SPE 136944

Optimization of Multilateral Well Design and Location in a Real Field Using a Continuous Genetic Algorithm

Ahmed Y. Bukhamsin, SPE, Saudi Aramco/Stanford University; Mohammad Moravvej Farshi, SPE; Khalid Aziz, SPE, Stanford University

This paper was prepared for presentation at the 2010 SPE/DGS Annual Technical Symposium and Exhibition held in Al-Khobar, Saudi Arabia, 04–07 April 2010.

Abstract

As many fields around the world are reaching maturity, the need to develop new tools that allow reservoir engineers to optimize reservoir performance is becoming more urgent. One of the more challenging and important problems along these lines is the well placement optimization problem. In this problem, there are many variables to consider: geological variables like reservoir architecture, permeability and porosity distributions, and fluid contacts; production variables, such as well placement, well number, well type, and production rate; and economic variables like fluid prices and drilling costs. Furthermore, availability of complex well types, such as multilateral wells (MLWs) and maximum reservoir contact (MRC) wells, aggravate this challenge. All these variables, together with reservoir geological uncertainty, make the determination of an optimum development plan for a given field difficult.

The objective of this work was to employ an optimization technique that can efficiently address the aforementioned challenges. Based on the success and versatility of Genetic Algorithms (GAs) in problems of high complexity with high dimensionality and nonlinearity, it is used here as the main optimization engine. Both binary GA (bGA) and continuous GA (cGA) were tested in the optimization of well location and design in terms of well type, number of laterals, and well and lateral trajectories in a channelized synthetic model. Both GA variants showed significant improvement over initial solutions but comparisons between the two types showed that the cGA was more robust for the problem under consideration. The cGA was, thereafter, applied to a real field located in the Middle East to investigate its robustness in optimizing well location and design in more complex reservoir models. The model is an upscaled version for an offshore carbonate reservoir, which is mildly heterogeneous with low and high permeability areas scattered over the field.

After choosing the optimization technique to achieve our objective, considerable work was performed to study the sensitivity of the different algorithm parameters on converged solutions. Then, multiple optimization runs were performed to obtain a sound development plan for this field. An attempt was made to quantify how solutions were affected by some of the assumptions and preconditioning steps taken during optimization. Finally, an optimization run was performed on the fine model using optimized solutions from the coarse model.

Results showed that the optimum well configuration for the reservoir model at hand can contain five or more laterals; which shows potential for drilling MRC wells. Other studies comparing results from the fine and coarse reservoir models revealed that the best solutions are different between the two models. In general, solutions from different runs had different well designs due to the stochastic nature of the algorithm but some guidance about preferred well locations could be obtained through this process.

Introduction

During the last two decades, horizontal and highly deviated wells have become common in oil field development projects. More recently, technological advancements have facilitated drilling of more complicated nonconventional well trajectories, which come in a variety of forms, such as multilateral wells (MLWs) and maximum reservoir contacts (MRCs). Several studies have shown that the performance of nonconventional wells is superior in many areas compared to conventional wells. These advantages include reducing unit development cost, increasing productivity index, extending reservoir contact length and drainage area, increasing net worth of the drilling investment, reducing operational drawdown pressure, and delaying gas and/or water breakthrough (Horn et al., 1997; Taylor and Russel, 1997; Saleri et al., 2003).

The development of nonconventional wells poses several challenges. The real oil fields are complex environments due to heterogeneities, presence of geologic discontinuities (e.g., faults, fractures, and very high and low permeability zones), and geologic uncertainties. Moreover, given the fact that MLWs require more initial cost than conventional wells, the incremental value of the former might not be realized unless they are optimally placed within the reservoir. Engineering intuition is generally not sufficient to guarantee the optimum placement of these wells due to geological complexity and the nonlinear nature of the problem. Similarly, the usual industry practice of trial and error to test multiple scenarios would rarely succeed in providing optimum solutions in the multidimensional well placement optimization problem. As a result, there is need for an optimization engine to evaluate the performance and viability of different well placement scenarios and determine their optimum design.

The main objective of this work is to evaluate an optimization technique that can be employed to identify a sound field development plan for a real field in the Middle East. Optimized parameters include well type (producer or injector), well placement, well and lateral orientation, and number of laterals in each well. Further, we wish to investigate and improve available optimization procedures. For this purpose, a review of the appropriate optimization procedures and a description of the optimization problem of interest are presented next.

Background

Copious and diverse research works related to well placement optimization have been discussed in the literature. While some studies focused on the placement problem, others have explored applying proxies to speed the optimization process. In addition, other studies have tried to assess the impact of uncertainties on the optimum solution. A brief survey of the most relevant studies is presented next.

To begin this survey, we would like to shed some light on general well placement optimization work done with genetic algorithms (GAs), which is the optimization method used in this study. Bittencourt and Horne (1997) developed a hybrid binary genetic algorithm (bGA), where they combined GAs with the polytope method to benefit from the best features of each method. They tried to optimize the placement of vertical or horizontal wells in a real faulted reservoir. The algorithm sought to optimize three parameters for each well: well location, well type (vertical or horizontal), and horizontal well orientation. The study also integrated economic analysis and some practical design considerations in the optimization algorithm.

Montes et al., (2001) optimized the placement of vertical wells using a GA without any hybridization. They tried to discern the effects of internal GA parameters, such as mutation probability, population size, initial seed, and the use of elitism. Their tests were applied on two synthetic rectangular models (a layercake model and a highly heterogeneous one). For the tested cases, they found that the ideal mutation rate should be variable with the generations. Using random seeds for their problem showed little sensitivity while the use of elitism showed significant improvement. Their study of population size suggested that an appropriate size was equal to the number of variables in the problem. When they used very big populations, solution convergence was deterred as more poor quality chromosomes had to be evaluated.

Emeric et al., (2009) implemented an optimization tool based on GA to optimize the number, location, and trajectory of a number of deviated producer and injector wells. They proposed a method to handle unfeasible solutions by creating a reference population consisting only of fully feasible solutions. Any unfeasible solution encountered during optimization was repaired by applying crossover between it and an individual from the reference population until a new feasible solution was obtained. They applied this technique in three full-field reservoir models based on real cases using two different strategies: the first one with the whole initial population defined randomly; and the second one by including an engineer's proposal in the initial population. Better results were observed in the second strategy and solutions were more intuitive for the tested case. They also suggested and tested an alternative optimization approach by only optimizing well type and number based on an engineer's proposal. Although final results were not as good as the full optimization, they concluded that this approach can be used when there are time limitations to perform the full optimization for complex cases.

Nogueira and Schiozer (2009) proposed a methodology to optimize the number and placement of wells in a field through two optimization stages. The procedure started by creating reservoir sub-regions equal to the maximum number of wells. Then, a search for the optimum location of a single well was performed in each sector. The second stage aimed to optimize well quantity through sequential exclusion of wells obtained from the first stage. After a new optimum number of wells is reached, the first stage is performed again until no improvement in the objective function is observed. This strategy showed efficiency when tested on a heterogeneous synthetic model with light oil. They optimized both vertical and horizontal wells in separate studies. They also concluded that, for their reservoir, the proposed modularization of the problem speeds up the optimization process.

Guyaguler and Horne (2001) applied a hybrid optimization algorithm, which also combines the features of bGAs with the polytope method. Furthermore, they utilized several helper functions including Kriging and Artificial Neural Networks (ANNs) that act as proxies for the expensive reservoir simulations to reduce the optimization cost. This study optimized the locations of several vertical injectors for a waterflood project with the net present value (NPV) as the objective function. Guyaguler and Horne concluded that Kriging was a better proxy than neural networks for the problems tested. They also conducted an uncertainty assessment study based on the decision theory framework. An extensive sensitivity study was performed to determine the effect of the GA parameters.

Yeten et al., (2002) applied a bGA to optimize well type, location, and trajectory for nonconventional wells. Along with that, they developed an optimization tool based on a nonlinear conjugate gradient algorithm to optimize smart well controls. Several helper functions were also implemented including ANN and the Hill Climber (HC). In addition, they applied near wellbore upscaling, which approximately accounts for the effects of fine scale heterogeneity by calculating and applying a skin factor for each well segment. The study included fluvial and layered synthetic models, as well as a section model of a large Saudi Arabian field. An experimental design methodology was introduced to quantify the effects of uncertainty during optimization. The study also conducted sensitivity analysis in a manner similar to that of Guyaguler and Horne (2001).

