International Journal of Soft Computing 1(3): 207-214, 2006

A Comparison of Soft Computing Methods for Reservoir Simulation

Guadalupe Janoski, Srinivas Mukkamala and Andrew H. Sung
Department of Computer Science, New Mexico Institute of Mining
and Technology, Socorro NM, 87801, USA

Abstract: As time progresses, more and more oil reservoirs reach maturity; consequently, secondary and tertiary methods of oil recovery have become increasingly important in the petroleum industry. This reality has increased the industry's interest in using simulation as a tool for reservoir evaluation and management to minimize costs and increase efficiency. This study presents and compares several control methods in regards to the well-known reservoir simulation task of history matching that is performed to calibrate simulators.

Key words: Soft computing, resevoir simulation

INTRODUCTION

As we enter the new century, the petroleum industry is becoming increasingly dependent on secondary and tertiary methods of oil recovery. This necessity has added to the industry's interest in using simulation as a tool for reservoir evaluation and management to minimize costs and increase efficiency. This study presents that a combination of soft computing algorithms and cluster computing techniques provides realistic hope for obtaining accurate simulation results in a cost-effective fashion. In particular, we show that parallelized genetic algorithms can be used to perform reservoir production history matching and obtain solutions efficiently.

An important step in calibrating a petroleum reservoir simulator (in our case MASTER, developed by the U.S. Department of Energy), is to perform history matching on a particular reservoir, or field[1,2]. History matching predicts the production of a petroleum reservoir based on its past history. Initial calibration of the simulator is achieved by matching simulator predicted production curves (consisting of the output of oil, gas, water in our experiment) to the reservoir's historical production (in our problem a set of data spanning 1960-1991). This attempted curve matching is named history matching. While appearing simple, it is an extremely computationally complex problem. For example dealing with only a small 8 production well section, which we wish to match, we must deal with a search space of over 2^{12954} different possible solutions.

Part of the problem overhead is that fact that we must include 17 wells in the surrounding area for environmental data and use a multi-layer grided cube consisting of 7 layers, 25 simplified grid regions, with each grid area having over 32 parameters with real number ranges. This is easily visible by using the single level map in Fig. 1 of the well layout[3].

This leads us to three problems. Firstly, we have an enormous data space in which we must locate the best solution, since by nature of the simulator and the data tracked we may never reach an exact solution.

Secondly, we would never be able to solve the problem using traditional methods (trail and error with manual adjustment of parameters, using a single computer) without exponentially increased computational power. Therefore we must hunt for solutions smartly through quick, parallelized searches-thus the ARIA controller (described later) and usage of the cluster.

Thirdly there exists a need for the reduction or elimination of human intervention in the simulation process. The need of a human simulation engineer to manually perform history matching to calibrate a simulator often becomes the most expensive part of the task. Minimizing the necessity for such intervention is thus of high importance, in terms of both cost and efficiency. This study address this by using a fuzzy/genetic approach for a parameter control to obtain a history match.

In dealing with these problems, we know that the size of the problem will never be reduced. In fact it will only increase with time, since more accurate simulation results depend on finer grids. The other two obstacles, however, can be overcome by implementing a controller for automatic parameter adjustment to minimize human intervention and executing the integrated simulator-controller on a cluster of computers.

This study describes our initial investigation pf performing history matching for a reservoir in southern New Mexico that was the site of a CO_2 injection project. Promising preliminary results that show the potential of

Corresponding Author: Guadalupe Janoski, Department of Computer Science, New Mexico Institute of Mining and Technology, Socorro NM, 87801, USA

