Stanford University

ENERGY 291: Optimization of Energy Systems

Project – Final Report (P5)

Multilateral Smart Well Performance Surrogate-Based Optimization

Mohammad (Jabs) Aljubran
Anthony Boukarim

03/19/2019

Introduction:

Recent technological innovation in drilling and completion allowed for achieving complex multilateral wells with significantly higher recovery rates at marginal additional costs. These advanced installations resulted in improved reservoir contact, ultimate recovery, zonal isolation and pressure drawdown control, and management of water/gas breakthrough. Equipped with state-of-the-art instrumentation and control systems, smart wells provide continuous flowrate and pressure data streams in space and time. One major component of these completions is the inflow control valve (ICV) which regulates the flow of reservoir fluids from different branches.

Numerical reservoir simulations represent the standard approach to analyzing oil and gas fields and deciding on the optimal ICV settings. However, using a solver, e.g. genetic algorithm (GA), to optimize over numerical simulators involves many computationally expensive function evaluations. Hence, surrogate-based optimization (SBO), seen in **Fig. 1**, is commonly used to efficiently find the optimal decision variables. Yet, an accurate, fast, and computationally inexpensive surrogate model is critical to SBO. Based on a few function evaluation sample points, proxy formulation techniques, i.e. Kriging, machine learning, etc., have to accurately map the objective function across the problem domain for the optimizer to locate the global optimum.

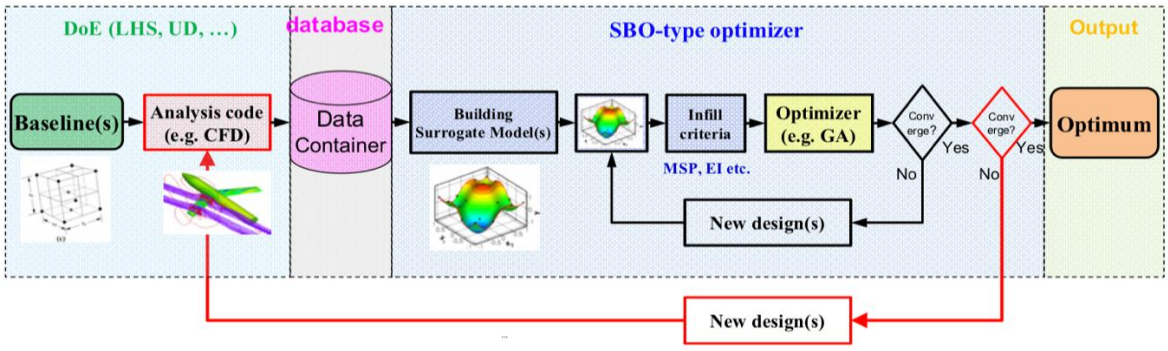

Fig. 1—SBO Approach: SBO uses a space-filling sampling design initially at which the reservoir model is evaluated to construct the initial database. A surface response is then constructed using a surrogate modeling technique which is used for optimization. Additional samples are selected adaptively to evaluate the model and add to the database for a new optimization iteration to be conducted. This process is repeated until convergence is achieved based on a set of selected criteria (Han 2016).

Data & Method:

Note that the 'Data' and 'Methods' sections are combined as they relate to critical and inseparable aspects of the overall model development.

This project will focus on handling the complex task of optimizing ICV configurations to maximize profit. It will replicate part of the SBO approach where 1) initial and space-filling Latin Hypercube Sampling (LHS) is conducted, 2) Sample points are evaluated using a numerical reservoir simulator, 3) Artificial neural network (ANN) surrogate model is trained, and 4) Nonlinear solver is run on the ANN proxy to locate the global optimum.

Eclipse, a commercial reservoir simulator, is used to construct a reservoir model, seen in **Table 1**. This model, seen in **Figs. 2 and 3**, simulates a homogenous oil reservoir with two-phase flow (oil and water), mainly driven by an underlying water aquifer. A segmented, trilateral producer well is

drilled at the grid center and completed across the reservoir top layer with three ICV devices (each holds 11 discrete settings from 0 to 10 indicating constriction area range of 0 to 0.001 ft^2) which are installed at each lateral tie-in segment. Each simulation run outputs wellhead flow rates, and downhole flow rates and pressure drops at each ICV over a period of 2,000 days.