Previous studies have shown the viability of using GAs for field development problems; however, there is an apparent lack of real field applications with complex geologic structures. Furthermore, the complexity of the problem implies that the objective function surface can contain several local optima, so the exploration criterion of the selected method must avoid converging to points with an unsatisfactory objective function. These reasons, along with some other considerations discussed in the following section, favor the employment of stochastic search methods that are suitable for solving complex problems. GAs are one of the most common algorithms that belong to this category and they were chosen to solve this problem because they are versatile, and easy to parallelize and hybridize. In particular, we want to test the performance of GAs for a real field.

The objective of finding optimum well location and design was approached in this work through four main stages. First, the performance of two variants of GA, the bGA and the continuous genetic algorithm (cGA), was compared and a decision was made on the more robust algorithm for this problem. Second, different internal algorithm parameters were tuned such that they consistently provided good results. This stage also included quantifying the contribution of helper tools and hybrid techniques to the search for optimal solutions. Third, the tuned algorithm was applied to a full field reservoir model based on a real case. The final stage involved uncertainty analysis and testing of the effects of some of the assumptions made during optimization.

Optimization Tools

Genetic Algorithms

The GA is a stochastic and heuristic search technique based on the theory of natural evolution and selection. The basic idea revolves around survival of the fittest and solutions are evolved through mating (information exchange) of the best performing solutions. An occasional alternation of fit solutions is allowed to explore other parts of the search space or to avoid entrapment into local optima (Mitchell, 1996). Using GAs for the well placement optimization problem has been found to be ideal due to the following reasons:

- The algorithm can be easily parallelized because each of the individuals can be evaluated separately.
- The search process is geared towards finding the global optimum rather than local optima.
- They perform well in problems where the fitness function is complex, discontinuous, noisy, changes over time, or has many local optima (Holland, 1992).
- The algorithm is capable of manipulating many parameters simultaneously.
- No gradients are required during the optimization process.
- Since the initial population is composed of multiple solutions rather than a single one, we have the opportunity to explore more of the search space at each generation.
- The algorithm can be enhanced and hybridized with other techniques.

The GA starts by generating a number of possible solutions to the problem (individuals). Each individual (chromosome) can be coded by concatenating different solution variables. At each iteration (generation), a selected objective function value (fitness) is evaluated for the collection of individuals (population). Then, good solutions are evolved by keeping a number of the fittest individuals (selection), mating properties of the good solutions from the previous generation (crossover), and randomly altering some properties (mutation). These three mechanisms are the main GA operators. This procedure continues until a predetermined convergence criterion is met. Common convergence criteria include: reaching a specified maximum number of generations (applied to this study), attaining a population of very similar individuals, and observing no improvement in the objective function of the best individual for a number of consecutive generations.

Binary vs. Continuous GAs

Two GA types are utilized in optimization problems: bGA and cGA. In bGAs, the optimization process embodies coding the value of each variable to its corresponding binary value, applying GA operators to the chromosome, obtaining the resulting offsprings, and remapping them into the real space. Alternatively, cGAs use real-valued numbers directly.

The selection procedure is similar in both algorithms. In this study, the fittest member of each generation is carried to the next generation without any alteration (also called elitist selection). A fraction of the current generation is selected as potential parents for the next generation according to a predefined selection fraction. The parent selection criterion for crossover is based on rank weighting, where fitter individuals are more likely to be selected according to a ranking scale (Farshi, 2008).

Crossover has been described as the key element that distinguishes GAs from other optimization methods. This is because it gives individuals the opportunity to evolve by combining the strengths of both parents. In bGAs, crossover can be implemented by cutting the parents' chromosomes at a random point and swapping the two resulting portions (called single-point crossover). More complicated forms of crossover are multipoint crossover, in which several points of exchange are set, and uniform crossover is performed (each bit of the offspring's chromosome can be taken from either parent with equal probability). Crossover is only performed on a certain percentage of the population according to the crossover probability (P_{xo}). As for cGA, crossover with blending was used to maintain solution diversity according to the following formula defined by Radcliff (1991):

$$P_i^{new} = \beta \cdot P_{Mi} + (1-\beta)P_{Fi}\,,\ 0 \le \beta \le 1\,, \tag{1}$$

where P_i^{new} is the i^{th} variable in a new individual, and P_{Mi} and P_{Fi} are the property values of the same variable from the mother and father individuals, respectively. β is a blending coefficient that can remain constant for each crossover operation, or can be randomly chosen for each property.

In contrast to crossover, which is responsible for the exploitation of the evolution process, mutation adds randomness to the search process to explore new areas of the search space. The main concept of mutation is to cause small random alterations at single points in the chromosome. The number of mutation occurrences is governed by the mutation probability (P_{mut}), which is usually small (of the order of 0.01 to 0.1). Since any bit of a binary chromosome can only take two values, mutation can be applied in bGAs by switching the value of bits that are selected for mutation, **Figure 1**. Although the implementation in cGAs is different, the main function of mutation remains the same. Mutation in this algorithm can be attained by adding a normally distributed random number to the variable selected for mutation as shown in **Eq. (2)** (Haupt and Haupt, 2004):

$$P_i^{new} = P_i^{old} + \sigma \cdot N(0,1)\,, \tag{2}$$

where P is as defined above and $N(0,1)$ is a standard normal distribution with a mean of zero and a variance of one. The standard deviation (σ) of this property is computed for the current population. The added random value is scaled by the standard deviation of the current property to make sure the property does not exceed its feasible range.

In addition to the GA advantages previously mentioned, cGAs in particular are more appealing in our optimization because individuals can assume any value in the search domain, thereby providing higher resolution when compared to the discrete bGA. Besides, the cGA is more suitable for handling some of the nonlinear constraints. In particular, crossover and mutation had to be controlled in bGA to make sure the new chromosome is still a valid solution. For instance, the chromosome of a wellbore that has a z-coordinate of 4,550 ft is represented in binary bits by 1000111000110. If the limits in this coordinate were between 4,470 ft and 4,620 ft, mutating any of the first six bits would generate invalid solutions. A similar check was performed for all other variables and analogous controls were enforced for the crossover procedure. These extra steps increased the average reproduction time to 42 seconds per generation in bGA as opposed to just 3 seconds in cGA. Other optimization steps were completed in about the same time for both algorithms. The reproduction procedure described by the three GA operators is just one part of the optimization loop. **Figure** shows a flow chart of the complete procedure.

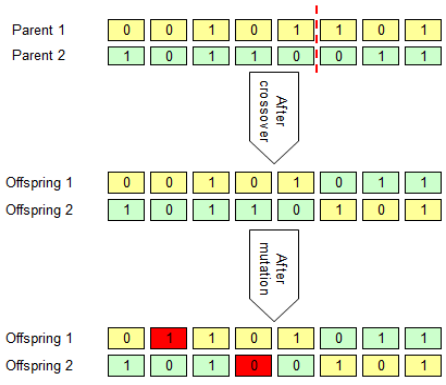

Figure 1: Simple crossover and mutation in bGAs.

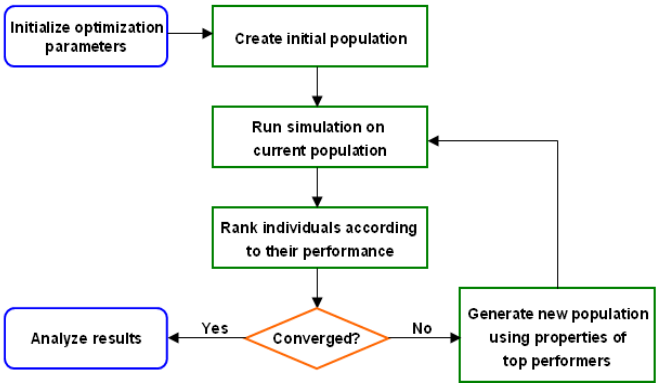

Figure 2: Flowchart of the overall optimization procedure using GAs.

cGAs Convergence

Since the GA is used here as the search engine, it is important to know if the algorithm would finally converge to a global optimum in a simplified problem with two optimization variables, and what conditions would affect convergence. Rudolph (1994) modeled the behavior of canonical GAs (standard GAs with no elitism and extra options) and showed that they do not guaranty convergence to the global optimum in their standard form.