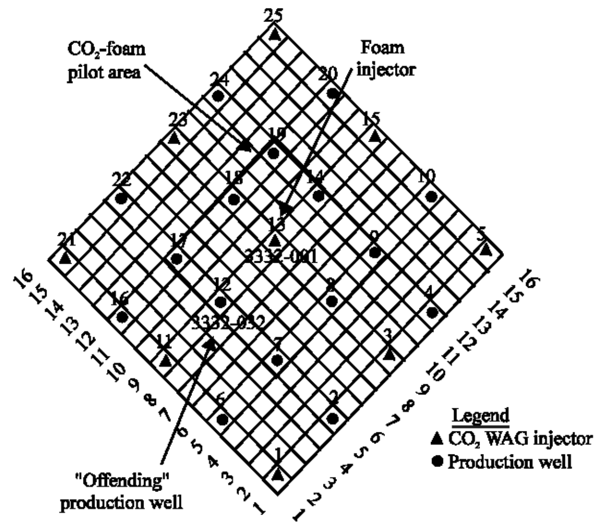

Fig. 1: Well map. The above image is a well map of the area that was chosen to be used for simulation. The interior 8 production wells [7, 8, 9, 12, 14, 17, 18, 19] were the wells that were to be matched, while the remaining wells were provided for environmental data

our approach are presented. In the following sections, we describe the basics of the simulator and the history match model and several methods for parameter adjustment.

Reservoir and simulator background: To advance the CO_2-foam technology for improved oil recovery, a pilot area in EVGSAU (covering 7025 acres, in Lea County, New Mexico) was selected in 1990 as a site for a foam field trial to comprehensively evaluate the use of foam for improving the effectiveness of CO_2 injection projects. Specifically, the prime directive of the foam field trail was to prove that a foam could be generated and that it could aid in suppressing the rapid CO_2 breakthrough by reducing the mobility of CO_2 in the reservoir. Operation of the foam field trial began in 1991 and ended in 1993. The response from the foam field trial was very positive, it successfully demonstrated that a strong foam could be formed in situ at reservoir conditions and that the diversion of CO_2 to previously bypassed zones/areas due to foam resulted in increased oil production and dramatically decreased CO_2 production.

As part of the CO_2 project, the multi-component pseudo-miscible simulator MASTER (Miscible Applied Simulation Techniques for Energy Recovery), which was developed by the U.S. Department of Energy, was modified by incorporating a foam model and used to conduct a history match study on the pilot area at EVGSAU to understand the process mechanisms and sensitive parameters[1]. The ultimate purpose was to establish a foam predictive model for CO_2-foam processes[3,4].

Expert and fuzzy control: Our initial study of the problem was done by a 200 case parameter value study so as to tell the effects of differing parameters. This was extremely complex in deciphering the results since increasing the permeability in one location may have the opposite effect as an increase in another location when both production and injection wells (and their placement times) must be taken into account. To deal with the results and collate the data an initial simple expert system proved to be ideal in that the rapid prototyping and quick modification ability allowed a controller to be built. This initial expert system also proved to be extremely beneficial as a basis for the later fuzzy and genetic algorithm controllers.

Expert system: Our initial studies began with the creation of a simple Expert System (ES) based controller, which would later form the basis of several other controllers. The rules of the ES controller were formulated empirically from the initial parameter study. The ES was composed of 25 IF-THEN rule groups; one rule per well. These rules used a combination of actual well error values (current parameter set history vs. case) and predicted well error values. Figure 2 For ease of use the error values were divided into one of nine ranges that described the degree of error within the match and type: below or above the actual history. Each rule set was rune in sequence and the resulting predicted set passed to the next rule to be used as an actual set.

While the ES controller method proved invaluable as it allowed for rapid proto-typing and quick reduction of error in the match, it was not without its faults. The primary problem was the granularity induced by having only 9 error ranges. Standardized parameter alteration values, tended to cause oscillation as error ranges would tend to bounce between two opposing sets (such as H to L and L to H), in later runs Fig. 3. This was primarily due to the fact that wells 8 and 12 tended to work inversely of each other within the primary depletion period this led to oscillation which tended to occur as the match on one well would be improved, the other well's match would worsen. Despite this reductions in error by over 800% by the fourth or fifth iteration of the ES were not uncommon.

If well error value for well 8 is SH Slightly High and well error value for well 12 is SL Slightly Low then decrease parameter 3 by 30. Change predicted set for well 8 and 12 to K.