Table 1—Eclipse Model Input: Rock, fluid, and well input parameters that are used to generate the physical reservoir model for the optimization problem.

Reservoir Size (ft^3)	2000 x 900 x 270	Payzone Thickness (ft)	150
Porosity	10%	Horizontal Perm (mD)	10
Vertical Perm (mD)	0.1	Compressibility (psi^{-1})	3E-6 @ 14.7 psi
OWC (ft)	7150	Reservoir Pressure (psi)	6000
Oil Density (pcf)	40	Water Density (pcf)	62
Water Aquifer	Carter-Tracy	Aquifer Thickness (ft)	120
Segment Count	16	Branch Count	3
Tubing ID (ft)	36%	Tubing Roughness (ft)	0.001
Max Oil Rate (STB/D)	5000	Target BHP (psi)	2000 psi
ICV Setting "0" Area (ft^2)	0	ICV Setting "10" Area (ft^2)	0.001

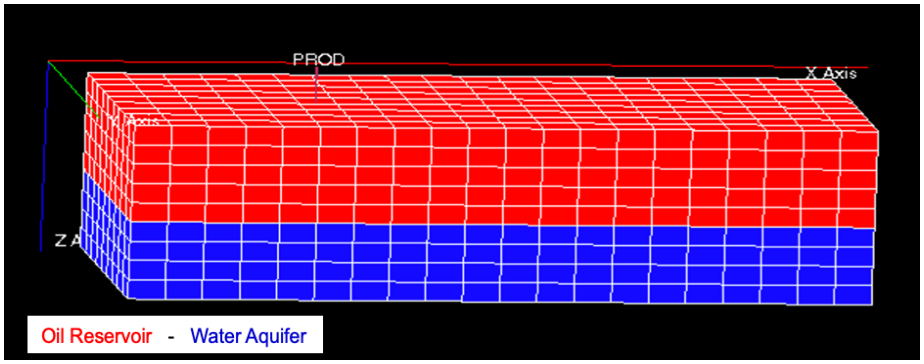

Fig. 2—Eclipse Model: Cartesian Eclipse reservoir model at the initial state where the oil reservoir (red) is 100% saturated with oil and occupies the upper layers while the water aquifer (blue) is 100% saturated with water and occupies the lower layers.

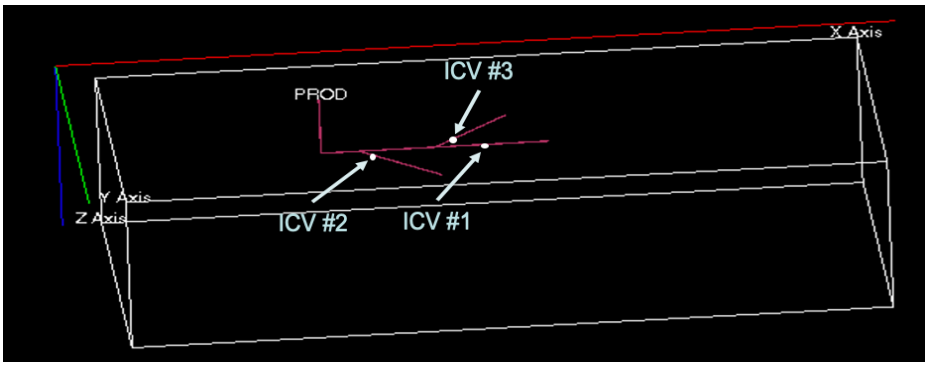

Fig. 3—Trilateral Producer: Cartesian reservoir model with a single trilateral producer situated around the center block. Each lateral is equipped with an ICV device around the tie-in point to control reservoir fluids inflow into the wellbore.