A simple optimization test to illustrate the convergence of a cGA was designed. The problem is to find a vertical well with the highest production in a 2D homogenous square reservoir with closed boundaries. The dimensions of the reservoir

model used are 101×101. The best solution is known to be in the center of the reservoir, which falls in grid block (51, 51). A cGA with parameters given in **Table 1** was tested for comparison. The evolution path of the GA along generations can be observed in **Figure 3**, which shows all individuals of the population at each generation until convergence.

Table 1. Convergence test cGA parameters.

GA Parameter	Value
Population size	20
Maximum No. of generations	100
P_{xo}	0.5
P_{mut}	0.1
Selection fraction	0.7
Ranking scale	3

Table 2. Results for cGA convergence test.

Converged	Value
Avg. No. of simulations	103.5
Median No. of simulations	91.5

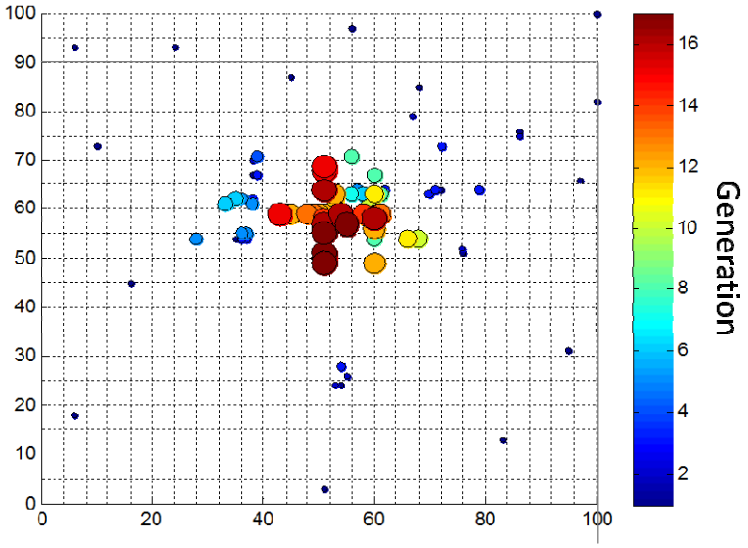

Figure 3: Evolution path of GA over generations.

To diminish the stochastic effects in testing cGA convergence, 100 different initial populations were used. Then, the cGA code was run 200 times for each initial population. Using the given parameters, all the 20,000 cases converged to the known optimum before 100 generations. Using the cGA, we converged after 103.5 simulations on average, **Table 2**. After an exhaustive search by simulating each location finds the solution/global optimum on average after 5,000 simulations.

Objective Function Evaluation

The described GA optimizations were performed using MATLAB, whereas all objective functions were evaluated with Schlumberger's reservoir simulator (ECLIPSE 2007a). Two types of objective functions are available in the formulation of this work: cumulative oil production and NPV. After each individual is simulated, calculation of each objective function is possible. While cumulative oil production represents a single value, NPV takes more consideration of the economics of the project. The economic model used here is based on the work of Yeten (2003). NPV is calculated based on a fixed yearly effective discount rate as:

$$NPV = \left(\sum_{n=1}^{Y} \sum_{p=o,g,w} \frac{1}{(1+APR)^n} Q_P^n \cdot C_p \right) - CAPEX, \quad (3)$$

where Q_p^n is the production rate of phase p during the year n, which is a simulation output, C_p is the revenue/cost per barrel of this phase, APR is the annual percentage rate, Y is the total number of discount years, and $CAPEX$ represents capital expenditures, which mainly consists of costs associated with drilling and completion. For this study, NPV parameters were assigned as listed in **Table 3**.

Table 3. Economic parameters used to calculate the NPV of the optimum lateral number study. These parameters were obtained from experience in this field.

Economic Parameter	Value
APR, %	10
Oil selling price, $/bbl	60
Water production cost, $/bbl	3
Water injection cost, $/bbl	3
CAPEX, MM$/well	1.5
C_{mill}, MM$/junction	1
C_{drill}, $/foot	800

Practical Application of cGA

Within the optimization process, numerous interactions take place between the GA and the reservoir simulator. The two parts have to be completely compatible to have a trouble-free process and reliable solutions. In this section, a good understanding of the reservoir model and optimized parameters is established to ease integration between the two parts.

Reservoir Models Considered

To understand the complexities associated with real reservoirs two reservoir models were used. One of them is a 40x40x7 channelized reservoir, **Figure 4**, whose properties are listed in **Table 4**. Only a single phase exists in this model and permeability is highly heterogeneous. The second model is for an offshore carbonate reservoir in the Middle East. In subsequent text, this reservoir will be referred to as the S1 reservoir. The reservoir extends over a 26 km x 41 km area and is currently under evaluation for full development. As can be seen in **Figure 5**, oil has accumulated due to a dome stratigraphic trap. Although only 14 vertical observation wells have been completed in S1, many wells were drilled to a deeper reservoir in the same field. Therefore, many core samples and open hole log data are available for the S1 reservoir. From this data, it has been recognized that permeability is mildly heterogeneous. While most areas have a permeability of around 200 mD, high permeability (1-2 Darcy) areas are scattered around the field, **Figure 6**. An areal isotropic permeability is used in the model with a vertical to horizontal permeability ratio of 0.05. Other important reservoir, rock, and fluid properties are listed in **Tables 5** and **6**. Because the field is operated above the bubble point pressure (P_{bub}), the simulation model only contains oil and water phases. The simulation model for this reservoir was built using structured corner point grids. Since the optimization process involved thousands of simulations, an upscaled version (23x41x14 cells) of the model was considered.

Table 4. Synthetic model properties.

Field Dimensions	6,000 ft×6,000 ft×210 ft
Φ	20%
k* (channels)	90 mD
k* (matrix)	1 mD
C_t	3.0×10^{-5} psi^{-1}
B_o	1.3

Table 5. S1 reservoir properties.

Φ*	23%
k*	300 mD
k_z/k_h	0.05
Avg. Thickness	151'
T*	145 °F
P^*_{res}	1,738 psig
C_r	4.5×10^{-6} psi^{-1}

Table 6. S1 fluid properties.

P_b	876 psia
Gas-Oil Ratio	224 scf/STB
Crude Grade	28.9 °API
μ_o	4.8 cp
B_o	1.13
C_o	6.5×10^{-6} psi^{-1}
λ_w	1.16
μ_w	0.52 cp
B_w	1
C_w	3.0×10^{-6} psi^{-1}

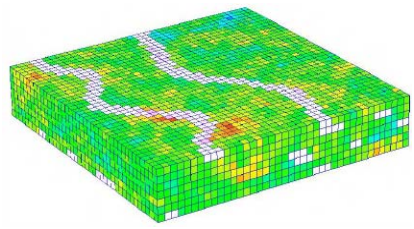

Figure 4. High permeability channels shown in the synthetic model.

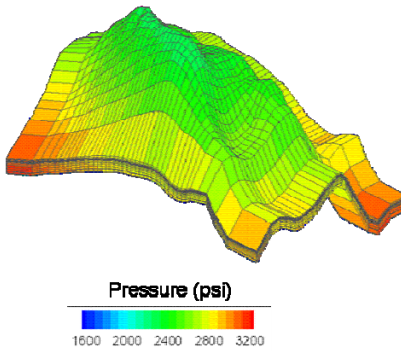

Figure 5. Average reservoir pressure for the S1 reservoir.

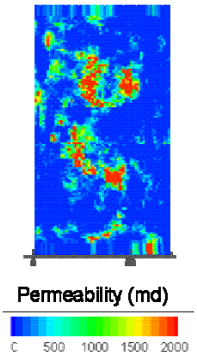

Figure 6. Permeability map for the S1 reservoir.

Multilateral Well Modeling

Variables

The optimized variables in this problem were chosen such that they possess three important characteristics. First, they have to be independent because each variable is selected and mated randomly with other individuals. Choosing independent variables also allows us to work with the lowest possible set of variables for the problem. Reducing the number of variables helps to

reduce the complexity of the problem and eases the optimization process. Second, the variables have to be selected such that the objective function is highly sensitive with respect to them. By doing so, GA operators are more capable of directing the search towards the optimum solution during the information exchange process. Finally, the variables should be easy to handle during the constraint enforcement stage. More elaboration on this issue will follow in the next section.