Fuzzy control: The granularity problem, along with the fact that as we would need to later increase the number of

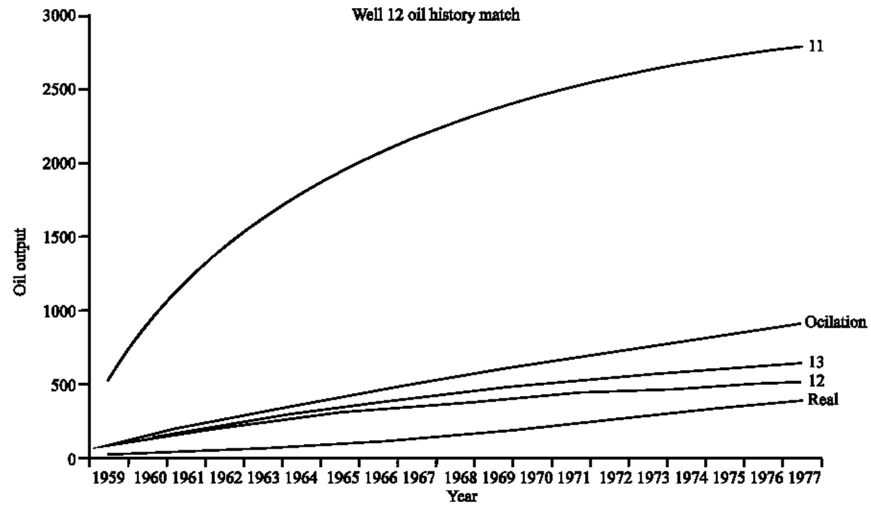

Fig. 2: The above is a partial IF-THEN rule for parameter 3, which is located at well 3 on the well map. The underlined text denotes well error ranges

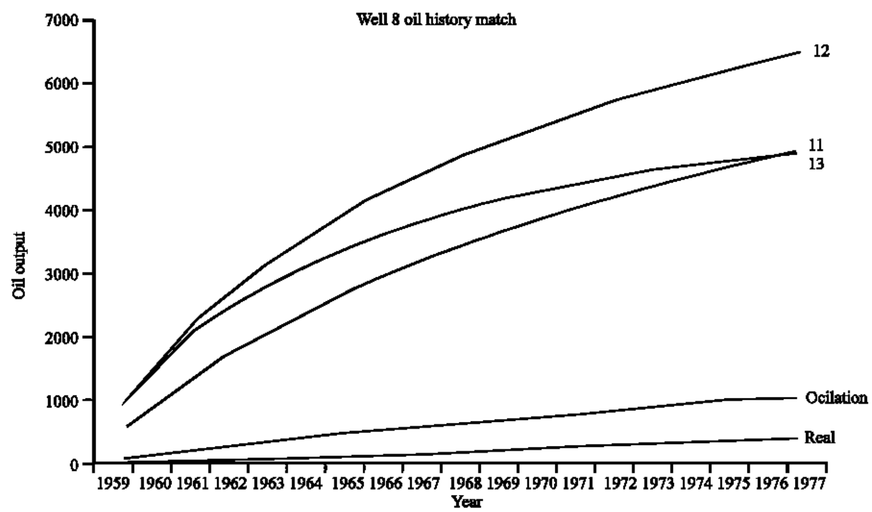

Fig. 3: The above two graphs, are a the result, of a match on the primary depletion period, for wells 8 and 12 (left and right, respectively). As can be seen, initial error of the match decreases rapidly, towards the case being matched (actual historical data in this simulation)

parameters and increasing rule set size and complexity, along with dealing with the secondary depletion period required that we find a better control method than the ES. As our next step we decided on a fuzzy logic based control system. This system would allow us to go from a twenty plus page rule sets, to a few simple tables, several of which could be reduced to a few simple equations.

The Fuzzy Controller (FC) also proved beneficial in that this system could be easily automated as it when through the simulation cycle. It was this controller that truly began dealing with the problem of reducing human intervention.

The purpose of the fuzzy controller was to do the parameter adjustment automatically and eliminate human intervention. The benefits of fuzzy control in this application are the ability to get around the problems of complexity in formulating exact rules and to deal with situations where there are multiple meta rules that may be applicable under similar circumstances. For example, expert opinion about permeability adjustment leads to the development of three different meta-rules:

- If both wells' outputs are too high, then choose those blocks whose reduction in permeability leads to low outputs.

- If wells' outputs are too low, then choose those blocks whose increase in permeability leads to high outputs.
- If one well's output is too high and the other's is too low, then choose those blocks whose alteration in permeability leads to proportional, corrective shifts of outputs.

