HORIZONTAL WELL PLACEMENT OPTIMIZATION IN GAS RESERVOIRS USING GENETIC ALGORITHMS

A Thesis

by

TREVOR HOWARD GIBBS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2010

Major Subject: Petroleum Engineering

HORIZONTAL WELL PLACEMENT OPTIMIZATION IN GAS RESERVOIRS

USING GENETIC ALGORITHMS

A Thesis

by

TREVOR HOWARD GIBBS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Co-Chairs of Committee,	Ding Zhu
	Hadi Nasrabadi
Committee Members,	Christine Ehlig-Economides
	Yuefeng Sun
Head of Department,	Stephen A. Holditch

May 2010

Major Subject: Petroleum Engineering

ABSTRACT

Horizontal Well Placement Optimization in Gas Reservoirs

Using Genetic Algorithms. (May 2010)

Trevor Howard Gibbs, B.S., Stephen F. Austin State University

Co-Chairs of Advisory Committee, Dr. Ding Zhu
Dr. Hadi Nasrabadi

Horizontal well placement determination within a reservoir is a significant and difficult step in the reservoir development process. Determining the optimal well location is a complex problem involving many factors including geological considerations, reservoir and fluid properties, economic costs, lateral direction, and technical ability. The most thorough approach to this problem is that of an exhaustive search, in which a simulation is run for every conceivable well position in the reservoir. Although thorough and accurate, this approach is typically not used in real world applications due to the time constraints from the excessive number of simulations.

This project suggests the use of a genetic algorithm applied to the horizontal well placement problem in a gas reservoir to reduce the required number of simulations. This research aims to first determine if well placement optimization is even necessary in a gas reservoir, and if so, to determine the benefit of optimization. Performance of the genetic algorithm was analyzed through five different case scenarios, one involving a vertical

well and four involving horizontal wells. The genetic algorithm approach is used to evaluate the effect of well placement in heterogeneous and anisotropic reservoirs on reservoir recovery. The wells are constrained by surface gas rate and bottom-hole pressure for each case.

This project's main new contribution is its application of using genetic algorithms to study the effect of well placement optimization in gas reservoirs. Two fundamental questions have been answered in this research. First, does well placement in a gas reservoir affect the reservoir performance? If so, what is an efficient method to find the optimal well location based on reservoir performance? The research provides evidence that well placement optimization is an important criterion during the reservoir development phase of a horizontal-well project in gas reservoirs, but it is less significant to vertical wells in a homogeneous reservoir. It is also shown that genetic algorithms are an extremely efficient and robust tool to find the optimal location.

DEDICATION

This research is dedicated to my parents, teachers, and mentors…

ACKNOWLEDGEMENTS

I would mainly like to thank the co-chairs of my committee, Dr. Zhu and Dr. Nasrabadi. This work would never have happened without their guidance.

I would also like to thank the remaining members of my committee, Dr. Ehlig-Economides and Dr. Sun.

Thanks also go to my parents, teachers, and mentors who have guided me from a young age, and raised me to be the person I am today.

Finally, I would like to thank both Texas A&M University and Texas A&M University at Qatar for providing me the opportunity to get a first-class education, and preparing me for my career.

NOMENCLATURE

GA	Genetic Algorithm
HGA	Hybrid Genetic Algorithm
l	String Length
MCF	Thousand Cubic Feet
MD	Millidarcy
MMSCF	Million Standard Cubic Feet
N_{POP}	Population Size of Genetic Algorithm
NPV	Net Present Value
P_c	Crossover Probability
P_m	Mutation Probability
P_s	Survival Probability
PSI	Pounds Per Square Inch

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

STATEMENT OF THE PROBLEM

Well location determination during the reservoir development phase of a project is a significant step. The decision relies on numerous factors, most of which are nonlinearly correlated parameters, making intuitive judgment difficult. Drilling a well in a non-optimal location leads to reduced hydrocarbon extraction, which in turn leads to a reduced Net Present Value (NPV) for the development project. Sometimes even a small difference in well placement can lead to a significant difference (positive or negative) in both well and field productivity. The problem becomes more significant when horizontal wells are being drilled because the contact between reservoir and well increases, and lateral direction relative to surface location of the well must be considered.

Since intuitive judgment is difficult during the location determination stage of the reservoir development phase, optimization models are considered due to their ability to evaluate complex interactions, such as those seen in a hydrocarbon-bearing reservoir. Reducing the number of simulations would increase the efficiency of the problem, and

---

This thesis follows the style of *SPE Journal.*

allow for applicability to more intricate problems. One of the potential problems rises when reducing the number of necessary simulations in the optimization problem. Getting stuck in local extrema is of main concern because it leads to suboptimal decisions. To combat such problems, an efficient algorithm must be used for computational feasibility. Without an efficient algorithm, the problem results in computation intensive, wasting time and money, which are not viable options in a real-world environment. The algorithm must also be able to find global optima, while avoiding local extrema. This requires a stochastic, as opposed to a deterministic, approach to the problem. The global optima requirement generally cancels out all calculus-based, hill-climbing methods as the main solvers. Also, the algorithm must be a generalized answer to the problem to allow usage over a wide variety of problems. The generalization characteristic of the algorithm requires the ability to handle varying types and numbers of parameters.

Taking the above information into consideration, this research proposes the use of a genetic algorithm to prove or disprove the notion that a correlation exists between well placement in a gas reservoir and the reservoir performance. Genetic algorithms are stochastic algorithms providing efficiency due to their robustness, which is “the balance between efficiency and efficacy necessary for survival in many different environments.” (Goldberg 1989) Genetic algorithms are also extremely flexible due to their generalized assumptions. All that is needed for a genetic algorithm to run is a population of strings with an associated fitness value for each string. The strings represent a combination of factors relative to the problem. In this research, the strings define the location of the wells, well type, and lateral orientation for horizontal wells. The fitness value is what the problem defines as the optimization parameter. This research optimizes the cumulative gas production based on well location. The fitness value can literally be whatever parameter is needed to be optimized in any problem. The mechanics of the genetic algorithms remain the same in every application, requiring only a change in the input parameters and fitness value function for different problems. Finally, genetic algorithms make use of parallelization since they modify several solutions simultaneously. All of these properties make genetic algorithms the logical choice to be the basis in answering the well location determination problem in a gas reservoir.

LITERATURE SURVEY

"Genetic algorithms are theoretically and empirically proven to provide robust search in complex spaces." (Goldberg 1989) A reservoir is a perfect model for a "complex space," providing an optimal basis for the GA to showcase its benefits over other methods. A reservoir can have thousands of local minima and maxima, making deterministic approaches extremely difficult to implement. The main concern of this research is to determine the effect on gas production of horizontal well placement in gas reservoirs with the use of a genetic algorithm approach. Although this specific application of genetic algorithms has not been studied (to the best of our knowledge), there has been research completed on the use of genetic algorithms in other applications.

Beckner and Song (1995) optimized the drilling schedule and well location in an oil reservoir through a traveling salesman structure with the use of Simulated Annealing. Bittencourt and Horne (1997) approached the well placement optimization problem using a genetic algorithm and polytope method combination, which they termed a Hybrid Genetic Algorithm (HGA). The HGA improved economic forecasts and decreased the computational workload during the optimization process. Pan and Horne (1998) used a kriging proxy to decrease the necessary number of simulations required to optimize well locations in a gas reservoir. The kriging proxy improves the local search of the genetic algorithm so the combination of the two is more powerful than the single contributors.

Mohaghegh (1998) used genetic algorithms to provide treatment design optimization, and economic analysis to select re-stimulation candidates based on the available data. He then developed a comprehensive software tool to aid engineers in selecting wells for re-stimulation in a gas storage field in Ohio.

Gűyagűler and Gűmrah (1999) used genetic algorithms to optimize the production rate for a gas storage field.

Montes and Bartolome (2001) developed a Simple Genetic Algorithm to optimize well placement in two different oil fields, with varying permeability and porosity values using ECLIPSE as the commercial simulator. One model is a simple three-layer model, and the other is a significantly more complex model. The two models were chosen to analyze the performance of the GA in terms of time and the quality of the solutions.

Sarich (2001) developed a method for using genetic algorithms to determine the best value-creating portfolio of projects taking into account numerous business constraints from a database of potential projects. Sarich applied his methodology in a project selection process of 30 wells, analyzing the NPV, capital requirements, and oil and gas volumes in an attempt to choose the combination of wells that result in the highest NPV. To make the problem more applicable to real-world problems and requiring some projects to be omitted, he used a capital budget of $30 million when the cost to drill all 30 wells with a positive NPV is $43.625 million. Sarich successfully showed an improvement over traditional oil and gas project selection techniques with the use of his methodology.

Gűyagűler and Horne (2001) developed a HGA to determine the uncertainty in well placement determination in terms of monetary value. Then they used their HGA, consisting of a simple genetic algorithm, polytope algorithm, kriging algorithm, and neural networks, in a doctoral thesis (2002) proposing a reduction in the required number of simulations for the optimal well placement problem in a waterflooding project in the Gulf of Mexico. They were able to determine the optimal placement of up to four water injection wells in the offshore Pompano field. Waterflooding with four injectors was deemed the most profitable in terms of Net Present Value (NPV). They also studied the optimization development plans for a real-world reservoir in the Middle East. The GA was setup in parallel on four processors because the reservoir model was half of a million cells. The drilling schedule of thirteen wells was optimized while still meeting the production target specified. The problem was set up as a traveling salesman problem with the order of the wells to be drilled as the optimization parameter. They also developed an approach to translate the uncertainty of data into monetary value uncertainties. They evaluated the methodology using a standard test case based on a real field known as the PUNQ-S3 model. The results were verified through exhaustive runs.

Özdoğan and Horne (2004) studied the correlation between time-dependent information and its effect on reduced uncertainty and increased Net Present Value. The researchers used a HGA as the optimization method, and a utility framework to determine optimum decisions for different risk attitudes. Their methodology incorporated time-dependent production history as the wells are drilled into the placement decisions, allowing not only maximum oil production, but improving future drilling by including prior information. The paper came to several conclusions providing evidence to the benefits of using their approach.

Yeten, Durlofsky, and Aziz (2003) use a genetic algorithm in combination with acceleration routines such as an artificial neural network, a hill climber, and a near-well upscaling technique to determine the optimal type, location, and trajectory of nonconventional wells. A significant advantage of this study is its ability to optimize the well type as well as other relevant well parameters, while maintaining populations containing a wide variety of wells. The methodology is applied to different oil-producing problems with varying reservoir and fluid properties. They were able to successfully increase the objective function, either cumulative oil produced or NPV, relative to its first generation value of the optimization. The optimum well type varied depending on the reservoir type, objective function, and degree of reservoir uncertainty.