In this work, we use the nonconventional well model introduced by Yeten (2003), with some modifications. Since the well's mainbore can be represented by a straight line in the 3D space, six variables are sufficient to define its trajectory. These variables are: the three coordinates of the midpoint (x_{mid}, y_{mid}, z_{mid}), total well length (L_{tot}), vertical well distance between the toe and the heel (Z_h), and the top-view rotation angle (θ). Using Z_h as a variable is very handy in creating wells within the vertical limits of the reservoir. When it comes to laterals, four variables completely define their orientation since two degrees of freedom are lost as they are fixed to the mainbore. As a consequence, defining a variable between zero and one to describe the junction position relative to the total mainbore length (J_p) can replace the three midpoint coordinates. The other two lengths and angle variables complete the lateral's definition. **Figure 7** provides a visualization of the variable set. Other dependent parameters that are needed during optimization, such as heel and toe coordinates of the mainbore and laterals, can be calculated from the independent variables stated above according to the following equations:

$$
\begin{aligned}
x^{well}_{heel,toe} &= x^{well}_{mid} \pm \frac{1}{2} L^{well}_{xy} \sin\theta^{well} \\
y^{well}_{heel,toe} &= y^{well}_{mid} \pm \frac{1}{2} L^{well}_{xy} \cos\theta^{well} \\
z^{well}_{heel,toe} &= z^{well}_{mid} \pm \frac{1}{2} Z^{well}_{h}
\end{aligned}
\qquad
\begin{aligned}
x^{lat}_{heel} &= x^{well}_{heel} + J_p\left(x^{well}_{toe} - x^{well}_{heel}\right) \\
y^{lat}_{heel} &= y^{well}_{heel} + J_p\left(y^{well}_{toe} - y^{well}_{heel}\right) \\
z^{lat}_{heel} &= z^{well}_{heel} + J_p\left(z^{well}_{toe} - z^{well}_{heel}\right) \\
L_{xy} &= \sqrt{L^2_{tot} - Z^2_h}
\end{aligned}
\qquad (44)
$$

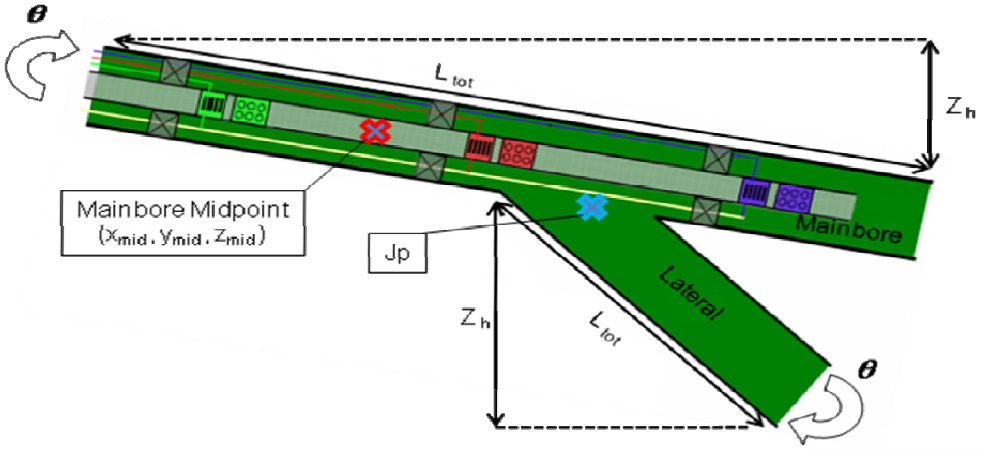

Figure 7. Well parameter representation in the well optimization problem.

The importance of using a low number of variables comes from the fact that the population size is typically taken proportional to this number. Increasing the number of wells and laterals can substantially increase optimization time and complexity as shown by the following equation:

$$Variable_{count} = Well_{count}\left(6 + 4\left(Lateral_{count}\right)\right). \qquad (5)$$

Constraints

In general, constraints for the synthetic model were imposed to keep generated wells within reservoir limits. When working with the real model, more constraints had to be introduced to make sure all proposed solutions are physically feasible. Some of these constraints were imposed to guarantee that the resulting wells are drillable, while others avoid creating solutions that are known to perform poorly due to violating common engineering practices. Considering that we have control in the GA initialization process, the constraints can be easily applied to the initial population. Nevertheless, reproduction might result in invalid solutions that are excluded from evaluation. Constraints were enforced on well/segment distance, total length of each lateral, vertical depth, and angle difference between any two segments. Once the maximum and minimum limits are defined for any of these properties (denoted as P), initial population properties can be generated using the following equation:

$$P_{init} = P_{\min} + rand(0,1)\left(P_{\max} - P_{\min}\right). \qquad (6)$$

Using a corner point grid and different cell heights in the S1 reservoir results in tilted planes as ceiling and floor of the cells, and therefore, different z-coordinate limits for any point in the xy-plane. Some manipulation had to be performed to keep

the toe and heel points of each segment within reservoir boundaries. This is done by devising plane formulas for the uppermost and lowermost planes bounding the x and y coordinates of the toe and heel:

$$n_x x_{plane} + n_y y_{plane} + n_z z_{plane} + d = 0 \quad , \tag{7}$$

where n_x, n_y, and n_z are the three components of an arbitrary vector, n, which is normal to the plane. According to this equation, the Z_h value for each well was chosen randomly such that it honored the z-coordinates limits.

Results

In this section, performance comparisons between the two GA variants are presented. Then, we proceed by applying the more robust algorithm to the reservoir models. The first step in this stage is to tune the different GA parameters by performing a sensitivity study. Then, we investigate the possibility of further empowerment of the algorithm by combining it with other helper tools. The resulting hybridized algorithm from this step was then run on the reservoir model to draw some conclusions on the optimal field development plan. The final step was to examine the reliability of obtained optimum solutions using the original fine grid model of S1.

Unless otherwise mentioned, optimization parameters listed in **Table 7** were used. The rationale of choosing these values is discussed in the GA parameter sensitivity section. Additionally, all producers were simulated with a bottom-hole pressure (BHP) control of 1,000 psig, which is above P_{bub}. Injectors were operated with a BHP of 2,500 psig. The population size of each run was equal to the number of variables, which is calculated from **Eq. (5)**. The maximum number of generations was adjusted appropriately to accommodate the complexity of the problem. Three runs were made to get a dependable average of the case studied. In a few cases, one of the three runs was omitted due to abnormally low objective function value and a replacement run was performed. The runs in this study were parallelized over 20 computer nodes.

Table 7. Common GA parameters used in this work.

GA Parameter	Value
Population size	30
Maximum No. of generations	35
P_{xo}	0.8
P_{mut}	Dynamic: 0.01 to 0.05
Selection fraction	0.5
Ranking scale	2

Comparing Performance of the Two GAs

After understanding the mechanisms behind the two variants of GA, it is of interest to test which of them is more suitable for the S1 reservoir optimization problem by comparing their performance. Two examples were designed to achieve this objective in the two reservoir models. In the first example, the locations of two producers and one injector, each with two laterals, are optimized. The objective function used here is the NPV of the field over four years. **Figure 8** compares the evolution of best individuals using bGA and cGA for this example. By comparing the evolution path of optimizations with binary and continuous GAs, we observe more gradual evolution for cGAs, as opposed to typical jumps and flat periods observed in the evolution path using bGAs. The more steady evolution for cGAs increases the chance of achieving fitter outcomes. The higher dimensionality of continuous search space and absence of the jumps observed with bGA, have resulted in slower convergence rates in some cases.

In the second example, the optimization algorithm found the best location and configuration for three deviated producers and two deviated injectors completed in the S1 reservoir model. A population size of 30 was used, which is equal to the number of variables. Cumulative oil production over a 10 year period is used as the objective function. All other GA and optimization parameters were kept the same for the two GAs. Both algorithms were run six times with a different random initial population for each run. **Figure 9** compares the evolution of the objective function for the best individual averaged over all the runs, as well as the best out of the six runs. A number of observations can be made from this plot. First, similar to the first example, the cGA evolves the solution in a gradual fashion. Conversely, the bGA in general shows some jumps followed by flat regions. Second, the averages of the two methods are close throughout the run; the cGA is only 3% higher than the bGA. The best solution from the cGA was higher than that from the bGA by around the same percentage.