A total of 1,330 simulation runs are conducted with an eight-day timestep, generating 667,660 temporal sample points. **Table 2** shows a simplified example where the oil and water cumulative flow rates of cases 1,2 and 3 are generated using Eclipse which then are used to train an ANN model. The latter acts as a surrogate to predict the cumulative production at the cases 5,6, and 7 where the ICV settings are different and originally unseen by the ANN.

Table 2—Problem Example: The ANN approach is responsible for generating the oil and water cumulative production profiles for unseen ICV settings when given a few others.

Variable	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
ICV #1	10	8	1	0	3	7
ICV #2	10	10	3	6	4	5
ICV #3	10	0	2	2	5	7
Oil and Water Cumulative Production	Given Data			Predicted Output		

Only 50 simulation runs will be used to fit the surrogate in the beginning. Based on the fit and optimization results, this number can either be deemed sufficient or rather increased. These 50 simulation runs are selected using LHS, seen in **Fig. 4**. LHS is commonly used in Monte Carlo simulations to reduce the variance given a fixed number of sample points from a multidimensional domain (Zolan and Hasenbeein 2017). Compared to random sampling, LHS provides a more representative space sample, allowing for training a more robust ANN fit with a smaller set of simulation runs, which is aligned with the SBO objective in locating the optimal solution with minimal computational cost.

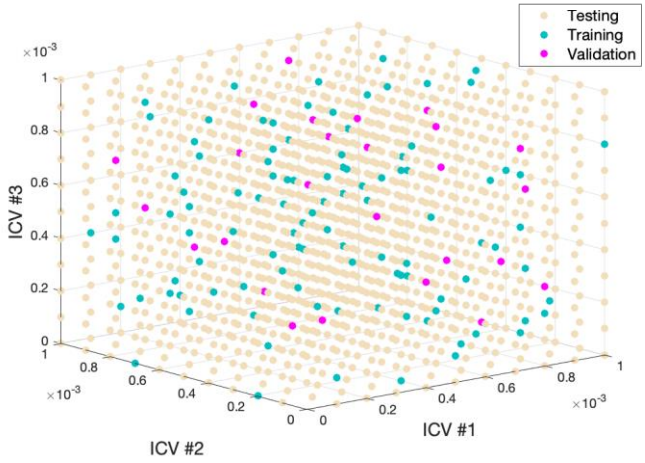

Fig. 4—LHS Approach: LHS is superior to random sampling in generating the initial sample that is used to train and validate the ANN, and allows for fitting a more representative proxy with a smaller number of numerical simulation runs (function evaluations).

A simple one-layer, ten-neuron regularized ANN, seen in **Fig. 5**, was trained on the selected 50 simulation runs. The ANN training process followed the MATLAB implementation of the Bayesian Regularization Backpropagation which takes place within the Levenberg-Marquardt algorithm. MacKay (1992), and Foresee and Hagan (1997) provide details on this ANN training approach. Note that each training sample point represents the ICV settings (input variables) and cumulative oil and water production after 2,000 days (output variables).

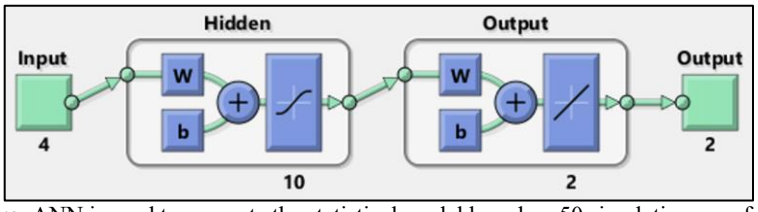

Fig. 5—ANN architecture: ANN is used to generate the statistical model based on 50 simulation runs from the Eclipse numerical model. Note that there are four inputs (Time and settings of ICV #1,2 and 3) and two outputs: cumulative oil and water production at the end of the chosen 2,000-day simulation period. Note that 'W' and 'b' represent the weights and bias parameters while each neuron uses sigmoid as the transfer function.