OBJECTIVES

The objective of this research is three-fold. First, an optimization methodology must be developed, which can be coupled with a commercial simulator to make an exhaustive run on a gas reservoir model to determine optimal well placement based on cumulative gas produced. Second, the conventional exhaustive run output will be analyzed to determine how much well placement in gas reservoir matters, if at all. Finally, a simple genetic algorithm will be built to evaluate performance of the same gas reservoir to determine the benefits over the conventional simulation approach. The exhaustive run outcome is the basis to which the genetic algorithm will be compared.

CHAPTER II

THEORETICAL APPROACH

GENETIC ALGORITHM THEORY

Genetic Algorithms (GA) are search algorithms based on the mechanics of natural selection and natural genetics (Goldberg 1989) which use random choices as a tool to guide a highly exploitative search through a coding of a parameter space. The algorithm analyzes a population, represented by a series of strings, with a random, yet structured survival of the fittest method. Genetic Algorithms look for causal relationships between similarities of strings and high fitness. The benefits of GA include its ability to find global optima (while not getting stuck in local extrema), use of objective functions (as opposed to derivatives), parallelization abilities, and use of probabilistic transition rules (randomized operators). These benefits allow the genetic algorithm to be a robust, stochastic, and streamlined optimization method. Genetic Algorithms "efficiently exploit historical information to speculate on new search points with expected improved performance." (Goldberg 1989)

The GA population is represented by a series of strings. The string length is determined according to the range of parameters in the optimization problem. In this study, bit values (0 or 1) are used as the individuals that characterize each string.

Each string has an associated fitness value which serves as the basis for string-comparison during the optimization process. Strings with higher fitness values are held in higher regard than lower fitness strings. Ultimately, the string with the highest fitness value in the search space is chosen. Tournament selection is used to determine which strings to perform GA operations on. Basically, the highest-fitness-valued string is chosen from a selection of strings, and reproduced to the next population.

Genetic Algorithms search from a population of points, as opposed to the more traditional search of a single point. This feature enables the GA to be extremely robust in its search techniques because it allows for simultaneous modifications across the entire population. Therefore, it is impossible to get stuck in local extrema when searching for global optima. Searching from a population of points allows the crossover operators to combine optimal solutions from different areas to possibly create a better one, which is then introduced into the new population.

GENETIC ALGORITHM OPERATORS

An algorithm for a single generation of the GA operators is shown below in Figure 1, followed by a description of each operator's purpose.

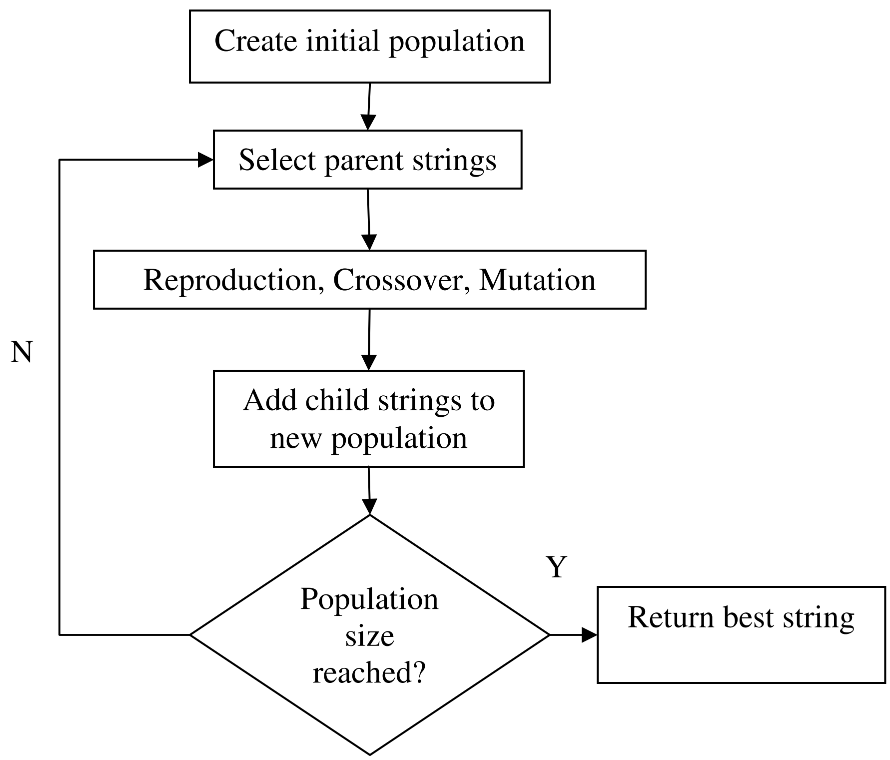

Fig. 1 – Algorithm for single generation of GA

A generation occurs after the population size has been reached, and the GA returns the best string.

Genetic Algorithms use three major operators to analyze a population of strings – reproduction, crossover, and mutation. Reproduction occurs when individual strings are copied to the next population according to their function or fitness value. Essentially, the best strings get more copies, the average stay even, and the worst die off. Reproduction alone does not sample new points in the search space; it simply carries over the highest fitness valued strings. The reproduction operator ensures the highest-fitness-valued string is never lost in the optimization process. The crossover operator allows the GA to exploit information by reproducing value strings according to their performance and crossing the strings with other high valued strings. The crossing point within the string is randomly selected, and is carried out with a probability of p_c. An example of the crossover operator being applied to two strings is seen below in Figure 2, with the "|" representing the crossover point.

Chromosome 1	**1101100 | 100110110**
Chromosome 2	**1100111 | 100001010**
Offspring 1	**1101100 | 100001010**
Offspring 2	**1100111 | 100110110**

Fig. 2 – Example of crossover operator

The mutation operator changes an individual bit value within the string from a zero to one or vice versa, and is applied after the crossover operator with a probability of p_m. Figure 3 is an example of the mutation operator.

Offspring 1	110**1**10010**0**001010**0**
Offspring 1a	110**0**10010**1**001011**1**

Fig. 3 – Example of mutation operator

The purpose of the mutation operator is to sporadically introduce new strings into the search space by slightly changing the string values in the current population.

SCHEMATA

Similarities between highly-fit strings guide the search of the GA, so it is of importance to understand how strings can be similar. The notion of schemata was developed to enhance understanding of string similarity. "A schema is a similarity template describing a subset of strings with similarities at certain string positions." (Goldberg 1989) If an alphabet of {0,1,*} is considered, with the '*' being able to take the form of either 0 or 1, then a schema matches a string whenever a 0,1, or * are present at the same location in both the string and schema. An example from Goldberg's book (1989) on genetic algorithms is considered: The schema *0000 matches two strings, {10000, 00000}. Also, the '*' is not actually processed by the GA, but is used only to help define the schema concept.

Introducing the '*' character greatly enhances the total number of schemata in the search space. Considering the example above, the number of schemata increases from $2^5 = 32$ (0,1) to $3^5 = 243$ (0,1,*). Intuition deduces schemata increase the difficulty of the search optimization, but that is not the case in reality. To prove, we need to consider the total number of schemata in the population. When using the binary alphabet, a string contains 2^l schemata. Therefore, an n-sized population contain between 2^l and $n \cdot 2^l$ schemata. The GA operators must be considered when determining how many schemata are processed by the GA. Reproduction affects a schema by giving higher probabilities of selection to high-fit strings. Depending on the defining length of a schema, crossover may or may not cut the schema. Again, another example from Goldberg's (1989) book is used: the schema 1***0 is likely to be cut during the crossover operation because it has a large defining length (several spaces between the two known values), while the schema **11* is unlikely (although possible) to be cut due to a short defining length. Mutation can be disregarded due to its infrequent use (based on normal, small mutation probability value). Therefore, "highly fit, short-defining-length schemata are propagated generation to generation by giving exponentially increasing samples to the observed best." (Goldberg 1989)

THE SCHEMA THEOREM

Adding to the schemata discussion above, order and defining length are two important schema properties. Order is the number of fixed positions in the schemata. For example, the schema 10**1* has an order o(10***1*) equal to 3. The defining length of a schema refers to the distance between the first and last fixed position in the schemata. Therefore, the schema 10**1* has a defining length δ(10**1*) equal to 4. Schemata and their properties "provide the basic means for analyzing the net effect of reproduction and genetic operators on building blocks contained within the population." (Goldberg 1989)

Reproduction affects the number of schemata in the population by requiring above-average schemata to increase, and below-average schemata to die off. Reproduction copies string strictly based on their fitness values. Let $m(\mathrm{H},t)$ be the examples of a schema H within the population $\mathbf{A}$(t). A string, at generation i, gets selected with probability $p_i = f_i / \Sigma\, f_j$. At generation $t + 1$ we have

$$m(H,t+1) = m(H,t)\frac{f(H,t)}{\overline{f}}. \tag{2.1}$$

where $f(\mathrm{H},t)$ represents the average fitness of the strings of schema H at time t. This equation shows all schemata in a population grow or decay based on their averages strictly under reproduction in parallel.

Assuming schema H remains above average an amount c $\overline{f}$, with c a constant, then the Eq. 2.1 can be rewritten as

$$m(H,t+1) = m(H,t)\frac{\overline{f}+c\overline{f}}{\overline{f}} = (1+c)\cdot m(H,t) \tag{2.2}$$

At t = 0, with c staying constant

$$m(H,t) = m(H,0) \cdot (1+c)^t \tag{2.3}$$

This equation shows reproduction acts on schemata in an exponential manner, meaning reproduction "allocates increasing (decreasing) numbers of trials to above- (below-) average schemata" (Goldberg 1989).

Reproduction alone does not sample new points within the population. The crossover and mutation operators are introduced for new sampling. Crossover is a random, but structured information exchange between two strings that introduces new sampling points in the population. Crossover decreases the schema survival probability from a high of $p_s(H) = 1.0$ to

$$p_s = 1 - p_c \frac{\delta(H)}{l-1}, \tag{2.4}$$

where l represents the string length, p_s represents the survival probability, and p_c represents the crossover probability. This is important because it disrupts schemata growth.

Similar to crossover, mutation negatively effects schemata growth. The mutation operator is not used as extensively as reproduction or crossover, and its sole purpose is to introduce variety into the population. The survival probability for mutation is given by

$$p_s(H) = 1 - p_m o(H), \tag{2.5}$$

where p_m represents the mutation probability. Taking into account all three GA operators, the true schema growth equation is given by:

$$m(H,t+1) = m(H,t)\frac{f(H)}{\overline{f}}\left[1 - p_c \cdot \frac{\delta(H)}{l-1} - o(H)p_m\right]. \tag{2.6}$$

The above equation is known as the Fundamental Theorem of Genetic Algorithms, or simply the Schema Theorem. It states short, low-order, above-average schemata receive exponentially increasing trials in subsequent generations.