Rules of the third type were the most difficult to obtain, since many factors need to be considered before a decision was made regarding which blocks' permeability to increase and which blocks to decrease, thus the need for developing the rules empirically. If it was not for the ES that formed a basis for the fuzzy controller and the initial parameter study creating these rules would have been impossible as some of the wells interactions where highly complex.

The fuzzy controller consists of sections:

- Fuzzification Module: Accepts condition/Input and calculated membership grades to express measurement uncertainties.
- Fuzzy Inference Engine: Uses the fuzzified measurements and the rules in the rule base to evaluate the measurements.
- Fuzzy Rule Base: contains the list of fuzzy rules.
- Defuzzification Module: converts the conclusion reached by the inference engine, into a single real number answer

The primary benefits of using fuzzy control is that it is easy to design and tune and it avoids the difficulty of formulating exact rules for control actions. The fuzzy controller's rules are empirically obtained, based on a parameter study in which a single well's permeability value was altered while the rest of the 24 permeability values were held constant. The fuzzy controller implemented for permeability adjustment is of the simplest kind in that percentage errors and control actions are fuzzified, but only rarely will more than one rule fire[5,6]. The control action applied is thus usually only scaled by the membership grade of the percentage error in the error fuzzy set. The adaptive controller works as follows.

- Fuzzification is accomplished by usage of membership functions. After a simulation is run, an error calculation is made from the simulated and the synthetic case or historical data based on a percent error formula. This value is then used to determine error values membership in each fuzzy set: {EL Extremely Low, VL Very Low, L Low, SL Slightly Low, K within tolerance, SH Slightly High, H High, VH Very High, EH Extremely High}. The corresponding fuzzy set values are -4, -3, -2, -1, 0, 1, 2, 3, 4, respectively.
- Inference begins once the membership grades are calculated. It assigns the fuzzy set with the highest membership value for each well. If an equilibrium condition is reached between two sets, the set value closest to K is chosen.
- Rule Firing is our next step. Within the fuzzy rule base there are 3 types of rules: I (increase production rules), D (decrease production rules) and P (shift production from one well to another). Based on the fuzzy set assigned to each well, we can decide the rule type that needs to be applied.. Based on the fuzzy set value assigned to each well, we can calculate the average set distance from K and decide the change degree (firing strength) of a rule that needs to be applied, that needs to be applied.
- The final step is application of the control action. The action taken depends on the chosen rule type and the degree change needed. The parameters for the next simulation run are now altered.

Many experiments have been conducted[7]. The fuzzy controller's performance depends, naturally, on the definition of fuzzy sets for error and the definition of the fuzzy sets for control actions; therefore, the rule base needs to be fine tuned for optimal performance. Since the rules must be based on empirical observations, other factors, such as scaling factors of the controller[7,8], may not be quite as critical. The basic idea of using a fuzzy controller for automatic parameter adjustment in history matching, however, has been validated by using a specific controller with crisp control actions. In this case we were able to obtain very good matches within 5 iterations for the two wells over their primary production period of 18 years. Previously, with manual adjustment, such close matches would easily take several weeks to a few months to achieve.

Genetic algorithms: Initial Genetic Algorithm (GA) trials were run using differing crossover methods. These studies proved interesting in that little information was needed in creating the GA system, but at the same time proved to have a huge drawback. As simulation times could range up to 45 min on even a 600MHz, creating initial populations and simulating future generations, became extremely costly providing a large negative to this method. As a result in our study smaller populations for initial testing were used, thus limiting the GA, as large degrees of similarity occurred between population members in succeeding generations.