When comparing the six runs from the two algorithms, it was noticed that the individual runs in the cGA are more clustered around its average than the bGA. Standard deviation calculations confirmed this statement, which might give an indication about the robustness of the algorithm; we are more likely to get a good answer with cGA if fewer runs are performed. Another measure of robustness is provided by the repeatability of the algorithm. The moving average for each algorithm was calculated by averaging objective function results by the end of optimization after an additional run has been performed. This exercise showed that the average in cGA is stabilized after around three optimization runs. We consider the average to be stabilized if the change from adding an additional run was within ±1% because it is unlikely that a decision

would be changed based on such a small change. In contrast, the bGA required five runs for its average to reach a stable region as defined above. In all subsequent runs, the cGA will be repeated three times to obtain a representative average. Another advantage that gives more preference to the cGA is the increased flexibility it provides in handling constraints as mentioned earlier.

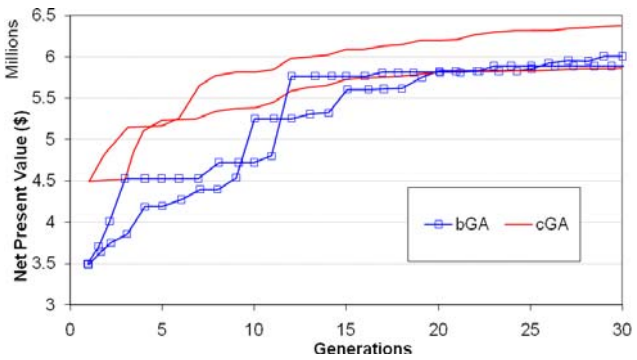

Figure 8. Fittest individual performance comparison of two runs from the bGA and the cGA in the synthetic model.

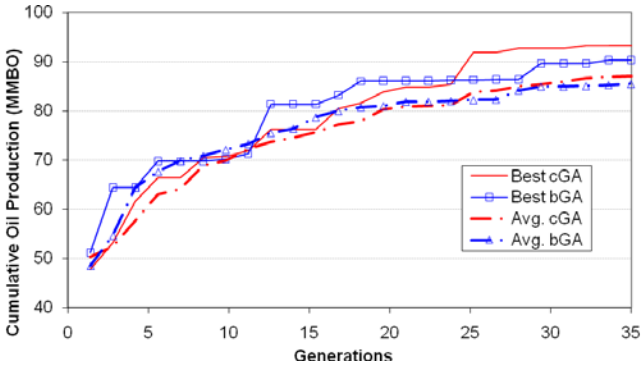

Figure 9. Fittest individual performance comparison averaged over six runs and the best out of all the runs.

GA Parameter Sensitivity

Due to the stochastic nature of GAs, the final solution of the same problem is usually different when the algorithm is run a number of times. This is related to the different GA probabilities, selection fractions, initial population, and number of generations. A summary of the effects of some important GA parameters on the resulting final solution is presented next.

Crossover Probability (P_{ox}): Several values ranging from 0.4 to 1.0 were tested. In general, better results were obtained with a high crossover mutation probability. This was observed in both reservoir models and is consistent with the conclusions of Guyaguler (2002) and Yeten (2003), where the optimum P_{xo} values were found to be between 0.8 and 1.0. In the synthetic model, faster convergence was achieved with high values.

Mutation Probability (P_{mut}): A range of values from 0.001 to 0.2 was tested. Values around 0.05 obtained better solutions for most cases. This was not true for all optimization stages. Specifically, using low P_{mut} value appears to be better in the early stages of optimization. As we reach the later parts of optimization, a low P_{mut} seems to cause premature convergence with no further improvement. A similar study by Montes et al., (2001) and Emeric et al., (2009) reached the same conclusion. They suggested the need for low mutation rates in the beginning because allowing high rates at this stage would overshadow the role of crossover. In later generations, however, a high mutation rate is needed as solutions cluster and become more similar. This high mutation rate maintains diversity in the population, which increases the possibility of finding better solutions. Results are plotted in **Figure 10** for P_{xo} = 0.8 and a selection fraction of 0.5, but most other combinations showed similar trends. To show the advantages of medium and low mutation probabilities, an additional run was made with a dynamic mutation probability. The orange curve in **Figure 10** shows the result for this run where P_{mut} was set to 0.01 until the end of the tenth generation, then increased to 0.05 thereafter. This change shows an improvement of approximately 8% over P_{mut} = 0.05. Repeating the same procedure with different initial populations revealed that the optimum point at which mutation should be changed is different in the different runs. This suggests changing the mutation probability automatically when a low value stops improving the solution.

Initial Population Size: As observed in other stochastic search algorithms, seed fitness has a big impact on final solutions. The population size depends on the nature, complexity, and number of variables of the problem. Typically, the population is generated randomly such that it covers a broad range of possible solutions. A couple of studies have suggested that problems of moderate complexity should have a population size equal to the chromosome's string length in bGA (Alander, 1992; Montes et al., 2001; Goldberg, 2004). An analogous population size recommendation for cGAs has not been established in the literature. In this study, we evaluated this effect by solving the same problem using a population size equal to the number of variables in the problem (base case). Then, we solved it again by doubling the population size and by using a dynamic population size (starts with a high population that decreases in later stages). After optimizing the problem three times, the latter two cases were, on average 6% better than the base case by the end of the run. Consequently, this improvement was associated with higher computational cost; therefore, they were not used for subsequent studies.

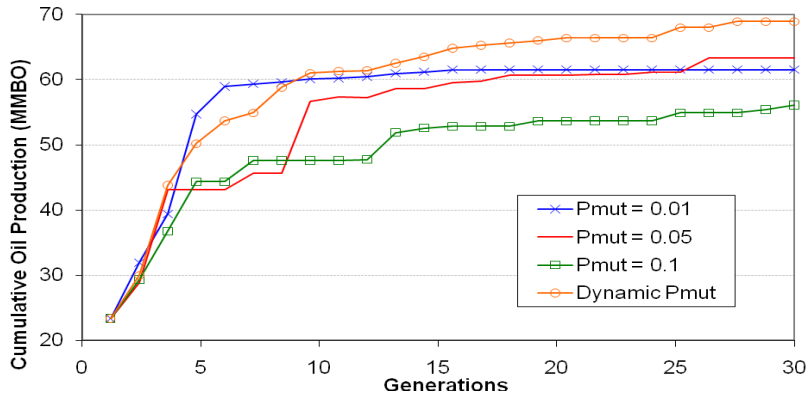

Figure 10. Fittest individual performance comparison averaged over three runs for different mutation probabilities.

Required Number of Generations: Here, it is desired to find the number of generations required for a certain optimization to converge. This, of course, is also dependent on the complexity of the problem mainly represented by the number of variables. To discern the effect of the number of variables on the maximum required number of generations, three cases were optimized for 60 generations. The three cases were picked to represent a relatively simple, a moderately complicated, and a complex problem. The first case contains three deviated producers and two deviated injectors. The second and third cases consist of the same number of wells, but each well has two and four laterals, respectively. The resulting number of variables, **Eq. (5)**, for this set-up is 30, 70 and 110. Each case was repeated three times. Assuming that convergence occurs when no improvement in the objective function is observed for 10 consecutive generations, it appears that 25 generations are sufficient to provide a converged solution when 30 variables are used, as shown in **Figure 11**. This number increased to 32 and 45 when the number of variables was increased to 70 and 110, respectively.

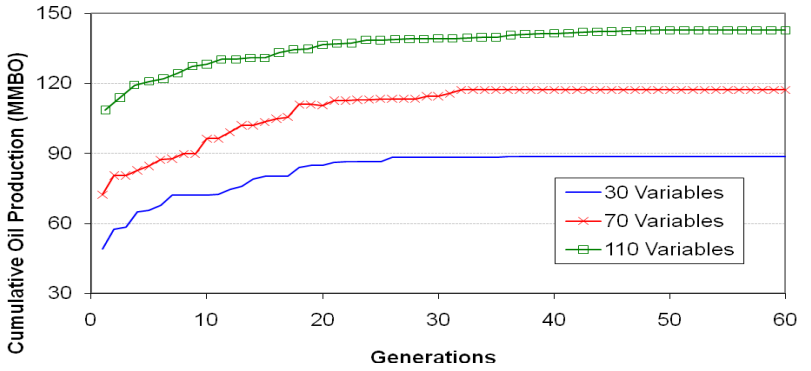

Figure 11. Convergence of the fittest individual averaged over three runs for problems with different number of variables.