ANN training results showed relatively small mean-square error when tested on the unseen 1,280 runs which confirms acceptable model generalization properties. Hence, such a simple ANN surrogate model is desirable for SBO purposes as it fairly captures the desired output with much lower computational cost. Note that increasing the simulation sample input and further complicating the ANN architecture is possible, yet the current model will be examined at first to test the SBO approach effectiveness when using a less-than-perfect proxy.

The objective will be to maximize profit, which is based on the ANN output of cumulative oil $g(x)$ and water production $h(x)$ after 2,000 days. Meanwhile, the decision variables are the three ICV discrete settings (x_1, x_2, x_3). The objective function model assumes marginal oil price (p_o) of \$50/STB and waste water management cost (c_w) of \$20/STB. The upper limit of ICV settings and their integer nature are the only problem constraints. Since the objective function includes both $g(x)$ and $h(x)$ which are inherently nonlinear, the optimization problem at hand is classified as mixed integer nonlinear programming (MINLP) category. The following mathematical formulation summarizes the model:

$$\max_x \; f(x) = p_o\, g(x) - c_w\, h(x) \quad s.t. \begin{cases} x_1, x_2, x_3 \leq 10 \\ x_1, x_2, x_3 > 0 \\ integers \end{cases}$$

Due to the nonlinearity of the problem, a genetic algorithm will be used to facilitate global search over the multidimensional domain of the problem. GA's represent a branch of a broader field of study called evolutionary computation, in that they imitate the biological processes of reproduction and natural selection to solve for the 'fittest' solutions. Similar to evolution, many of the GA processes are random, yet GA is flexible and allows for various levels of control over randomization. These algorithms are far more powerful and efficient than random search and exhaustive search algorithms, yet they do not require further details about the given problem. This feature allows them to find solutions to problems that other optimization methods cannot handle due to the lack of continuity, derivatives, linearity, or other features. (Rao 2009)

As seen in **Fig. 6**, the algorithm initiates a random assortment of chromosomes (initial candidate solution), which serves as the initial population. Then each chromosome in the population is evaluated by the fitness function. Next, the selection operator chooses some of the chromosomes for reproduction based on a probability distribution defined by the fitness of the solutions. The fitter a chromosome is, the more likely it is to be selected. Once grouped in pairs, a crossover operator is used to swap a subsequence of these chromosomes to create two new offspring from each couple. As a final step, a mutation operator is applied (with a very low probability) to randomly flip individual bits in the new chromosomes. The second generation is then tested by the fitness function, and the cycle is repeated until any of the selected stopping criteria is met.

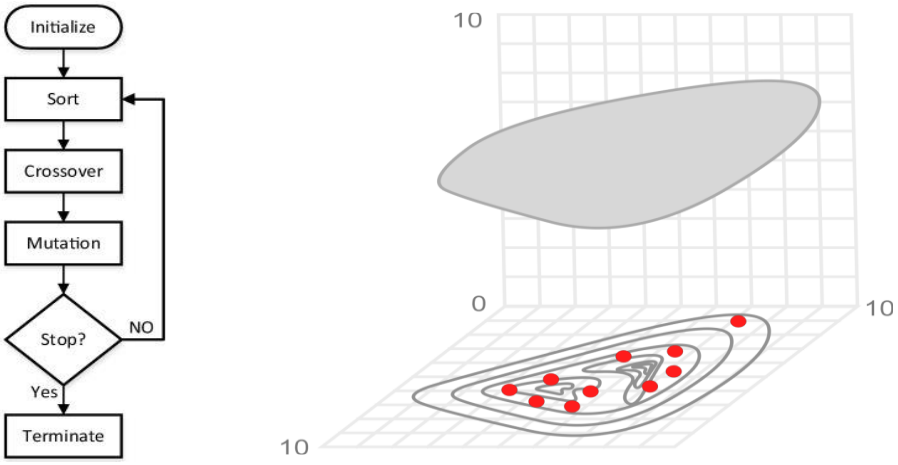

Fig. 6— GA: The flow chart (left) represents the standard GA while the hyperplane drawing (right) illustrates how an initial population design spans most of the domain.