PARALLELISM

Implicit Parallelism: Another benefit of genetic algorithms is the concept of *implicit parallelism*, which states that despite processing only *n* structures per generation, the genetic algorithm processes roughly n^3 schemata (Goldberg 1989). This means the genetic algorithm is able to process many more schemata than computationally proportional to the population with no extra work to the computer's memory.

Explicit Parallelism: Genetic algorithms are also attractive due to their simplicity in regards to explicit parallelism. When the problem is complex enough, making explicit parallelism necessary, genetic algorithms are relatively easy to set up in parallel to simultaneously run on different CPU's.

STOPPING CRITERIA

Different problems require different stopping criterion depending on the application. Several options exist when deciding how to stop the GA run, and they are listed below.

1. The GA stops by setting a maximum generation limit in the inputs. This means the GA will stop at the maximum generation even if it has not yet reached the maximum fitness value (method used in this study).
2. Genetic algorithms can be stopped after reaching some time limit. Again, the GA theoretically can stop without determining the maximum fitness value.
3. The algorithm stops when some fitness limit has been reached. This requires prior knowledge or determination of the maximum fitness value for the population.
4. The algorithm stops if there is no improvement in the fitness value during some interval of time. Similar to methods 1 and 2 above, this method theoretically allows the GA to stop without reaching the maximum fitness value.
5. The percentage change between fitness values over several generations reaches some minimum. This is a difficult option to implement because the maximum fitness values (per generation) over several generations can be equal to each other without being equal to the true global maximum fitness value. Under this condition, the GA would stop without finding the true maximum fitness value within the population.

IMPLEMENTATION OF EXHAUSTIVE SEARCH

This research uses a simple genetic algorithm to approach the well placement optimization problem. Specifically, horizontal wells in gas reservoirs are of main interest.

Before the theory for implementation of a genetic algorithm into the optimization problem is explained, a basis must be formed to compare with the genetic algorithm output. The most thorough, yet time-consuming and costly, approach is that of an exhaustive run. First, a grid, representing the reservoir, must be built. The exhaustive approach refers to placing a single well in a grid block and analyzing the reservoir output through simulation. This process is repeated for every grid block in the reservoir. For example, a 16 X 32 grid would require 512 simulation runs. Also, for this approach to be even remotely feasible, an automation process must be integrated.

In this study, a system of a reservoir with a horizontal well is studied. The reservoir is grid as a 16 X 32 grid in areal direction. There are 16 grid blocks in the x-direction for a total of 1867 ft., and 32 grid blocks in the y-direction for a total of 3734 ft. The reservoir thickness is 20 ft. The lateral of the horizontal lies in the middle of the reservoir and is 750 ft in length, or 8 grid blocks. The drainage area is 320 acres. Due to the horizontal well length, restrictions must be placed on where the well can be drilled and in what lateral direction. This research only considers lateral directions in the +x, -

x, +y, and –y directions. Refer to Figures 4 and 5 to see which locations in the grid are valid drilling locations dependent upon the lateral direction. For example, a horizontal well with a lateral orientation in the –x direction cannot be drilled in columns 1-7 of the reservoir grid because the horizontal well has a lateral length of eight grid blocks; it is valid in columns 8-16 because there are at least 8 available grid block locations for the well to be drilled in the –x direction. This thought process is analogous for the +x, +y, and –y lateral orientations for the wells. The red outline in Figure 4 represents the valid locations for +x direction wells. The blue, shaded area in Figure 4 represents the valid locations for –x direction wells.

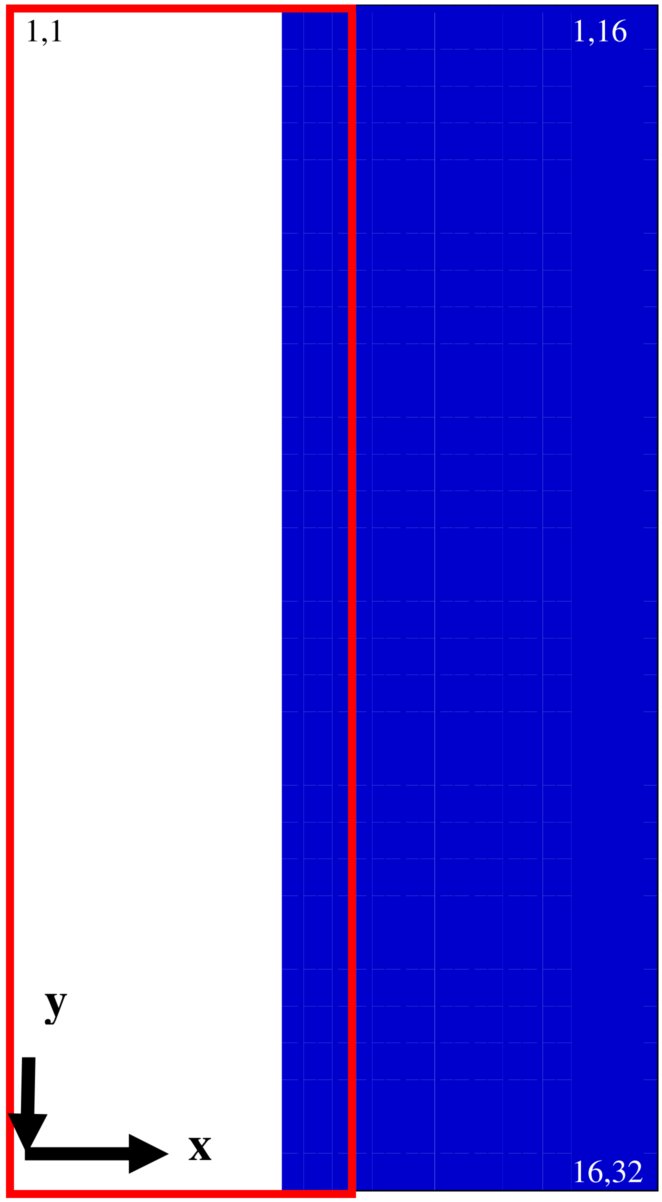

Fig. 4 – Valid well placement locations for lateral orientations in ±x directions

The red outline in Figure 5 represents the valid locations for +y direction wells. The blue, shaded area in Figure 5 represents the valid locations for –y direction wells.

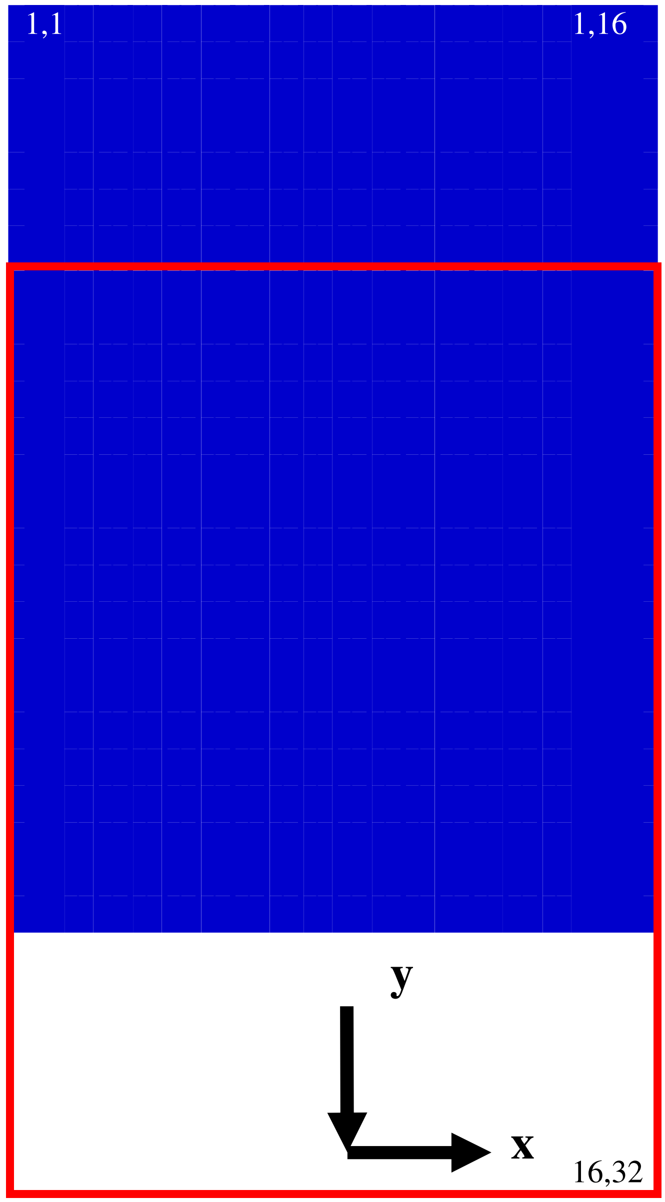

Fig. 5 – Valid well placement locations for lateral orientations in ±y directions

Since the handlings of all four lateral directions are analogous, the positive-x lateral direction will be used to discuss the methodology of how to automate the exhaustive method.

In the 16 X 32 grid, representing reservoir model, the surface location for the positive x-direction lateral can be drilled for the x-value range between 1 and 9, and for the entire y-value range of 1 through 32 (Figure 4). The first grid block drilled of the lateral, out of the 8 total, represents the surface location of the well. The reservoir model data file is opened and read until a location trigger is found. The location trigger is a variable in the reservoir model data file that locates the x, y, and z grid block values of the lateral. Next, the system calls the newly written data file and proceeds to call an executable file from the commercial simulator; in this case, CMG. A simulation is carried out with the current lateral position within the grid, and an output file is created. The fitness value, cumulative gas produced, is read from the output file and written to another file (along with the surface location of the well) for later analysis. This process is repeated for every valid grid block for positive x-direction lateral wells. The process is then repeated for all other lateral directions with the necessary changes to the x- or y-values of the lateral. An algorithm for a single run of the conventional exhaustive approach is shown in Figure 6.