Effects of Using Helper Tools

One of the advantages of GAs is the ability to hybridize them with different optimizers and helper tools. Some of these helper tools are designed to speed up the algorithm, while others focus on improving the final GA solutions. Some of the most commonly used tools in the literature are ANNs, the HC, and rejuvenation. The first helper tool was not implemented as Rigot (2003) found that the method requires in complex reservoir models substantial simulation time to train the neural network to see plausible improvement. The other two functions were already tested by Yeten (2003) on a synthetic model and are considered in this work to find out how well the algorithm performs in real field applications. Results of their implementation are presented next.

Rejuvenation

The basic concept of rejuvenation is quite simple; the best solutions encountered during optimization are resurrected after some generations. These solutions from older generations are referred to as ancestors (Fichter, 2000). These ancestors might have been mutated in an unfavorable manner and bringing them back creates opportunities for better mating of the top individuals, and therefore, better offsprings.

Rejuvenation was tested by optimizing the location of three producers and two injectors, each having three laterals. It was applied by keeping a pool of fittest individuals during the course of optimization that is of similar size as the population size. Every five generations, this pool makes up the new population. Mutation is still applied to all individuals except the fittest one,

but no crossover is considered during this generation. This optimization case will be referred to as Case 1, whereas the Base Case was run normally without rejuvenation. Case 2, another variation of Case 1 was run by composing only half of the generation from ancestors while the other half was created by regular crossover. Each case was run three times and the average results are plotted in **Figure 12**. By comparing the outcome of the Base Case (blue curve) with Cases 1 and 2 (red and green curve, respectively), it can be concluded that rejuvenation did not introduce noticeable improvement to the objective function of the fittest individual.

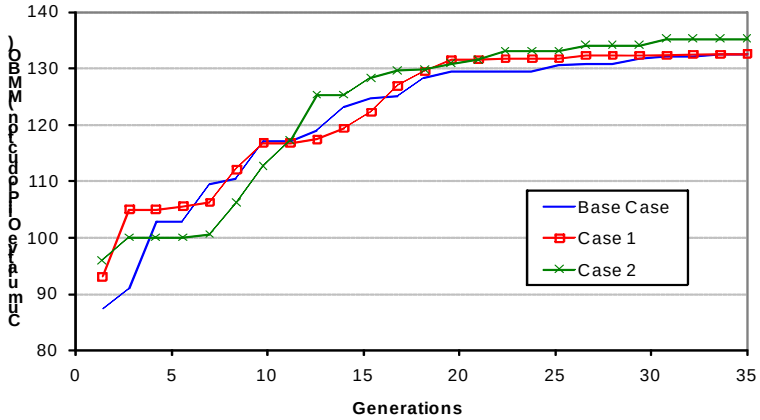

Figure 12. Fittest individual performance comparison averaged over three runs with different rejuvenation scenarios. Base Case: no rejuvenation applied. Case 1: rejuvenation applied to the whole population. Case 2: rejuvenation applied to half of the population.

Hill Climber

The HC method applied here is a modification of the Hooke-Jeeves direct search algorithm (Hooke and Jeeves, 1961). As explained in the works of Reed and Marks II (1999) and Koza et al., (1999), the method is similar to GAs in being a derivative-free method, though it is more systematic and less random. Despite this, it is considered to be less applicable for complicated problems with multiple optima because it converges to local optima (it starts with only one initial solution, providing less exploration of the search space). The hill climbing procedure starts by taking a forward and backward step along one of the coordinates of the problem. Search within the current coordinate is halted when no further improvement is realized in this direction, after which, the next coordinate is tested. A series of new steps are attempted in the direction that improves the objective function. The greediness of the method arises from the fact that the search is only performed in the neighborhood of the initial guess with no information about other areas of the objective function surface. Blending GAs with the HC helps to escape entrapment in local optima. At each generation, the optimal solution found by the GA can be further improved by applying the HC, which searches for improvement in the neighborhood of that solution. The overall optimization process is now modified as depicted in **Figure 13**.

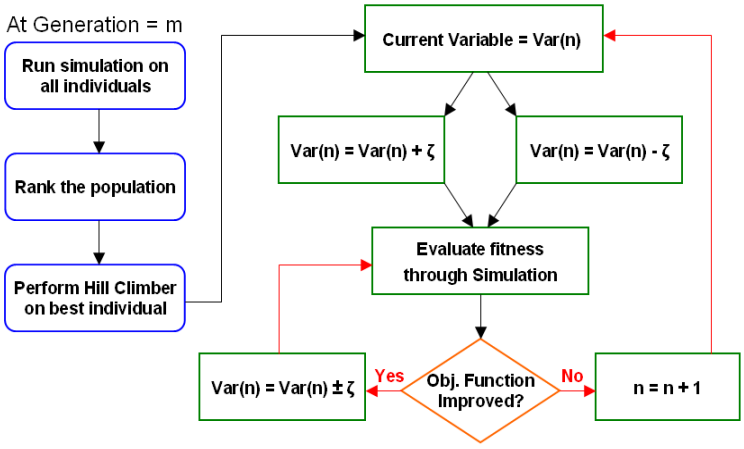

Figure 13. Optimization flowchart using genetic algorithms with the HC.

Four cases were tested to evaluate the benefits of adding the HC to the GA. The studied optimization problem consisted of two deviated producers and a deviated injector (18 variables). The Base Case represents optimizing the problem with the GA alone. In Case 1, the HC was applied to the best individual of the initial population; while Case 2 executed the HC on the final solution given by the GA. Finally, the two algorithms were hybridized in Case 3; that is, the HC was run on the best individual that the GA found at each generation. Results of the four cases are plotted in **Figure 14**.

The HC by itself (Case 1) was able to improve the solution by around 20% in 118 simulations, but the optimization appears to converge prematurely. This improvement was achieved by the GA alone (Base Case) in about 200 simulations. Running Case 2 at the end of the Base Case causes rapid improvement but the optimization again stopped too early. The benefit of the hybridized algorithm can be recognized by the higher objective function value achieved by the end of the run in Case 3 when compared to other cases. Since most of the improvement in Case 3 took place in the early stages of optimization, it was only applied in the first five generations in subsequent tests to avoid the high computational cost it carries. It is important to highlight that the above conclusions cannot be generalized as the performance of the HC alone is governed by the number of variables in the problem. The good performance observed by using the HC alone is expected to decrease for a more complex optimization problem.

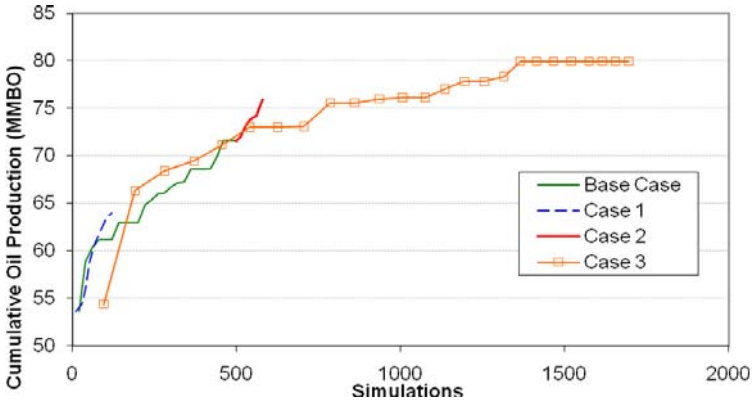

Figure 14. Fittest individual performance comparison averaged over three runs for different HC scenarios. Base Case: GA was run alone. Case 1: HC was run alone. Case 2: HC run at the end of GA. Case 3: both algorithms run concurrently.

Optimum Number of Laterals

This case strives to assess the sensitivity of the objective function to the number of laterals in each well. It can be used as a quick screening test to evaluate the performance of each well configuration and shows the incremental value of adding more laterals. To achieve this, the locations of three deviated producers and two deviated injectors were optimized first. Then, we added laterals to each well until the fourth lateral is reached. Optimization parameters are as described earlier with NPV as the objective function to account for associated drilling costs according to **Table 3**.

Results for the five optimization cases averaged over three runs are presented in **Figure 15**. As the number of laterals increases up to the fourth lateral, NPV improves as well, but after that fourth lateral there is not a significant value increment. It is worth mentioning that the resulting well locations from each run were considerably different with apparent preference to particular regions that have higher permeability. The optimum well locations of each case can be seen in **Figure 16** (the oil-water contact is shown for one of the top reservoir layers). Most of the scenarios gave a peripheral injection pattern.

