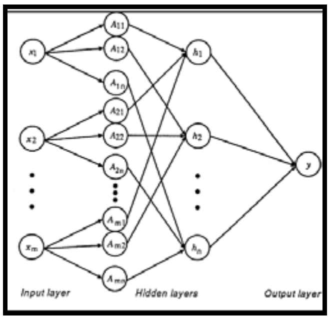

Fig. 2. Neural networks of the fuzzy system model under the conventional Neuro-Fuzzy learning algorithms[20].

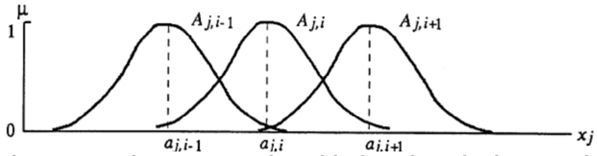

Fig. 3. Gaussian-type member ship functions for input variable xj.

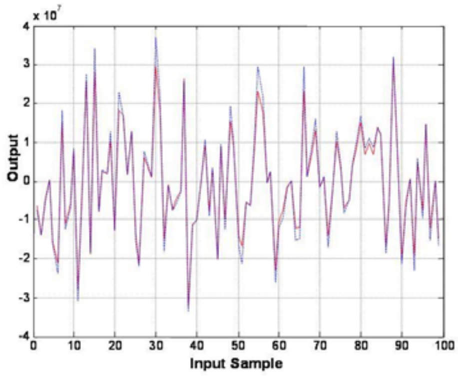

Fig. 4. The first 100 samples of training data. Solid red line is output of Neuro-Fuzzy model that trained and blue dash line is the real data.

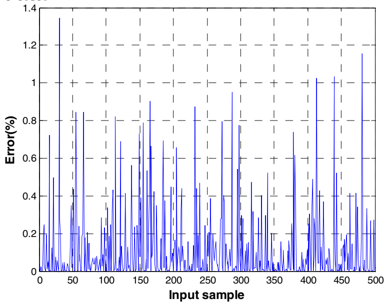

Fig. 5. The percent of error between real data and prediction of Neuro-Fuzzy model for train data after training.

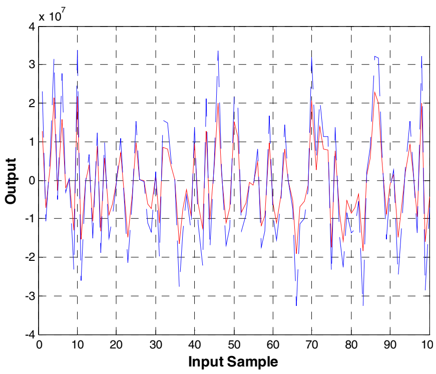

Fig. 6. The output of Neuro-Fuzzy model and real output for 100 sample test data. Solid line is output of Neuro-Fuzzy model that trained and the dash line is the real data.

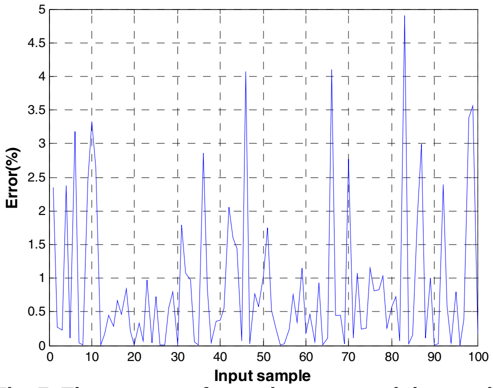

Fig. 7. The percent of error between real data and prediction of Neuro-Fuzzy model for test data.