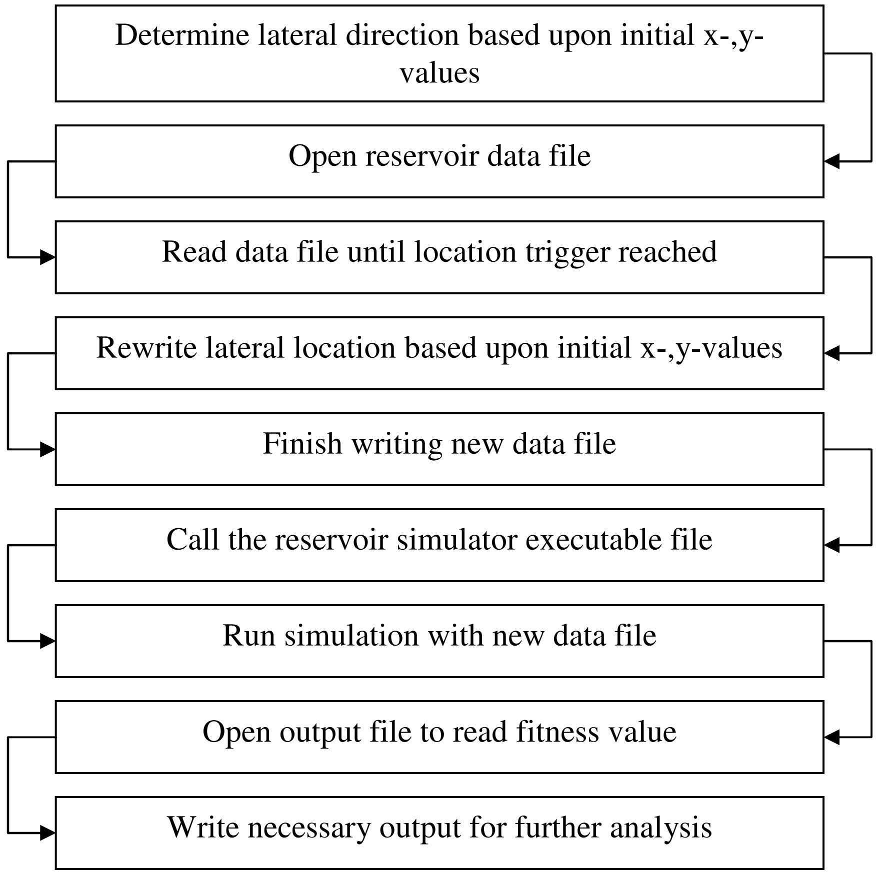

Fig. 6 – Algorithm for exhaustive search method

WELL-INDEXING AND BINARY ENCODING

To explain how the implementation of a genetic algorithm into the optimization problem works, well-indexing and binary encoding must first be understood. Similar to the exhaustive method above, a grid system must be built to represent the reservoir. As previously stated, the reservoir model will be represented as a 16 X 32 grid system with each grid active for well placement.

For the genetic algorithm to work properly with a vertical well model, each block is represented by a 9-bit string (due to binary encoding). The first four bits represent the x-value (values range from 1-16) of the grid, the following five bits represent the y-value (values range from 1-32). See Figure 7 for an example of the string representation for each gridblock, discounting the last two binary values, which are used in the horizontal well problem.

For the genetic algorithm to work properly with a horizontal well model, each block is represented by an 11-bit string due to binary encoding. The first nine bits are represented similar to the vertical well case described above, and the final two bits are added to the end of the 9-bit string to represent the lateral direction of the horizontal well. The lateral direction values range from 0-3, with 0 (00 in binary) representing the positive x-direction, 1 (01 in binary) representing the negative x-direction, 2 (10 in binary) representing the positive y-direction, and 3 (11 in binary) representing the negative y-direction. Figure 7 shows the well indexing for the top-left quarter of the grid, assuming a positive x-direction lateral. The grey-shaded cell shows how the binary values for the x- and y- values are combined with the binary value for the lateral direction to form an 11-bit string that is representative of the grid location and lateral direction. Please note that only the last two digits (bold, red) in the grey-shaded cell would change if the lateral direction was to be –y instead of +x (would change to 10).

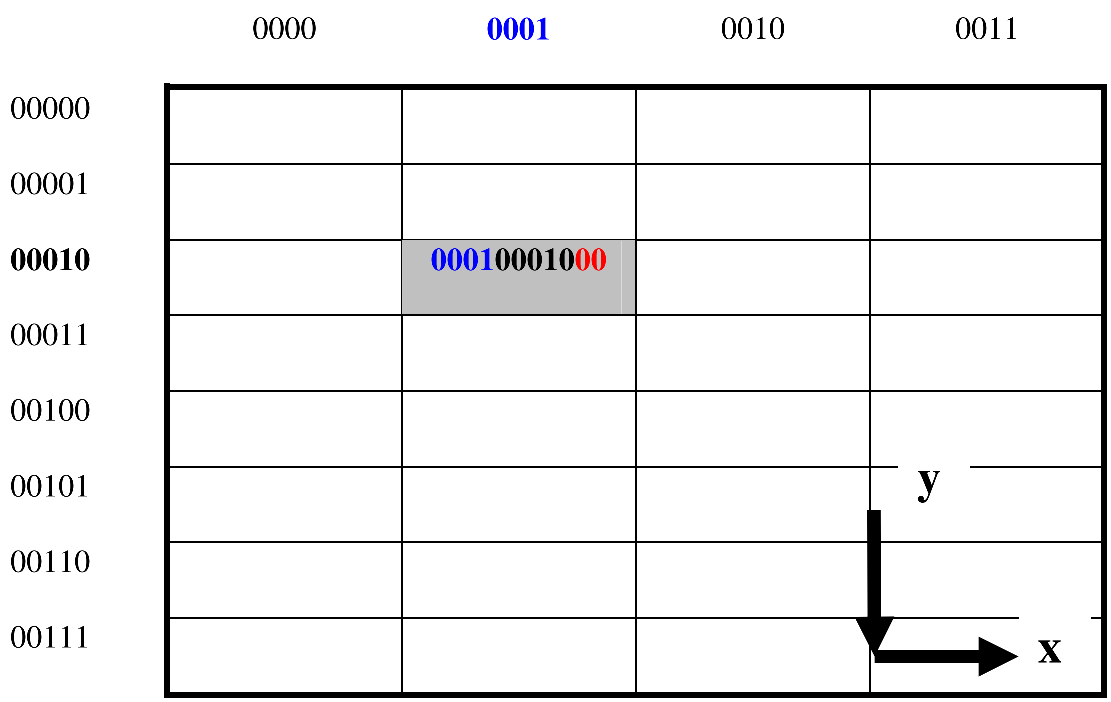

Fig. 7 – Well-indexing for reservoir grid example

The population size analyzed by the genetic algorithm at any given point is equal to the number of bits in the string. For example, in the three-parameter, 11-bit case from above, would result in an evaluation of eleven strings during each run.

The mutation probability is equal to the inverse of the population size. Therefore,

$$p_m = \frac{1}{Npop}, \tag{2.7}$$

where p_m is the mutation probability and npop represents the population size of a GA run. The equation allows for a consistently small mutation probability, as desired, no matter the population size.

IMPLEMENTATION OF GENETIC ALGORITHM

Genetic Algorithms are extremely flexible in the problems they can solve, and for the most part, only require a change in the function value subroutine when solving a new problem. This research builds off a simple genetic algorithm code (Carroll 2009*)* as a platform to handle the horizontal well optimization problem.

The function value for the genetic algorithm code is very intricate, but similar to the exhaustive run approach in many ways. First, the x-value, y-value, and lateral direction must be set equal to parent 1, 2, and 3 in that order. This initialization allows the rest of the genetic algorithm code to identify the location and lateral direction in a way it understands.

The reservoir grid represents the total search area for the problem. The initial population of strings is randomly chosen and based upon the grid location and binary encoding of the specific gridblock. The genetic algorithm randomly selects parent strings from the initial population to analyze. The parent strings are sent to the function value subroutine where they are assigned a fitness value, in this study, the cumulative gas produced. To assign the fitness value, a commercial simulator, CMG, is called, and the simulation is carried out based upon the reservoir properties and grid location(s) of the well. The commercial simulator produces a fitness value that is associated with the parent string, and the two are sent back to the main processing loop of the genetic algorithm. Next, the genetic algorithm operators act upon the parent strings and associated fitness values. The highest fitness-valued string(s) is reproduced, while the other parent strings are randomly selected to go through the crossover and mutation operators in search of a combination of bit values yielding a higher fitness value than the previous best. The string with the highest fitness value in the generation is written to an outside file for further analysis. A generation is an iteration of the genetic algorithm; the total number of generations is specified by the initial inputs. Refer back to Figure 1 for a review of genetic algorithm generations. This process is repeated until the stopping criterion is met.

The fitness value is determined similar to the exhaustive run approach from above, based upon the initial location values and lateral orientation of the horizontal well. The difference is the location values and lateral orientations are provided by the genetic algorithm instead of in a linear fashion similar to an exhaustive approach. If an invalid location value and lateral orientation combination is provided by the genetic algorithm, the function value for the string is set to zero since it is physically impossible to achieve. For example, a lateral direction in the negative x-direction at x-value = y-value = 1 would return a zero-valued function value.

Finally, the location values, lateral orientation, and associated fitness values are written to a file for further analysis.

CHAPTER III

RESULTS AND DISCUSSIONS

GA INPUT VALUES

This research uses a maximum generation value of 200 generations as the stopping criterion, even if the maximum fitness value is found beforehand, for several reasons. First, an analysis of the implementation of the GA is needed to confirm it is working properly. There would have been a fundamental mistake in the implementation of the GA if the GA returned a value higher than the largest fitness value from the exhaustive search. Allowing the GA to continue running after the maximum fitness value (determined from the exhaustive search) had been reached, helps to confirm the GA is working properly. The researchers also wanted to analyze the genetic algorithms tendencies after the fitness value had been reached. Again, making sure the maximum fitness value at each generation did not increase or decrease after the true global optimum was reached is of main concern. Also, an analysis of the generation number versus maximum fitness function value per generation was conducted to see if any trends emerged.

A crossover probability, p_c, of 0.6 is implemented in all cases. The value was determined after researching different applications of genetic algorithms (Goldberg 1989 and Gűyagűler 2002).

RESERVOIR DESCRIPTION

The case-specific values for several reservoir properties are shown in Table 1. All of the horizontal well cases take the lateral orientation of the horizontal into consideration. As stated in the explanation of the exhaustive search, the possible lateral orientation values are the +x, -x, +y, and –y directions when viewing the reservoir from directly above.

This research makes use of a single-phase gas model for every simulation. The reservoir is at irreducible water saturation and believed to be a volumetric reservoir with no water drive. The reservoir temperature is at 275 deg F. The gas gravity, relative density to air, is 0.68, and the permeability of the gas reservoir is 0.1 md.

Case-specific differences are discussed below under the respective case headings. The first four cases were carried out for a period of 10 years. All cumulative gas production values are based upon a 10 year production period for the first four cases. Case 5 was only produced for 1 year to determine the time effects on optimization. Case 5 is the best representation of a real-world example because it starts to take economic value on a time scale into account.