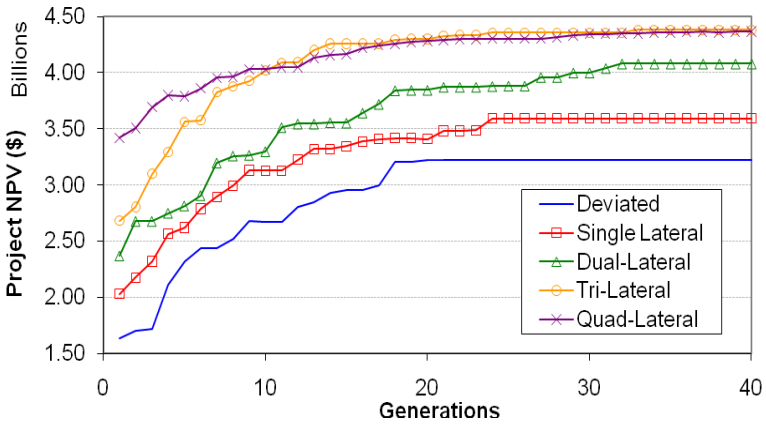

Figure 15. NPV of the fittest individual averaged over three runs when a different number of laterals are used.

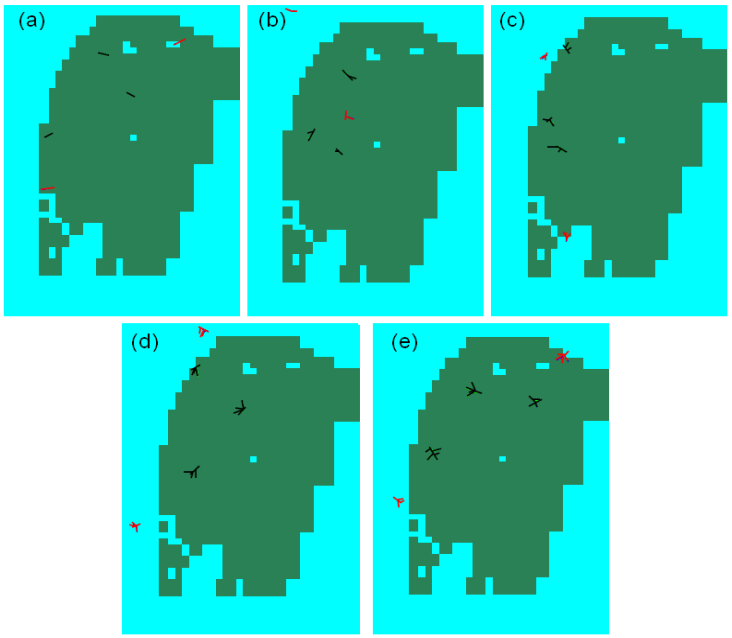

Figure 16. Optimum producer (black) and injector (red) well locations for: a) Deviated, b) Single lateral, c) Dual-lateral, d) Tri-lateral, and e) Quad-lateral wells.

To alleviate the effects of initial population and GA randomness in the optimization results, another run was made by using the initial population for the first case (deviated wells) as the base for each subsequent optimization. For example, the initial population for the dual-lateral case would be identical to that of the base case after adding two random laterals in each well of every individual. By doing so, the optimization would give more information about the effect of added laterals rather than changing well locations. As anticipated, the NPV in this optimization continued to increase even when four and five laterals were added, **Figure 17**, which indicates good potential for the MRC wells in this field. The incremental NPV decreased as the number of laterals increased.

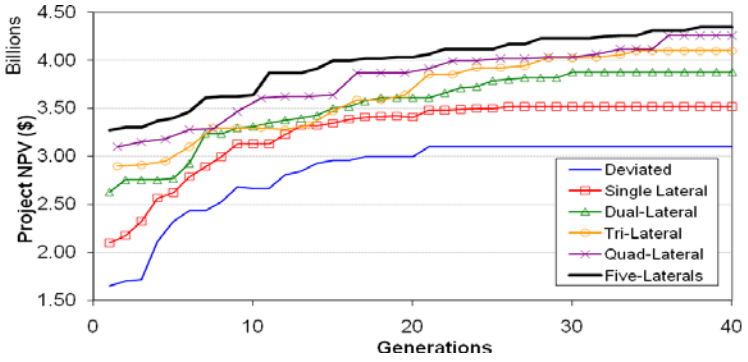

Figure 17. NPV of the fittest individual averaged over three runs when different number of laterals is used for the same initial population.

Fine Model Results

Using the upscaled model in previous sections allowed conducting several tests to evaluate the optimization algorithm. Running these optimizations on the original fine model available for this field, the optimization time would have increased by approximately fortyfold. Time limitations in reservoir management would not always tolerate such a long period of time before making a decision. Though, it is still important to see if solutions obtained from the upscaled model are viable ones when the fine model is used, which is the objective of this section. The original fine model consists of 69x122x14 cells and its other properties are as discussed earlier.

The initial solution for this fine reservoir model optimization run was composed from the multiple solutions already obtained from previous runs on the coarse reservoir model. All of these individuals have three producers and two injectors with three laterals. In addition to the solutions with highest NPV, a number of solutions with intermediate NPVs were added to the population to maintain diversity. As this population still has to go through the reproduction phase, we can compare how

well the individuals perform in the two models. **Figure 18** provides this comparison for 10 out of 30 individuals chosen for the initial population. These 10 individuals were selected according to the distribution of good and intermediate solutions from the initial population. Although most individuals that performed well in the coarse model are still considered good solutions for the fine model, the plot clearly shows that well performance ranking is not preserved in the two models. The upscaled model tends to result in higher cumulative production, with individuals G and H being the only exception. Since the coarse model was obtained by single-phase upscaling, these discrepancies might be addressed by applying multiphase near-well upscaling (Yeten, 2003).

Using this initial population, the optimization algorithm was applied on the fine reservoir model for 25 generations and was repeated twice. The HC was applied to the first five generations. The evolution of the objective function can be seen in **Figure 19**. Unlike most other runs, the percentage difference in the objective function value between the initial and final solutions is relatively small (around 10%) due to the high fitness of the initial population. To get an insight of the added benefit from using the upscaled model to get a better-quality initial solution for the fine model, two additional optimizations were run on the fine model starting from a random initial population. By the end of the run, the final solution from these runs yielded an objective function that is, on average, 22% lower than the previous case.

Best solutions from the fine and upscaled models are depicted in **Figure 20** to provide a comparison of resulting well locations. As expected and initially indicated by **Figure 18**, the best wells in the fine model took different locations from the best solution in the coarse model (individual A). These positions are, however, close to those of individual C with some differences in lateral orientation.

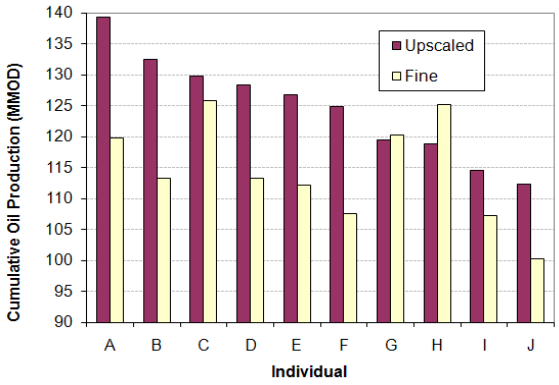

Figure 18. Individual performance comparison between the fine and coarse model.

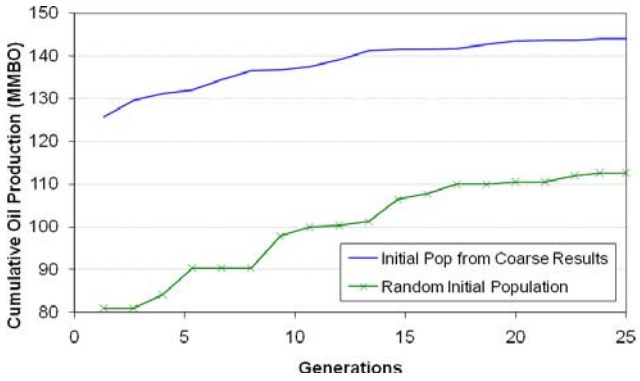

Figure 19. Objective function evaluation of the fittest individual for the fine model after two optimization runs. The initial population for this optimization is composed from the fittest individuals from the coarse model.