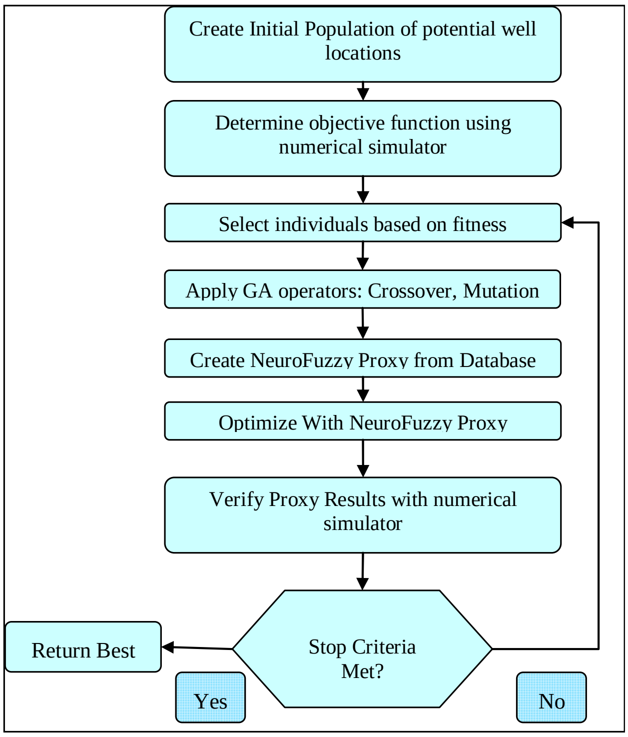

Fig. 8. Flowchart of hybrid Genetic Algorithm

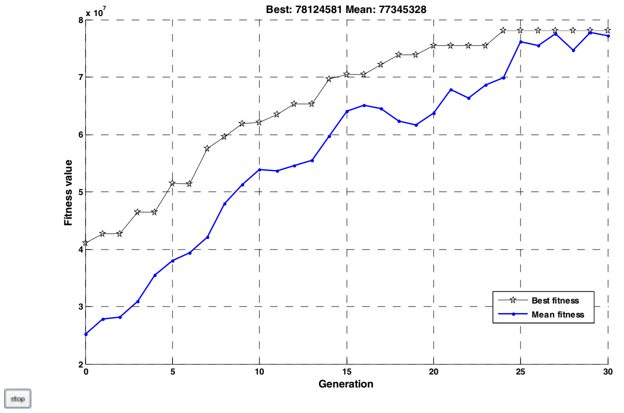

Fig. 9. Simple Genetic Algorithm

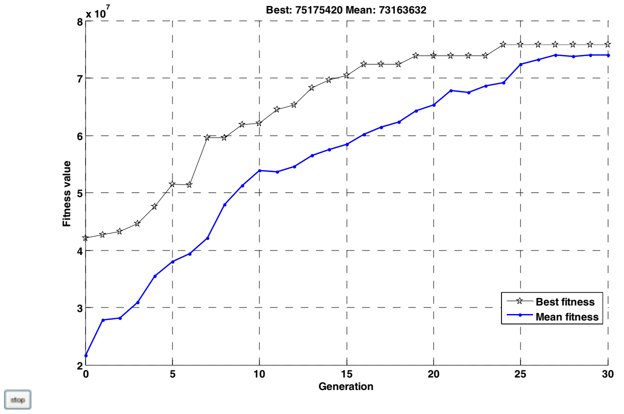

Fig. 10. Genetic Algorithm results without verification

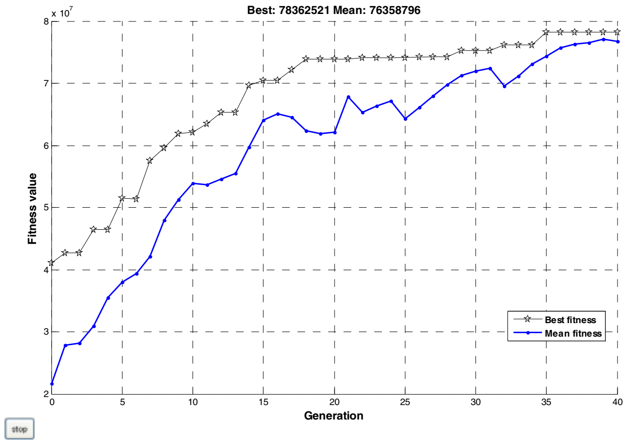

Fig. 11. Genetic Algorithm results with verification