Table 1 - Reservoir Properties for Each Case

Case Specific Reservoir Values

				Permeability (md)				
	Grid Top (ft)	**Grid Thickness (ft)**	**Porosity**	**i-vale**	**j-value**	**k-value**	**Pressure (psi)**	**S_w**
Case 1	2,000	50	0.2	10	10	0.1	1,000	0.2
Case 2	12,000	20	0.12	0.1	0.1	0.05	9,920	0.3
Case 3	12,000	Same as Case 2	0.11889+0.02277*LOG(k)	Rand()	Rand()	Rand()*0.5	9,920	0.3
Case 4	12,000	Same as Case 2	0.11889+0.02277*LOG(k)	Rand()	Rand()*0.3	Rand()*0.5	9,920	0.3
Case 4	12,000	Same as Case 2	0.11889+0.02277*LOG(k)	Rand()*0.1	Rand()*0.3	Rand()*0.5	9,920	0.3

The RAND() permeability values for Cases 3 and 4 are discussed in the text under 'Results and Discussion' for Case 3

Case 1: Homogeneous gas reservoir (vertical well)
Case 2: Homogeneous gas reservoir (horizontal well)
Case 3: Isotropic gas reservoir (horizontal well)
Case 4: Anisotropic gas reservoir (horizontal well)
Case 5: Anisotropic gas reservoir (horizontal well)

CASE 1

The first case is a single-layer, homogeneous gas reservoir. Case 1 has different reservoir properties than the other three cases for several reasons. This case represents the most simplistic case used in regards to a gas reservoir. Also, this research is not concerned with the benefits on cumulative production of a vertical well as opposed to a horizontal well, or vice versa, so there is no need to run a vertical and horizontal well on the same reservoir model. Case 1 is implemented solely for the purpose of checking the genetic algorithms ability to perform its job. After the GA was implemented successfully in Case 1, more computationally-intensive cases were considered.

The vertical well for Case 1 has two constraints that affect production. A maximum standard condition for stock tank surface gas rate of 1,000 Mcf/day, and minimum bottom hole pressure of 800 psi are implemented as well constraints.

CASE 2

The second case is a single-layer, anisotropic, homogeneous gas reservoir for a single horizontal well. Isotropy occurs only in the horizontal direction; therefore, k_x and k_y are equal to each other, but not equal to k_z. The lateral for the horizontal well is placed in the middle of the zone.

CASE 3

The third case is similar to case two except for the porosity and permeability values. Case 3 represents an anisotropic, heterogeneous reservoir. Isotropy only occurs in the horizontal, x- and y- directions. Therefore, k_x and k_y are equal to each other, but not equal to k_z. The permeability field was established by a random distribution of varying permeability ranges throughout the reservoir. Table 2 shows the permeability ranges, in millidarcies, for different sections in the reservoir. The reservoir grid is 16X32 gridblocks, and each section represents an 8X8 section of the reservoir grid.

Table 2 – Permeability Ranges for Different Sections in Case 3 and Case 4

0 ft 933.5 ft 1867 ft

1-10(md)	0.1-1(md)	933.5 ft
10-30(md)	1-10(md)	1867 ft
1-10(md)	0.1-1(md)	2800.5 ft
0.1-1(md)	0.1-1(md)	3734 ft

The permeability values range from 0.1 millidarcy to 30 millidarcy. Figure 8 displays a representation of the permeability ranges. The permeabilities in the x- and y- directions are equal for case three.

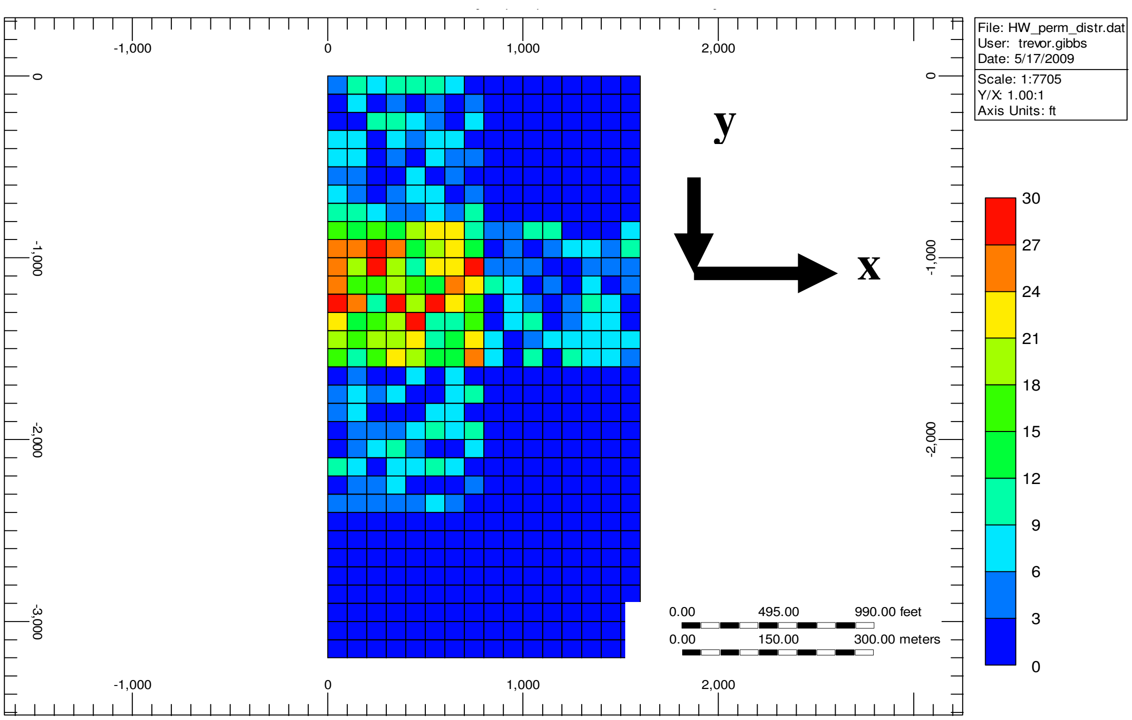

Fig. 8 – Horizontal permeability field for Case 3 and 4

The porosity values for each grid block are related to the permeability values by the following logarithmic equation (Gűyagűler 2002).

$$\phi_i = 0.11889 + 0.02277\log(k_i)$$

The porosity field is shown in Figure 9.

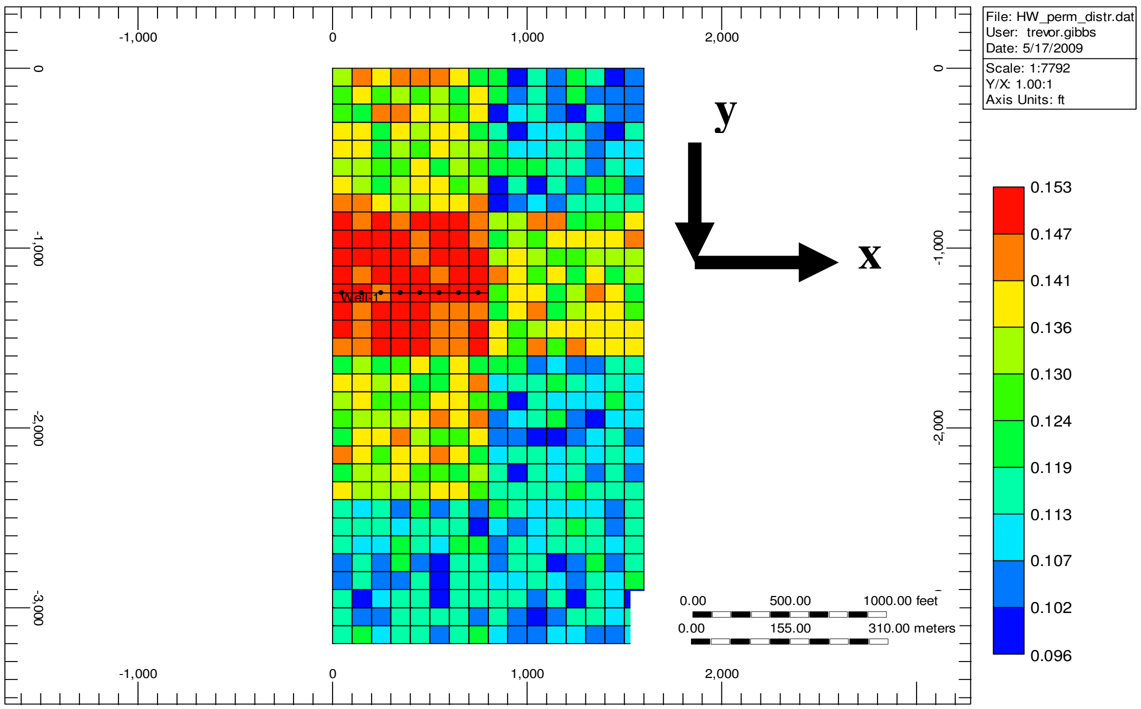

Fig. 9 - Horizontal porosity field for Case 3 and 4

The horizontal well for Case 3 is similar to Case 2 in regards to the well constraints. Case 3 has a maximum production for stock tank surface gas rate of ten million cubic feet per day, and minimum value of one-hundred thousand cubic feet per day. The last constraint is a minimum bottom hole pressure of 1500 pounds per square inch.

CASE 4

The fourth case is similar to the third case in every way except for the permeability in the y-direction. In Case 4, $k_j = 0.3k_x$. Therefore, k_x, k_y, and k_z are not equal to each other. The reservoir is an anisotropic (in every direction), heterogeneous reservoir. The well constraints for Case 4 are completely analogous to Case 2 and Case 3.

CASE 5

The fifth case is similar to case four except for the porosity and permeability values. Case 5 also shows results for only the first year of production. Case 5 represents an anisotropic, heterogeneous reservoir. Isotropy only occurs in the horizontal, x- and y-directions. Therefore, k_x and k_y are equal to each other, but not equal to k_z. The permeability field was established by a random distribution of varying permeability ranges throughout the reservoir. Table 3 shows the permeability ranges, in millidarcies, for different sections in the reservoir. The reservoir grid is 16X32 gridblocks, and each section represents an 8X8 section of the reservoir grid.

Table 3 – Permeability Ranges for Different Sections in Case 5

0 ft 933.5 ft 1867 ft

0.1-1(md)	0.01-0.1(md)	933.5 ft
1-3 (md)	0.1-1(md)	1867 ft
0.1-1(md)	0.01-0.1(md)	2800.5 ft
0.01-0.1(md)	0.01-0.1(md)	3734 ft

The permeability values range from 0.01 millidarcy to 3 millidarcy. The permeability and porosity fields look analogous to those of Cases 3 and 4, but with lower values. The permeability field for Case 5 is one-tenth of the values in Cases 3 and 4. The porosity values are correlated to the permeability values using the same equation as in Cases 3 and 4.