At first glance, the mutation operator may seem unnecessary. In fact, it plays an important role, even if it is secondary to those of selection and crossover. Selection and crossover maintain the genetic information of fitter chromosomes, but these chromosomes are only fitter relative to the current generation. This can cause the algorithm to converge too quickly and lose "potentially useful genetic material" (1's or 0's at particular locations). In other words, the algorithm can get stuck at a local optimum before finding the global optimum. The mutation operator helps protect against this problem by maintaining diversity in the population, but it can also make the algorithm converge more slowly (Rao 2009). The MATLAB implementation of GA algorithm is used to solve the current MINLP problem with the hyperparameters shown in **Table 3**.

Table 3—GA Parameters: Set of hyperparameters that are chosen to run GA. Different choices are always possible and it is important to tune these parameters to achieve consistent and accurate optimal solutions.

Parameters	**Value**
Size of initial population	$m = 50$
Probability of crossover	$p_c = \frac{F_i}{\sum_{j=1}^{m} Fj}$
Crossover fraction *(fraction of pop. for next generation)*	$f_c = 0.8$
Probability of mutation	$p_m = 0.01$
Termination criteria *(change in objective function)*	$\Delta f \leq 10^{-6}$

Results and Discussion:

Note that the 'Results' and 'Discussion' sections are combined as they relate to critical and inseparable aspects of the overall model development.

After tuning the GA hyperparameters and optimizing over the ANN proxy, the solver converged to the optimal ICV setting of $x^* = [10\ 2\ 2]$ with optimal profit of $f(x^*) = \$\ 91{,}265{,}846$ at the end of a 2,000-day production period. **Fig. 7** shows that the GA optimization session converged and stopped after 82 iterations. Meanwhile, the optimal solution is located around the 32nd generation. We can clearly see that the mean penalty value (of the chromosomes at each generation) is converging towards the optimal solution as the number of iterations increases. The best solution was updated 7 times during the optimization process.

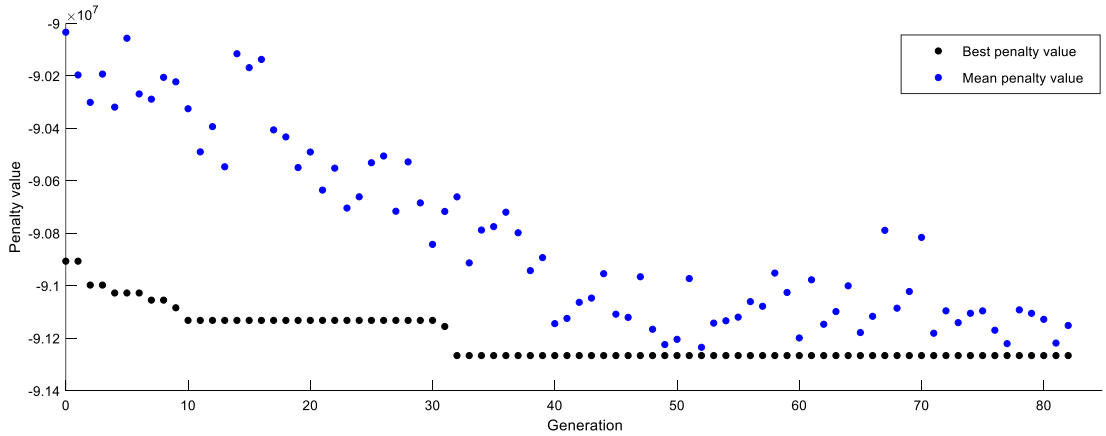

Fig. 7—Results from GA Optimization: The best solution was reached after 32 iterations. The algorithm kept searching for better solutions up to 82 iterations where the average change in the penalty fitness became less than 10^{-6}

To evaluate the current optimal solution, the actual optimum is enumerated using the 1,330 Eclipse simulation runs where all possible ICV configurations are examined. Enumeration resulted in an optimal ICV configuration of $x^* = [10\ 2\ 1]$ with total profit of $f(x^*) = \$\ 90{,}898{,}380$. There is a slight mismatch between the statistical proxy optimization result versus the enumerated physics-based Eclipse optimum where ICV #3 setting is slightly off. This observation is either due to the lack of accuracy in the proxy, or that the GA solver could not locate the global optimum solution within the proxy. To address the latter, three surface plots of the objective function, seen in **Fig. 8**, are generated by fixing a single decision variable at a time which allows to visualize the objective function surface under that constraint. Note that each variable is fixed to its corresponding optimal value which was found based on the proxy optimization process.