This method also proved to be interesting as an initial study would not require large amounts of time to study

Rule 1:	Classifier 1:
Error Environment	Error Calculations:
Match:	Error for well
Error Well X=range	N= Error
Actions:	Parameter List:
Change Parameter N by X	Parameter
Statistics:	N=pN
Age, Uses, Accuracy	

Fig. 4: ARIA example rule and classifier

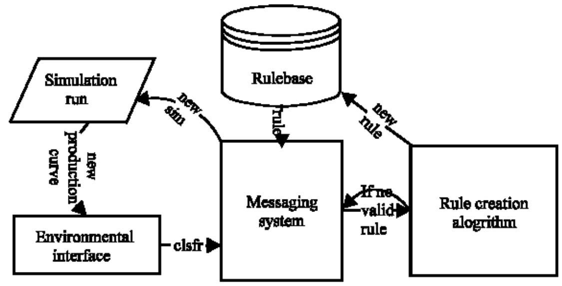

Fig. 5: ARIA pictorial overview

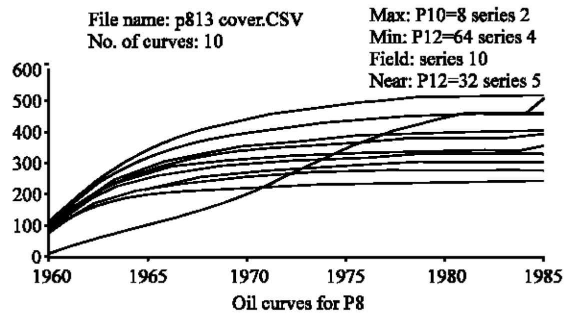

Fig. 6: The Figure demonstrates the bracketing curves for the synthetic case (solid line)

the problem and adapt a GA to solve it. In contrast, while results were promising, much fine-tuning based on previously acquired knowledge was needed to take this method to the next level.

In doing using multipoint crossover. Population improvement tended toward only a 1-3% change in the initial few generations. In using standard crossover the best results were found using a complete generational replacement scheme with small random number of crossover points for each new child.

Hybrid systems: a classifier system: In dealing with the previous systems one obvious question is how could we get a system to learn its own rules. The following hybrid system is the result.

ARIA (Automatic Recombinant Input Approach) uses a nonstandard Genetic Based Learning Machine (GBML) to create a set of rules that may be used for atom rule creation for control in the history matching.

Examples of ARIA rules and classifier are given in Fig. 4. Each rule consists of a set of atom like subsections. For example in rule 1 below each line is an atom section that we work with:

ARIA consists of rule populations that are tested using actual application data and are tracked based on their effectiveness, in altering a well parameter set. It uses a standard genetic algorithm classifier messaging system. The system consists of four parts:

- Part one is the error reception part (environmental interface) in which a parameter set to be adjusted is received.
- Part two is the rule base of previously composed rules.
- Part three is a genetic algorithm with fuzzy control section that creates new rules when a rule is not available from the database.
- The final part is the messaging system that tracks a rule's effectiveness, matches a rule to a parameter set and takes relevant action.

The first part of the ARIA process just accepts parameter simulation data and creates error estimates from the actual historical production data creating the first part of our environmental awareness. This error calculation is done by using a Sum of Squared Errors calculation between the predicted output of each well and its historical values. Once the SSE has been calculated for each well, these 8 values become the environmental pattern (classifier) which will be used in the messaging system for rule matching. Figure 6 is an example of an environmental classifier. It has two parts the Error calculations and the list of parameters that belonged to the simulation data.

These error values in the classifier are then matched based on a set of fuzzy rules in the messaging system to appropriate action rules. The fuzzy control in this section is very basic and consists of simplistic rules that determine if a rule exists within range, a tolerance factor and rates each rule by its statistical information and then determines which is the most appropriate rule. This is done by attempting to find a rule whose Error Environmental Match ranges for each well bracket the classifiers error calculations. Since there do exist 8 error calculations the idea of a tolerance factor was introduced, in which not all 8 error calculations must be in range. This calculation is done by using an averaging function to calculate how out of range the error set is. If an adequate

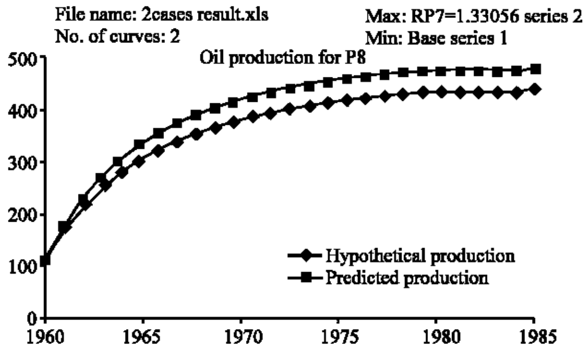

Fig. 7: Synthetic case match using the neural network

rule is found, it is then used and statistical success or failure data of its use on the simulation parameters is average together. On the other hand, if an appropriate rule cannot be located the ARIA system invokes the genetic fuzzy logic creation algorithm to create a new rule, which is then used.