The horizontal well for Case 5 is similar to Case 4 in regards to the well constraints. Case 5 has a maximum production for stock tank surface gas rate of ten million cubic feet per day, and minimum value of one-hundred thousand cubic feet per day. The last constraint is a minimum bottom hole pressure of 1500 pounds per square inch.

OPTIMAL LOCATION ANALYSIS

METHODOLOGY

The optimum location for a producer was determined for each case through the use of an exhaustive search. The purpose of the exhaustive search is two-fold. First, the exhaustive search is used to evaluate the effectiveness and computational benefits of the GA. Second, the exhaustive search is used to determine if well placement in a gas reservoir produces a significant difference, positive or negative, in the cumulative gas produced value.

Every case produced more than one optimum location during the exhaustive search, meaning there is more than one grid which produced the optimal fitness value during the simulation. On a single well case, as in this research, any of the locations which produce the highest fitness value are considered the optimal case since it does not make any difference, on a cumulative-gas-produced basis, which location the producer is placed.

The genetic algorithm was applied to the each case-specific problem after the exhaustive run analysis was completed. The purpose for implementing the GA is to determine the computational benefits over the exhaustive search; specifically, how many runs it takes for the genetic algorithm to find the optimum fitness value for the producer in the reservoir. In every case, the genetic algorithm found at least one of the optimum fitness values from the genetic algorithm. Since this research only handles the production of a single well at any given moment, it was of no concern that the genetic algorithm did not find all of the optimal well locations; it only needed to find one.

The need for other local-optima-searching techniques to be coupled with the genetic algorithm is also verified in each case.

CASE 1

Case 1 requires 512 simulations for completion. The exhaustive run for the first case produces seven different locations for optimal placement. The grid blocks yielding the highest fitness values are (2,13), (2,14), (2,19), (2,20), (15,13), (15,14), and (15,19. The cumulative gas produced from each location is 310.44 MMSCF.

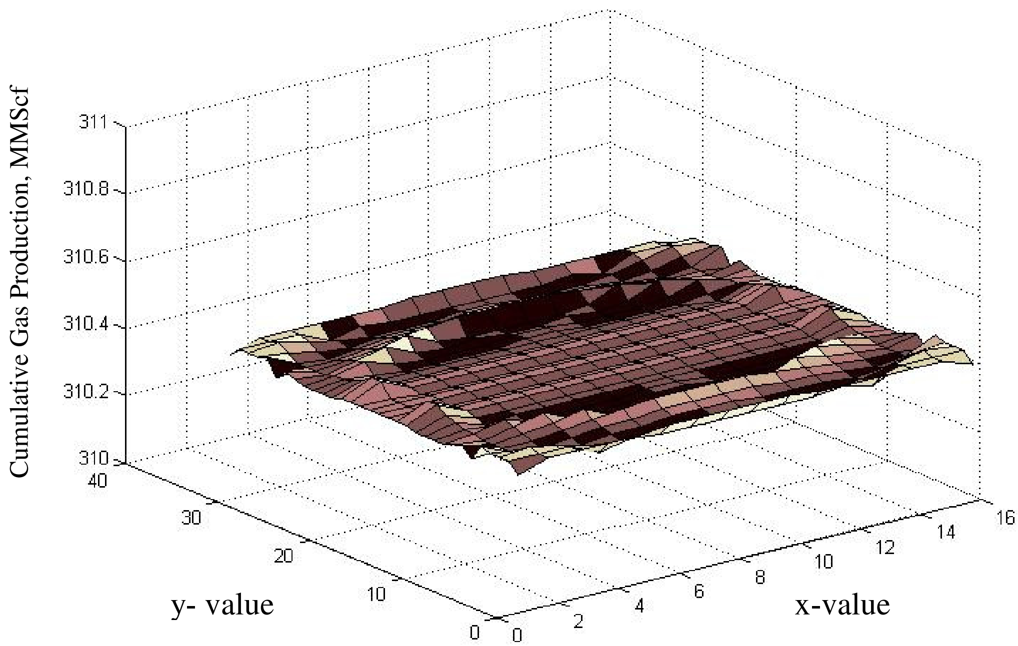

Fig. 10 – Case 1 fitness value vs. well location

Figure 10 above shows the fitness value versus well-location for the first case. The x and y axes represent the x- and y- grid values, and the z-axis represents the cumulative gas produced. The x, y, and z axes titles are the same for each case.

The lowest fitness value, which occurs in numerous grid blocks, is 310.38 MMSCF, which is a 0.0193 percent difference between the optimal fitness values of 310.44 MMSCF. The percent difference is most likely caused by numerical error, and is not a physical difference. Therefore, it makes no difference in where the well is drilled for Case 1 under current time frames and assumptions.

Case 1 has a total of 9 simulation runs per generation. The genetic algorithm found the optimum fitness value on the first run of the tenth generation for a total of 82 simulations; which is an 84.0% reduction in the required number of simulations it took to run the exhaustive search.

Figure 11 is a graph of the maximum function fitness value per generation for the first case.

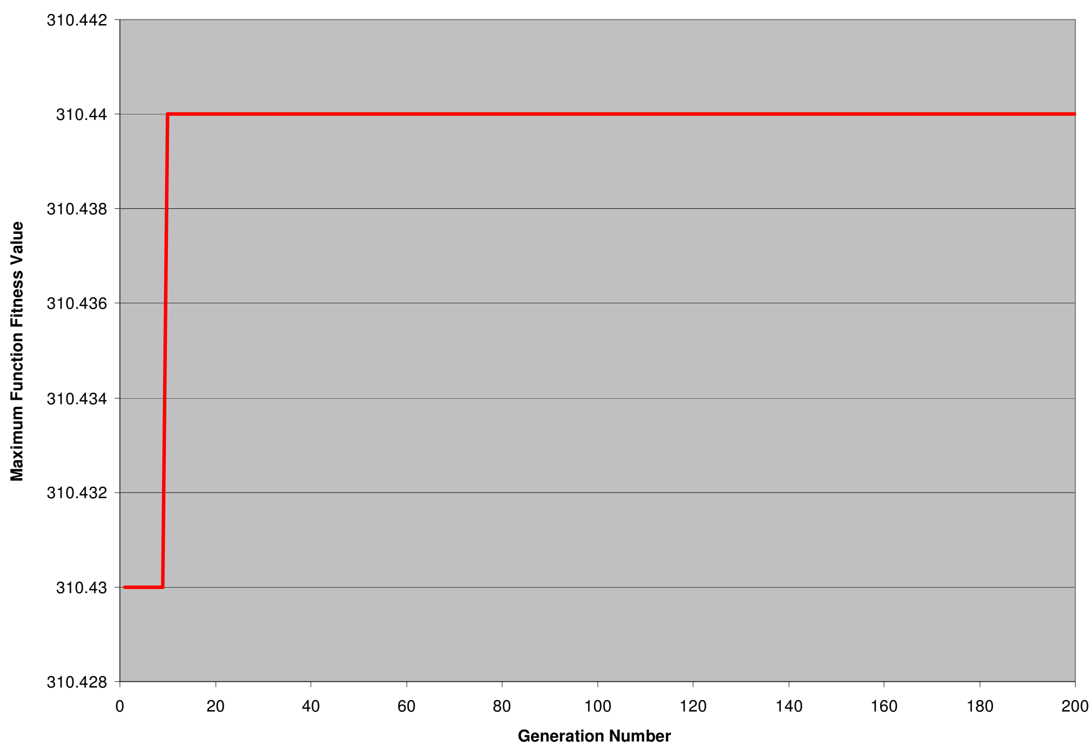

Fig. 11 – Maximum function fitness value vs generation number for Case 1

The maximum function fitness value for the first generation occurred at the first run at a value of 310.43, which is 99.99% of the maximum function fitness value for the entire reservoir. This is a good indication of the genetic algorithms ability to find the area of the global maxima, but lacks in its ability to find the global maxima. It took 81 more simulations to increase the fitness value 0.01% to the maximum fitness value.

CASE 2

The exhaustive run for the second case results in four different optimum well placements for a producer. The exhaustive run takes 1,376 simulations to analyze the entire reservoir grid. The number of simulations increases from 512 in the vertical well case because the lateral orientations of the horizontal wells are considered. The four optimum well locations are in grid blocks (6,19), (6,21), (11,19), (11,21), where the locations can either be the head or toe of the horizontal well.

Figure 12 shows the fitness value versus well-location for the second case.

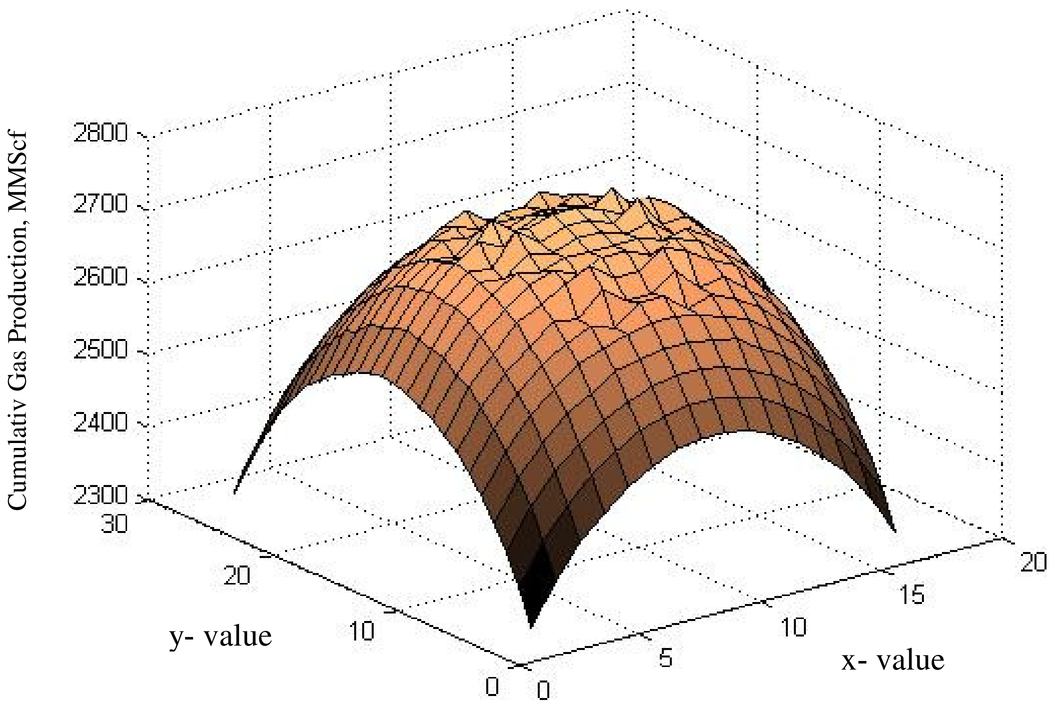

Fig. 12 – Case 2 fitness value vs. well location

The horizontal well for Case 2 has several well constraints, including a maximum standard condition for stock tank surface gas rate of 10 MMcf/day, and minimum value of 100 Mcf/day. The last constraint is a minimum bottom hole pressure of 1500 psi.