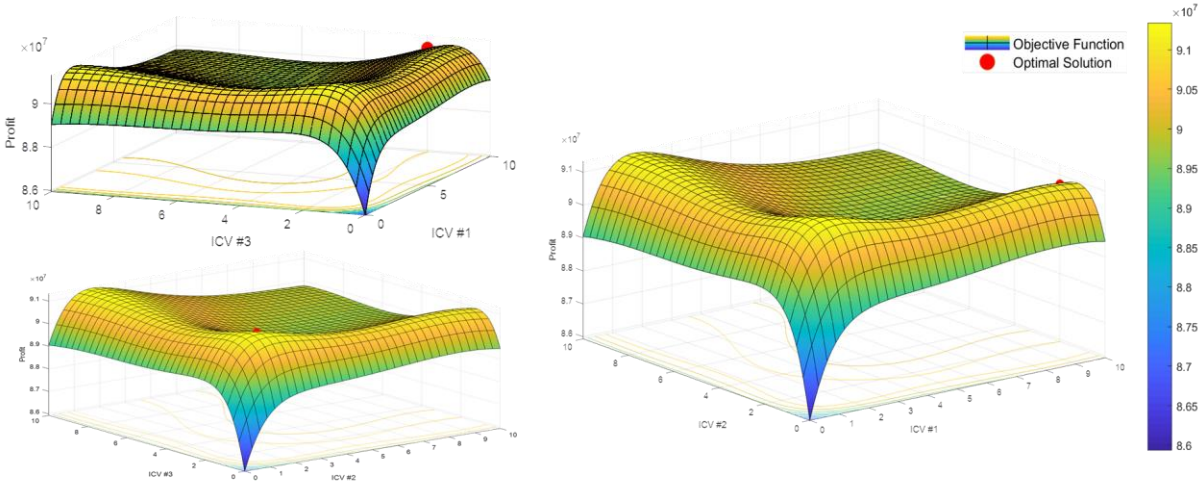

Fig. 8—Graphical Representation of the Objective Function: The surface plots are obtained by fixing one variable and plotting the objective surface with respect to the other two.

Fig. 8 confirms that the GA solution indeed solved for the global profit optimum of the proxy (since the decision variables are integers, the optimal solution will not be exactly at the tip of the hills). Hence, the aforementioned mismatch is likely due to the deviation of proxy model from the true numerical model. This aspect will be investigated in the following section.

Sensitivity Analysis:

Proxy Factors:

SBO accuracy is dependent on the proxy estimation technique. The adaptive sampling account for the imperfect results of the surrogate model at any iteration by sampling further to ensure a representative replica to the numerical simulator. This process is simulated here at a small scale where the sample size is increased from 50 to 200 simulation runs and the ANN optimal architecture is set to have two layers with ten neurons each. **Fig. 9** compares the validation regression fit of the original proxy (Case #1) and the new proxy (Case #2). Note that the latter is superior with a more accurate match especially at low oil and water production rates.