This method has shown some promise in application and is merely an extension off of the previous work for the MASTER WEB project in which fuzzy control was applied resulting in convergence and error control within ten generation and 200% error ranges, proving to be a very quick and accurate system. Currently the system has been running small numbers of iterations, as tests are being run to determine the best initial rule population. Currently small numbers of changes, that rely on being able to affect parameters within the lower third of their value ranges, without causing parameters to go out side of their allowed values, have shown the most successful ability to converge to a solution. They have been able to come within a 30 to 70% error within approximately 15 iterations.

The size of the rule base has also been shown to have a significant effect on the number of iterations, as the larger the size the more likely an appropriate rule will be found. Created rules have are extremely dependent on the genetic algorithm used, as wells have complex interactions.

This method had several benefits. For example, as we started to deal with the secondary depletion period and the increasingly complex well interactions, this system could actually discover them. This provided surprising knowledge and a system capable to adaptation, even if it started with false premises.

Neural networks: Using a neural network for modeling has produced interesting results. In this section we present the results of two different networks. The first section will display results obtaining permeability values using period I data with attempting to history match to a synthetic case. In section 6.2 we present the results of a second network that history matches using an increased number of parameters (i.e. an addition 7 relative permeability parameters in addition to the 25 permeability parameters).

Synthetic case: The neural net method we chosen for the problem takes a group of predicted oil production output as training data. The data was acquired by running the MASTER simulator with a set of ranged values so as to cover the hypothetical case's production output curve. This resulted in a set of curves, which bracketed the synthetic case in the first neural network and the actual history in the second. It was necessary for the training data to cover the synthetic case history so to restrict the range of the problems. The figure below shows a very small set of cases that cover the history. The solid line in is the synthetic case history.

Once the network is well trained, we feed the network with the historical data to get 25 permeability parameters and 7 relative permeability parameters. We then feed these parameters into the MASTER simulator to check if these parameters are acceptable and to create an estimate of the errors for these parameters.

The network we built for this study is a three-layer feed forward network, with 27 input units (historical data), 30 hidden units and 32 outputs (permeabilities). The scaled conjugate gradient descent algorithm is used in training. Figure 5 shows the comparisons between the desired output and the output from a trained network. We can see good matches between predicted value and desired value except for one pair. This mismatch can likely be attributed to the fact that certain permeability values have little effect on the output history. (For example during the first 20 years the 25th permeability value causes less than a 1% change across its complete value range.) Furthermore, Figure 7 above shows an experimental result in using the neural network to match the chosen hypothetical case, displaying a very close match. Currently the training data and testing data are all simulation results from MASTER simulator. The next section will demonstrate results obtained using real historical data with an increased number of parameters.

Extended synthetic case: The second network is a three-layer feed forward network, with 40 input units (historical data), 20 hidden units and 32 outputs (7 relative permeability parameters in addition to 25 permeability parameters) which uses gradient decent with momentum algorithm for faster training. In this network we use data from periods I-III.