The optimum well locations can be seen in Figure 13 for Case 2.

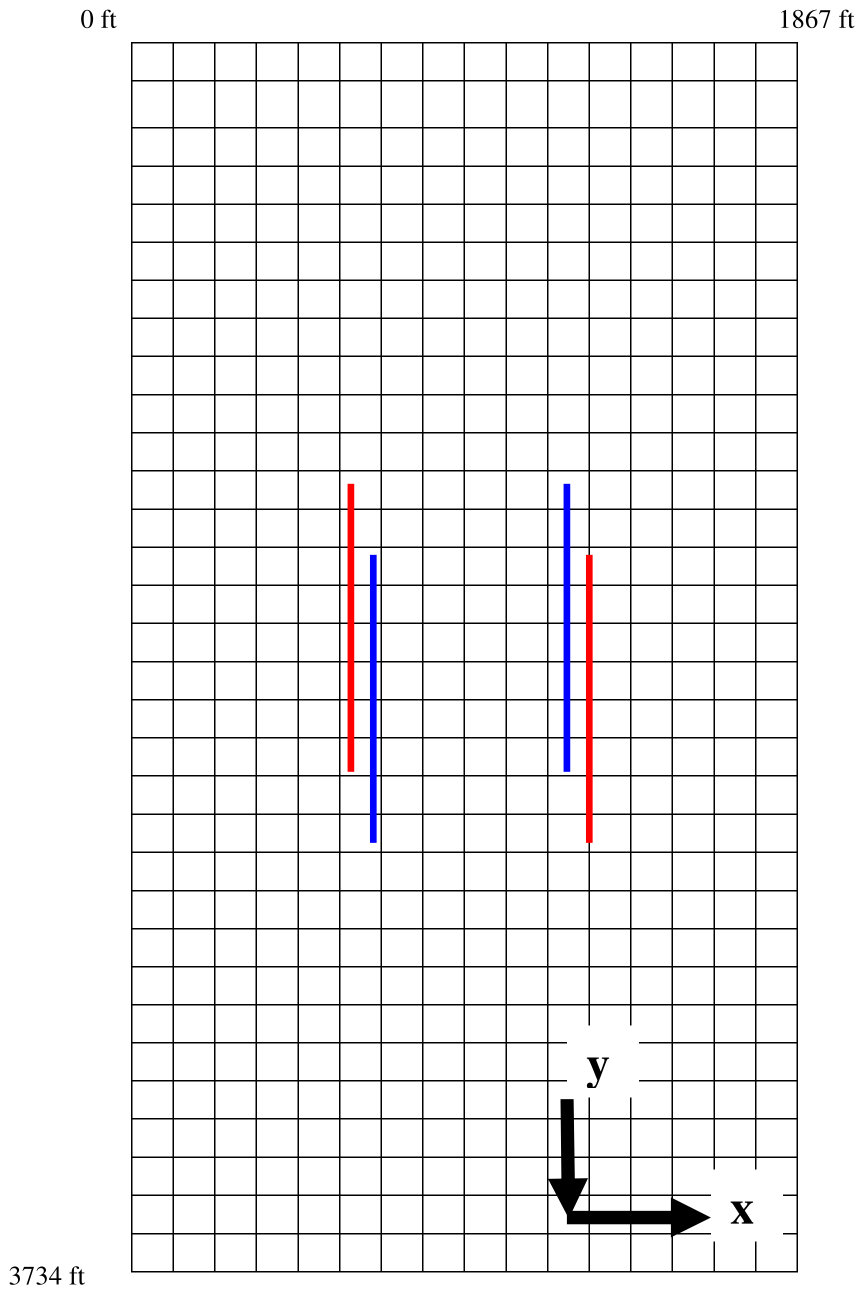

Fig. 13 – Optimal well locations for Case 2

The placement of the four wells in their respective grid blocks results in a cumulative gas produced value of 2753.4 MMSCF per well.

The lowest fitness value in the reservoir is 2304.0 MMSCF, which is a 16.32 percent difference between the optimal fitness values of 2753.4 MMSCF. The percent difference is significant; proving horizontal well placement in a homogeneous gas reservoir is of serious concern to a company, especially during times of high gas prices.

Case 2 has a total of 11 simulation runs per generation. The genetic algorithm found the optimum fitness value on the sixth run of the tenth generation for a total of 105 simulations. The reduction in simulations results in a 92.4 percent difference from the 1,376 simulations it took to run the exhaustive search.

Figure 14 is a graph of the maximum function fitness value per generation for the second case.

.

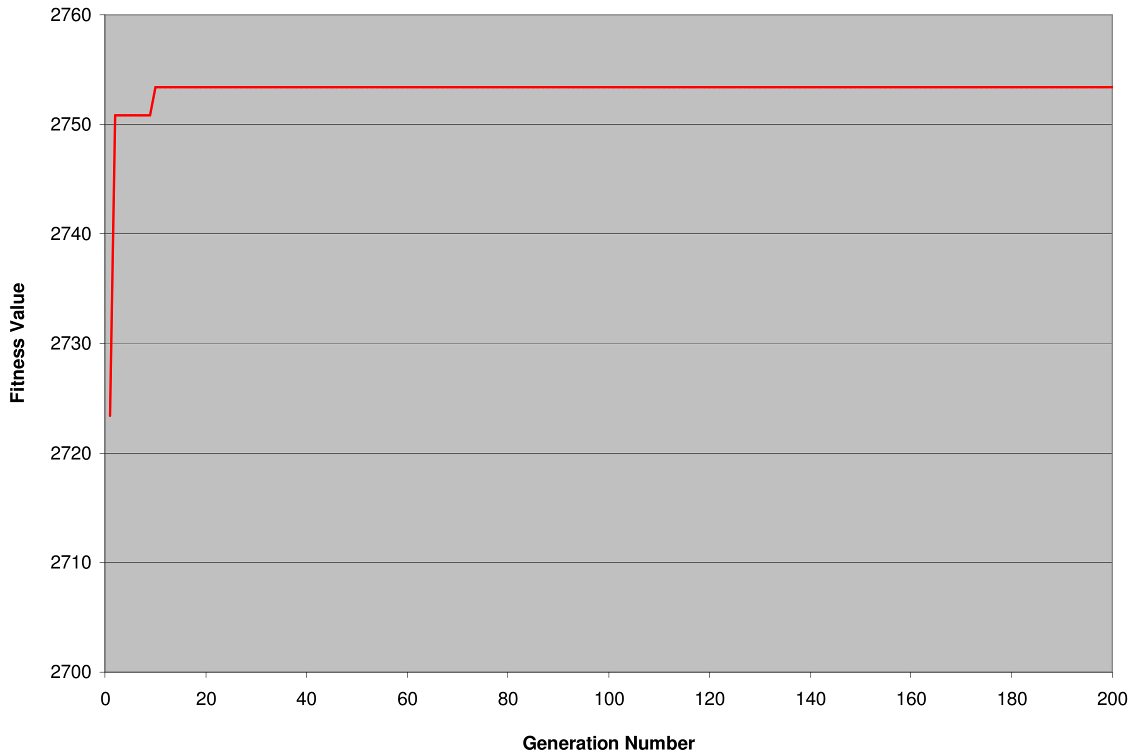

Fig. 14 – Maximum function fitness value vs generation number for Case 2

The maximum function fitness value for the first generation occurred at the ninth run at a value of 2723.4, which is 98.9% of the maximum function fitness value for the entire reservoir. Again, this verifies the genetic algorithms ability to find the area of the global maxima, but lacks in its ability to find the global maxima since it takes 96 more simulations to find the maximum fitness value of the entire reservoir.

CASE 3

The exhaustive run for case three results in one optimum well placement for a single producer well. The location is placed at grid block (4,27) in the reservoir, where the location can serve as either the head or toe of the horizontal well. The placement of the well in the grid blocks results in a cumulative gas produced value of 3323.4 MMSCF. Similar to case 2, the total number of simulations required for the exhaustive search is 1,376.

Figure 15 shows the fitness value versus well-location for the third case, and the well location is shown in Figure 16.