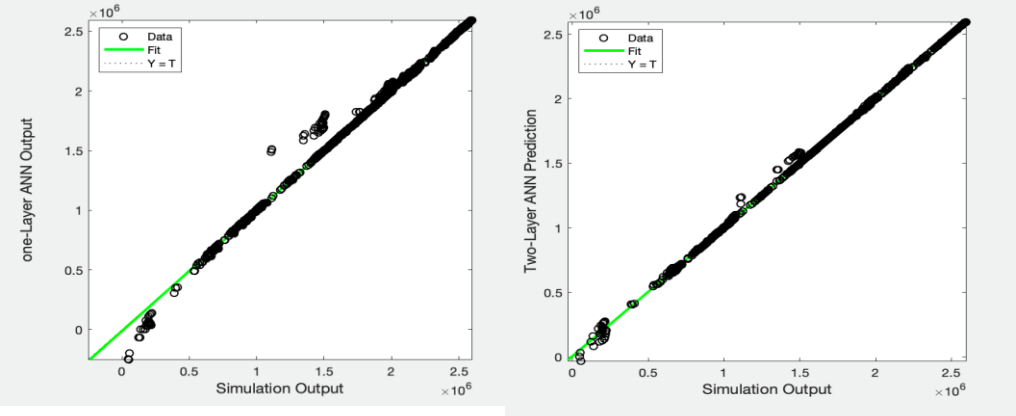

Fig. 9— ANN Regression: Testing the ANN surrogate predictions of the one-layer ANN Case #1 (left) and two-layer ANN Case #2 (right) against the unseen testing simulation runs. Note that Case #2 yields a more accurate proxy to the numerical model, especially at low production rates where the Case #1 proxy performs poorly.

The optimization problem is solved again with GA over Case #2 proxy and compared to Case #1 and the Eclipse optimal solutions. **Table 4** shows that both Case #1 and Case #2 yield close results given the aforementioned oil price and water cost. Yet, Case #2 proxy captures the actual model more accurately and replicates the actual optimal solution results. Note the accuracy-versus-time tradeoff amongst the three approaches.

Table 4— Optimal Solution Comparison: Optimal decision variable and profit results of the two proxy models are compared to the actual optimal solution enumerated using Eclipse. Note that the more accurate ANN proxy (Case #2) yields superior results that closely replicate what Eclipse offers while saving computational expense.

Simulation Model	Optimal Solution	Optimal Objective Value	Total Optimization Time
Case #1	$x^* = [10\ 2\ 2]$	$f(x^*) = \$\ 91{,}265{,}846$	≈ 1.3 hours
Case #2	$x^* = [10\ 2\ 1]$	$f(x^*) = \$\ 90{,}902{,}669$	≈ 2.0 hours
Eclipse	$x^* = [10\ 2\ 1]$	$f(x^*) = \$\ 90{,}898{,}380$	≈ 16.7 hours

Total optimization time, seen in Table 4, is roughly computed for each approach. It is observed that all converge after 80±5 GA generations; therefore, 80 will be considered as the average number of generations where each generation performs 50 function evaluations. Each function evaluation using Eclipse requires about 15 seconds while the ANN proxies take less than 1 second per function evaluation since they are simple. Furthermore, training the ANN proxies of Case #1 and Case #2 on MATLAB take about 30 and 45 seconds, respectively. The total optimization time with SBO over ANN proxy of Case #1 and Case #2 includes 1) Simulating the corresponding samples using Eclipse, 2) training the networks, and 3) Optimizing with the GA solver. As seen in Table 4, Case #2 arrives at the exact optimal solution while saving 88% of optimization time compared to optimizing over Eclipse.

Financial Factors:

Fossil energy market supply and demand are in constant fluctuation which directly impacts oil prices. Technological advancements and logistical challenges also affect the marginal cost associated with upstream, midstream, and downstream oil and gas activities. Meanwhile, policy changes and environmental regulations affect the cost of waste water management. Hence, it is important to study the sensitivity of the surrogate-based optimization to these changes.

Oil price and water cost ranges of 30 – 120 $/stb and 10 – 50 $/stb, respectively, were inspected. The GA solver was used along to optimize the Case #2 proxy objective function over the different combinations, seen in **Fig. 10**. Note that increasing the oil price to relatively high values while decreasing waste water management cost to relatively low values result in higher profit where the algorithm decides to set all ICV's fully open, seen in **Table 5.** Meanwhile, the optimizer chooses to shut off two ICV's while keeping ICV#2 at setting "4" when the water management cost is relatively high compared to the oil price.