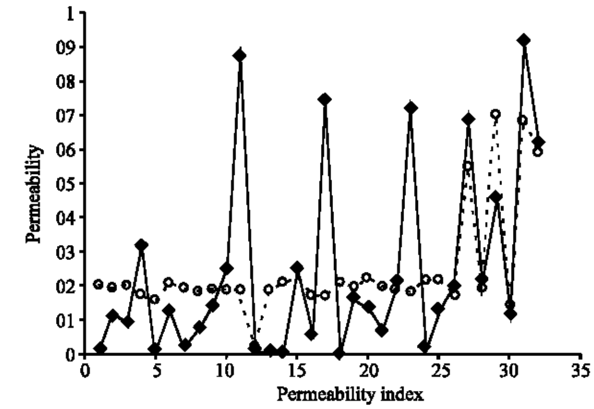

Fig. 8: Neural network results

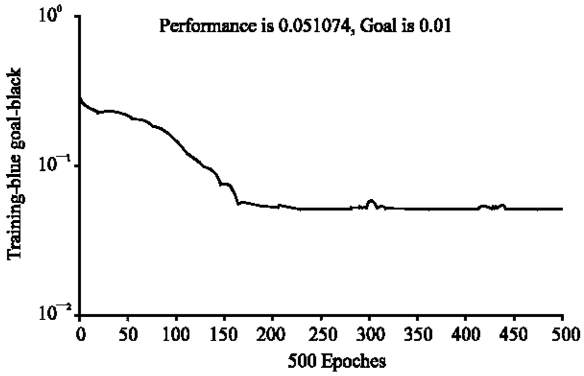

Fig. 9: Network training

Using the Mat lab Neural Network Toolbox, we created a network and trained it on 624 pre-chosen cases using the Gradient descent w/momentum and adaptive linear back propagation method. We ran the training for 500 epochs, although the system tended to stabilize by 300 epochs.

The graph of the progress of the training can be seen here below. After training, we plotted one data set used to train the network against the output of the network. As you can see in the graph in Fig. 8, the network seemed unable to match the permeabilities accurately, but worked rather well on the relative permeabilities. You will note that all permeabilities are between 28 and 60. This is consistent with Dr. Chang's suggestion that 50 are a sort of expected value. Finally, we passed these values to the simulator to see how closely the inputs drive the simulator to the actual production.

As expected, our results were not surprising, using the extended data did not provide us with as perfect of a match as the first network. This can be partially attributed to accumulative data corruption due to such things as well workovers, stock tank oil reserves, etc.

Neural network results: This method has a big advantage to it. Training may be initially costly, but the method has the benefit that once the neural net is trained, the solution can be obtained rapidly-unlike the ES and FC methods which usually take many simulation cycles to minimize error and thus take longer time to find solution.

CONCLUSION

While simulation experts have traditionally done history matching semi-manually, soft computing algorithms offer a great deal of promise for reservoir simulation in general and history matching in particular. While these algorithms have produced great experimental results in history matching, one must realize that there may be high costs within their use and design. For example the neural network may be costly (primarily in simulation time to create the training set), though actual run time after it has been trained is quick; while the expert and fuzzy controllers run several simulation iterations minimizing error each time.

Preliminary results of applying the soft computing algorithms with the MASTER simulator on the EVGSAU reservoir have shown good matches within hours. These results would have taken weeks to achieve using conventional methods. The algorithms' applicability is also sufficiently general, clearly demonstrating the potential of this approach.

ACKNOWLEDGEMENTS

We would like to gratefully acknowledge that support for this research was received from Sandia National Laboratories and the State of New Mexico. Dr. Eric Chang (previously with the Computer Science Department of New Mexico Tech) and Dr. Reid Grigg (of the Petroleum Recovery Research Center of New Mexico Tech) initiated our interest in this project and made invaluable contributions during its development.

REFERENCES

1. Ammer, J.R., A.C. Brummert and W.N. Sams, 1991. Miscible Applied Simulation Techniques for Energy Recovery-Version 2.0. Report to U.S. Department of Energy, Contract No. DOE/BC–91/2/SP.
2. Aziz, K. and A. Settari, 1979. Petroleum Reservoir Simulation. London: Applied Science.
3. Chang, S.H. and R.B. Grigg, 1998. History Matching and Modeling the CO2-Foam Pilot Test at EVGSAU. Paper SPE 39793 presented at the 1998 SPE Permian Basin Oil and Gas Recovery Conference, Midland, Texas.
4. Martin, F.D., J.E. Stevens and K.J. Harpole, 1995. CO2-Foam Field Test at the East Vacuum Grayburg/San Andres Unit. Spere, pp: 266.
5. Klir, G.J. and B. Yuan, 1995. Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall.

6. Jang, J.S.R., C.T. Sun and E. Mizutani, 1997. Neural-Fuzzy and Soft Computing, Prentice-Hall
7. Janoski, G., 1999. MASTER Web Reservoir Simulation. Technical Report, Department of Computer Science, New Mexico Tech.
8. Palm, R., 1995. Scaling of Fuzzy Controllers Using the Cross-Correlation. IEEE Tran. Fuzzy Systems, 3: 116-123.