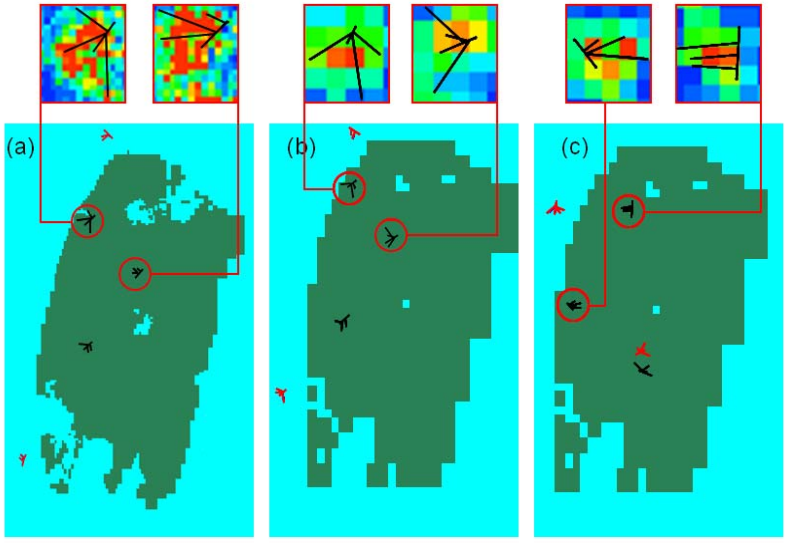

Figure 20. Comparison of producer (black) and injector (red) well locations between: a) best individual in the fine model, b) individual C from the coarse model, ranked third in the coarse model but with similar locations to the individual in a), and c) best overall individual from the coarse model.

Conclusions

1. The comparison of two types of GAs (binary and continuous) shows that for the type of problems considered here, cGA provided consistent results in shorter computational time.
2. The use of the HC with cGA delivered the best final solutions. It was found that complete hybridization of the two algorithms is more beneficial than running one after the end of the other. Moreover, most of the improvement from this tool was achieved in early generations. In contrast, other helper tools like rejuvenation did not have a major impact on results.
3. Several optimization runs on the real field model revealed that this reservoir has high potential for MRC wells as the NPV continuously increased for up to five laterals (additional laterals were not tested). However, having a proper initial population was crucial to reaching this conclusion.
4. While in general, well locations differed from one run to another, there was a tendency for producers and injectors to seek particular regions of the reservoirs. The orientation of well laterals, on the other hand, appeared to be arbitrary. Generally speaking, the wells were arranged in a peripheral injection with a central production system.
5. By running the optimization on the fine and coarse reservoir models, it was found that the fittest individuals are not necessarily in agreement in both models. These differences might be reduced by implementing near-wellbore upscaling.

Nomenclature

Abbreviations

ANN	artificial neural networks
APR	annual percentage rate
bGA	binary genetic algorithm
BHP	bottom-hole pressure
cGA	continuous genetic algorithm
GA	genetic algorithm
HC	hill climber
MLW	multilateral well
MRC	maximum reservoir contact well
NPV	net present value

Symbols

B	formation volume factor
c	compressibility
CAPEX	capital expenditures
i	index
J_p	junction position
k	permeability
L	length
N	standard normal distribution
n	normal vector, ranking of an individual
rand	random number
P	probability, property
r	ranking scale factor

Symbols (cont.)

Y	total discount years
Z_h	vertical distance between heel and toe
β	blending coefficient
μ	viscosity
σ	standard deviation
φ	porosity
Θ	top-view rotation angle

Superscripts

lat	lateral
mb	mainbore
well	well
*	average

Subscripts

F	father
i	index
M	mother
mut	mutation
o	oil
p	phase
r	rock
tot	total
w	water
xo	crossover

Acknowledgements

The authors would like to thank Saudi Aramco, Stanford Reservoir Simulation Consortium (SUPRI-B) and Stanford Smart Fields Consortium for supporting this work. We are particularly thankful to Dr. David Echeverria Ciaurri for his helpful comments on this work.

References

Alander, J. (1992). "On Optimal Population Size of Genetic Algorithms," In: Proceedings of CompEuro '92 Computer Systems and Software Engineering, pp. 65-70.

Bittencourt, A. and Horne, R. (1997). "Reservoir Development and Design Optimization," paper SPE 38895, presented at the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, 5-8 October 1997.

Emeric, A., Silva, E., Messer, B., et al. (2009). "Well Placement Optimization Using a Genetic Algorithm with Nonlinear Constraints," paper SPE 118808, presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, 2-4 February 2009.

Farshi, M. (2008). *Improving Genetic Algorithms for Optimum Well Placement*, Master's Report, Department of Energy Resources Engineering, Stanford University, California.

Fichter, D. (2000). "Application of Genetic Algorithms in Portfolio Optimization for the Oil and Gas Industry," paper SPE 62970, presented at the SPE Annual Technical Conference and Exhibition, Dallas, Texas, 1-4 October 2000.

Goldberg, D. (2004). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading.

Guyaguler, B. and Horne, R. (2001). "Uncertainty Assessment of Well Placement Optimization," paper SPE 71625, presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, 30 September-3 October 2001.

Guyaguler, B. (2002). *Optimization of Well Placement and Assessment of Uncertainty*, Ph.D. dissertation, Department of Energy Resources Engineering, Stanford University, California.

Haupt, R. and Haupt, S. (2004). *Practical Genetic Algorithms*, 2nd edition, John Wiley & Sons, New York City, New York.

Holland, J. (1992). Genetic algorithms. *Scientific American*, July, pp. 66-72.

Hooke, R. and Jeeves, T. (1961). Direct search solution of numerical and statistical problems, *Journal of the ACM*, Vol. 8, No. 2, pp. 212-229.

Horn, M., Plathey, D. and Ibrahim, O. (1997). "Dual Horizontal Well Increases Liquids Recovery in the Gulf of Thailand," paper SPE 38065, presented at the SPE Asia Pacific Oil and Gas Conference, Kuala Lumpur, Malaysia, 14-16 April 1997.

Koza, J., Bennett, F., Andre, D. and Keane, M. (1999). *Genetic Programming III: Darwinian Invention and Problem Solving*. Morgan Kaufmann Publishers, San Francisco, California.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms.* The MIT Press, Cambridge, Massachusetts.

Montes, G., Bartolome, P. and Udias, A. (2001). "The Use of Genetic Algorithms in Well Placement Optimization," paper SPE 69439, presented at the SPE Latin American and Caribbean Petroleum Engineering Conference, Buenos Aires, Argentina, 25-28 March 2001.

Nogueira, P. and Schiozer, D. (2009). "An Efficient Methodology of Production Strategy Optimization Based on Genetic Algorithms," paper SPE 122031, presented at the SPE Latin American and Caribbean Petroleum Engineering Conference, Cartagena, Colombia, 31 May-3 June 2009.

Radcliff, N. (1991). *Formal Analysis and Random Respectful Recombination*. In: Proceedings of 4th International Conference on Genetic Algorithms, San Mateo, CA: Morgan Kauffman.

Rudolph, G. (1994). "Convergence Analysis of Canonical Genetic Algorithms," IEEE Trans. on Neural Networks, special issue on Evolutionary Computation, Vol. 5, No. 1, pp. 96-101.

Reed, R. and Marks II, R. (1999). *Neural Smithing, Supervised Learning in Feedforward Artificial Neural Networks*. The MIT Press, Cambridge, Massachusetts.

Rigot, V. (2003). *New Well Optimization in Mature Fields*, Master's Report, Department of Energy Resources Engineering, Stanford University, California.

Saleri, N., Salamy, S., Mubarak, H., Sadler, R., Dossary, A.,and Muraikhi, A. (2003). "Shaybah-220: A Maximum Reservoir Contact (MRC) Well and its Implantations for Developing Tight Facies Reservoirs," paper SPE 81487, presented at the SPE 13th Middle East Oil Show and Conference, Manama, Bahrain, 5-8 April 2003.

Taylor, R. and Russel, R. (1997). "Drilling and Completing Multilateral Horizontal Wells in the Middle East," paper SPE 38759, presented at the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, 5-8 October 1997.

Yeten, B., Durlofsky, L. and Aziz, K. (2002). "Optimization of Nonconventional Well Type, Location and Trajectory," paper SPE 77565, presented at the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, 29 September-2 October 2002.

Yeten, B. (2003). *Optimum Deployment of Nonconventional Wells*, Ph.D. dissertation, Department of Energy Resources Engineering, Stanford University, California.