Table 5— Optimal Solutions: Optimal decision variable combinations for the extreme combinations (around the Fig. 10 surface plot corners) of oil price and water cost combinations.

p_o ($/stb)	c_w ($/stb)	Optimal Solution
120	10	$x^* = [10\ 10\ 10]$
120	50	$x^* = [10\ 2\ 1]$
30	10	$x^* = [10\ 2\ 10]$
30	50	$x^* = [0\ 4\ 0]$

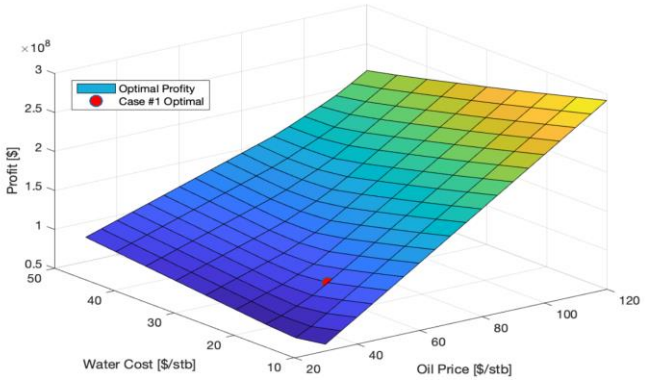

Fig. 10— Financial Sensitivity: Optimal solution sensitivity to oil price and water cost is examined by solving for the optimal solution over Case #2 using GA on different oil price and waste water management cost combinations.

Challenges and Future Work:

Several challenges were encountered throughout optimizing the problem at hand, including multi-dimensional MINLP and proxy reliability. The problem involves three decision variables where visualizing the objective function and constrains is not an option. In addition, the nonlinear nature of the oil and water production functions requires a global-search solver, i.e. GA. Meanwhile, constructing a representative proxy is not a trivial task and requires the selection of an appropriate fitting technique to closely replicate the physics-based numerical model while providing computationally inexpensive evaluations to the objective function.

The current problem can be improved to further replicate reality through incorporating radial grid, rock heterogeneity, multiphase reservoirs, more producers and ICV's, water and gas injection wells, etc. The model can also be enhanced into a multi-objective problem where it minimizes the environmental impact of greenhouse gas emissions and water production. In addition, the model can be fully integrated with adaptive sampling techniques and appropriate convergence criteria to fully represent an SBO loop.

References:

[1] *Abukhamsin, A. 2016. Inflow Profiling and Production Optimization in Smart Wells Using Distributed Acoustic and Temperature Measurements. Stanford University PhD Dissertation.*

[2] *Al-Anazi, A., Isichei, O., Al-Yaha, M., et al. 2017. Innovative Production Optimization Technique for Smart Well Completions Using Real-Time Nodal Analysis Applications. SPE Symposium: Production Enhancement and Cost Optimization.*

[3] *Al-Ghareeb, Z. 2009. Monitoring and Control of Smart Wells. Stanford University MS Thesis.*

[4] *Foresee, F., and Martin T. 1997. Gauss-Newton approximation to Bayesian learning. Proceedings of the International Joint Conference on Neural Networks.*

[5] *Han, Z. 2016. A Generic Surrogate-Based Optimization Code for Aerodynamic and Multidisciplinary Design. 30th Congress of the International Council of the Aeronautical Sciences.*

[6] *MacKay, D. 1992. Bayesian interpolation. Neural computation. Vol. 4, No. 3.*

[7] *Rao, S. 2009. Engineering Optimization: Theory and Practice, 4th ed.*

[8] *Sheridan, R., and Kearsley, S. 1995. Using a Genetic Algorithm to Suggest Combinatorial Libraries. J. Chem. Inf. Comput. Sci, Vol 35, No. 2.*

[9] *Zolan, A., Hasenbein, J., and Morton, D. 2018. Optimizing the Design of a Latin Hypercube Sampling Estimator. Winter Simulation Conference.*