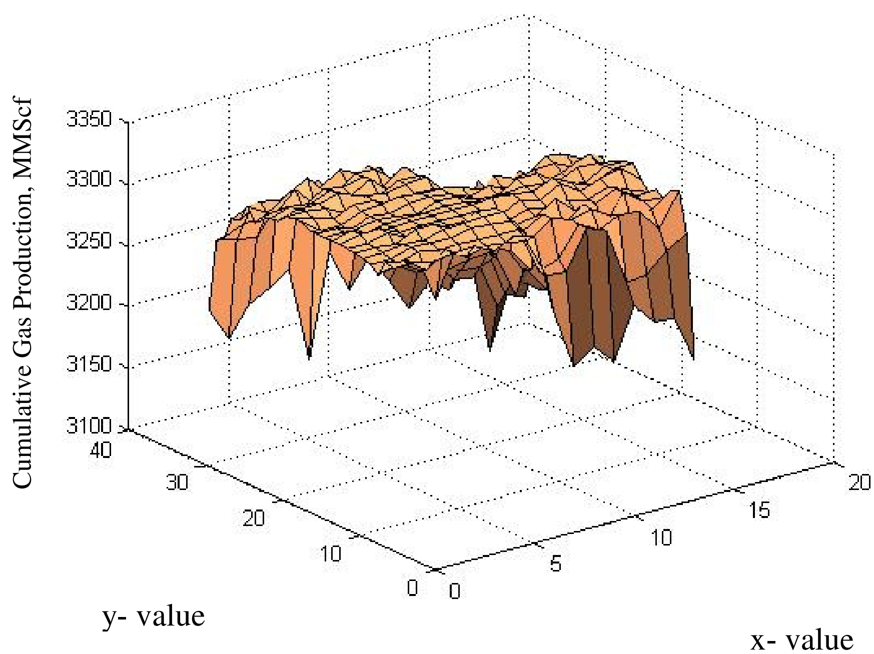

Fig. 15 – Case 3 fitness value vs. well location

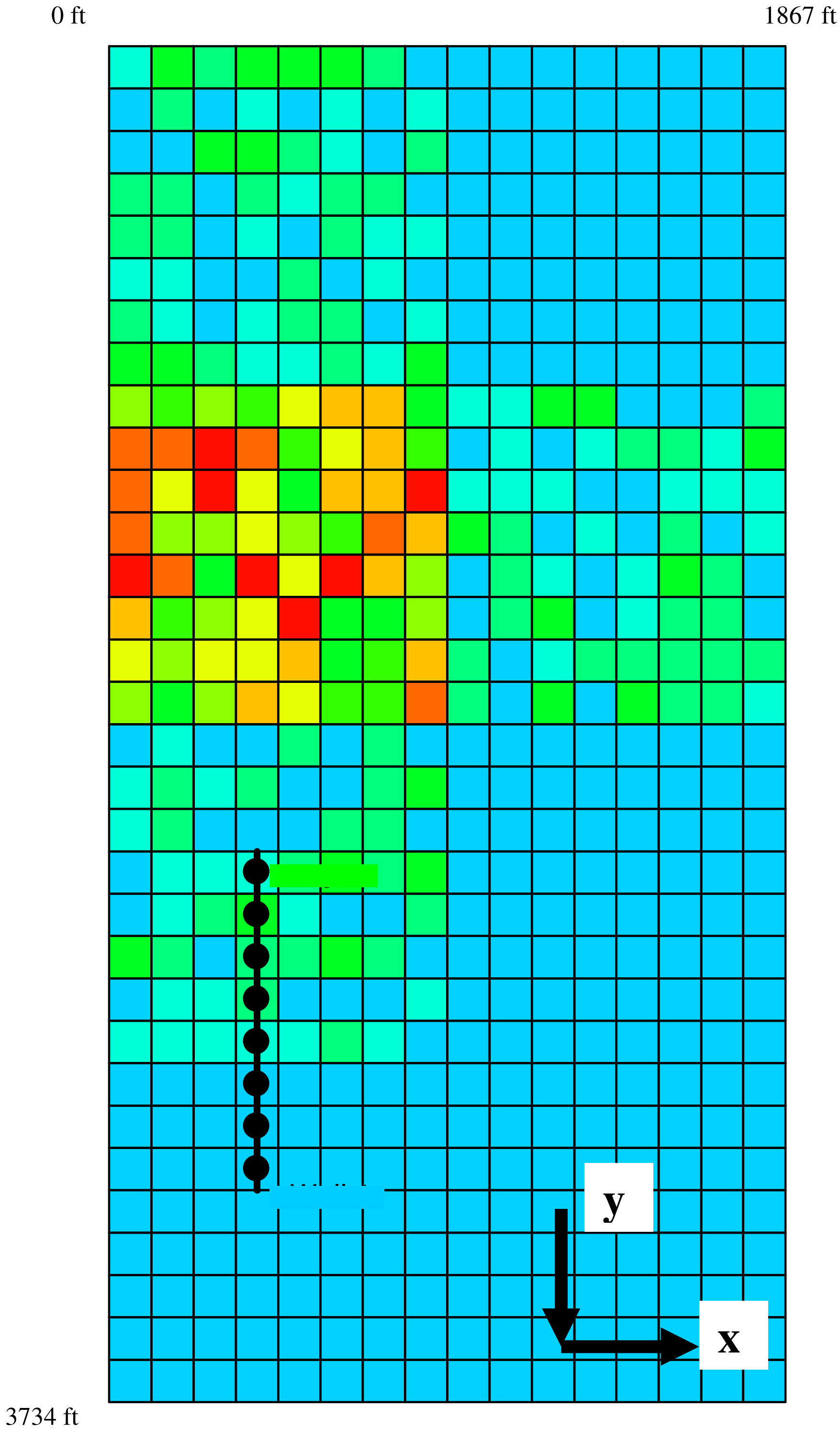

Fig. 16 – Optimal well locations for Case 3

The lowest fitness value in the reservoir is 3103.8 MMSCF, which is a 6.61 percent difference between the optimal fitness values of 3323.4 MMSCF. The percent difference is not nearly as high as the value in Case 2, but it is still significant enough to be of concern. Considering it should theoretically cost the same amount to drill and produce the well anywhere in the reservoir, a 6.61 percent increase in production at no extra costs is a wise business decision.

Case 3 has a total of 11 simulation runs per generation. The genetic algorithm found the optimum fitness value on the first run of the 46th generation for a total of 496 simulations. The reduction in simulations results in a 64.0 percent difference from the 1,376 simulations it took to run the exhaustive search.

Figure 17 is a graph of the maximum function fitness value per generation for the third case.

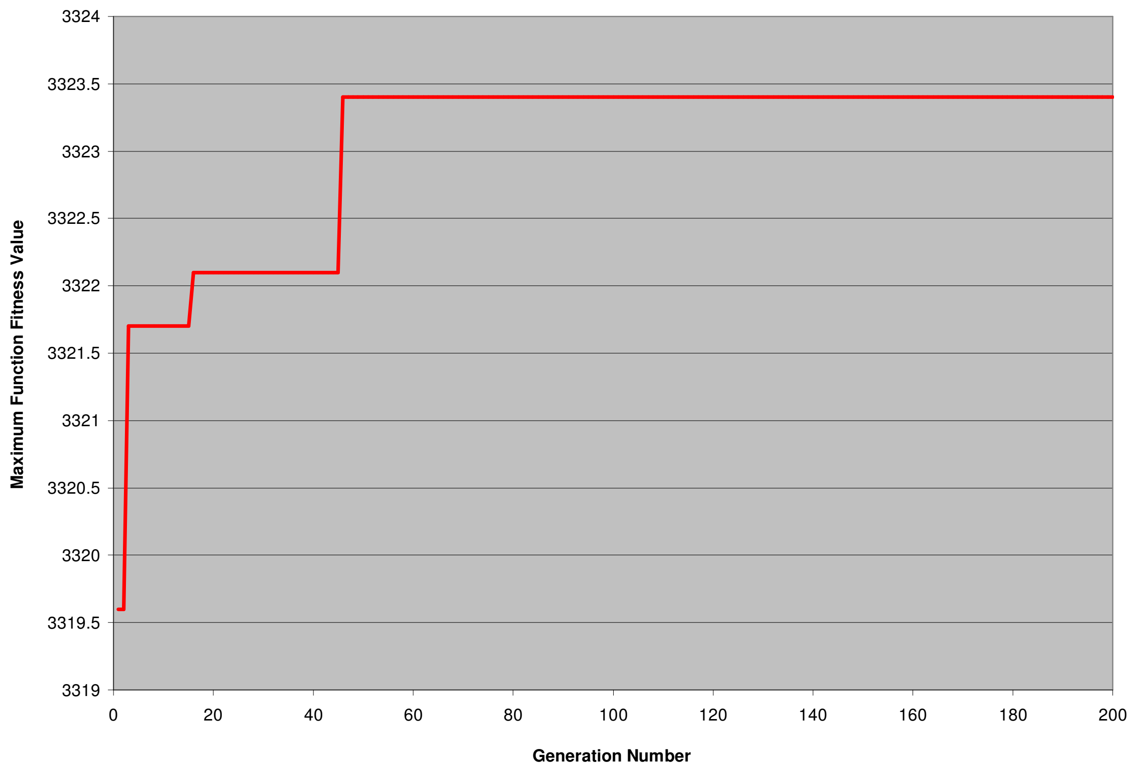

Fig. 17 – Maximum function fitness value vs generation number for Case 3

The maximum function fitness value for the first generation occurred at the eighth run at a value of 3319.6, which is 99.89% of the maximum function fitness value for the entire reservoir. It takes the genetic algorithm to 488 more simulations to find the maximum fitness value of the entire reservoir, which verifies the need for local-optima-searching techniques to be coupled with the genetic algorithm.

CASE 4

The exhaustive run for case four results in one optimum well placement for a single producer well. The location is placed at grid block (6,23) in which the location can act as either the head or toe of the well. The placement of the well results in a cumulative gas produced value of 3312.2 MMSCF. Similar to the previous two cases, the total number of simulations required for the exhaustive search is 1,376.

Figure 18 shows the fitness value versus well-location for the fourth case. The well location is shown in Figure 19.

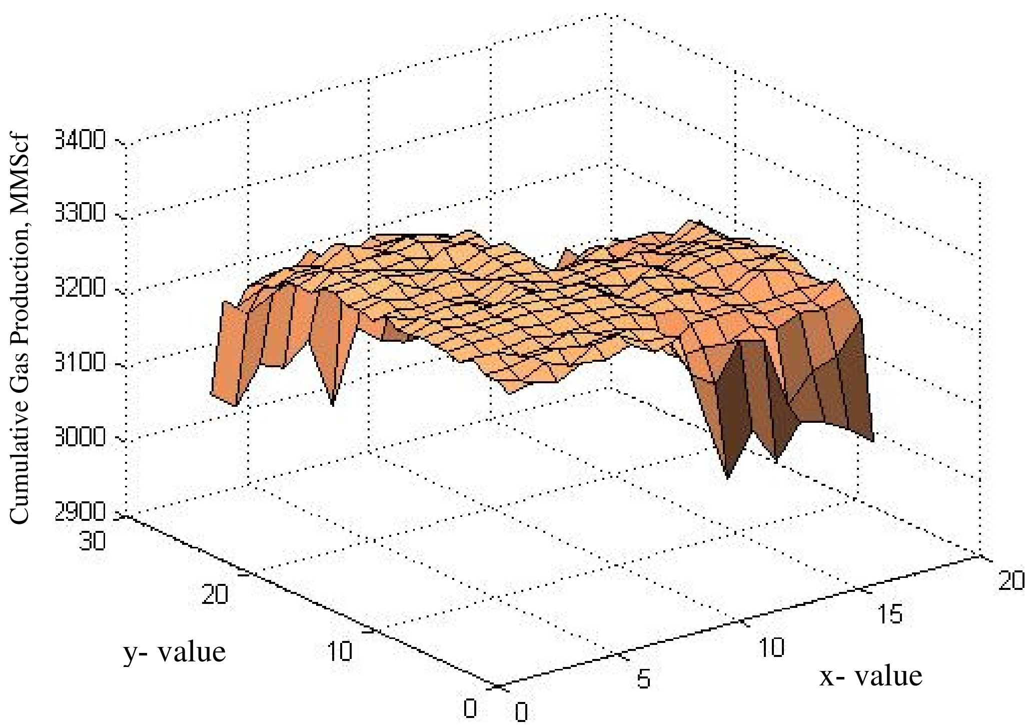

Fig. 18 – Case 4 fitness value vs. well location

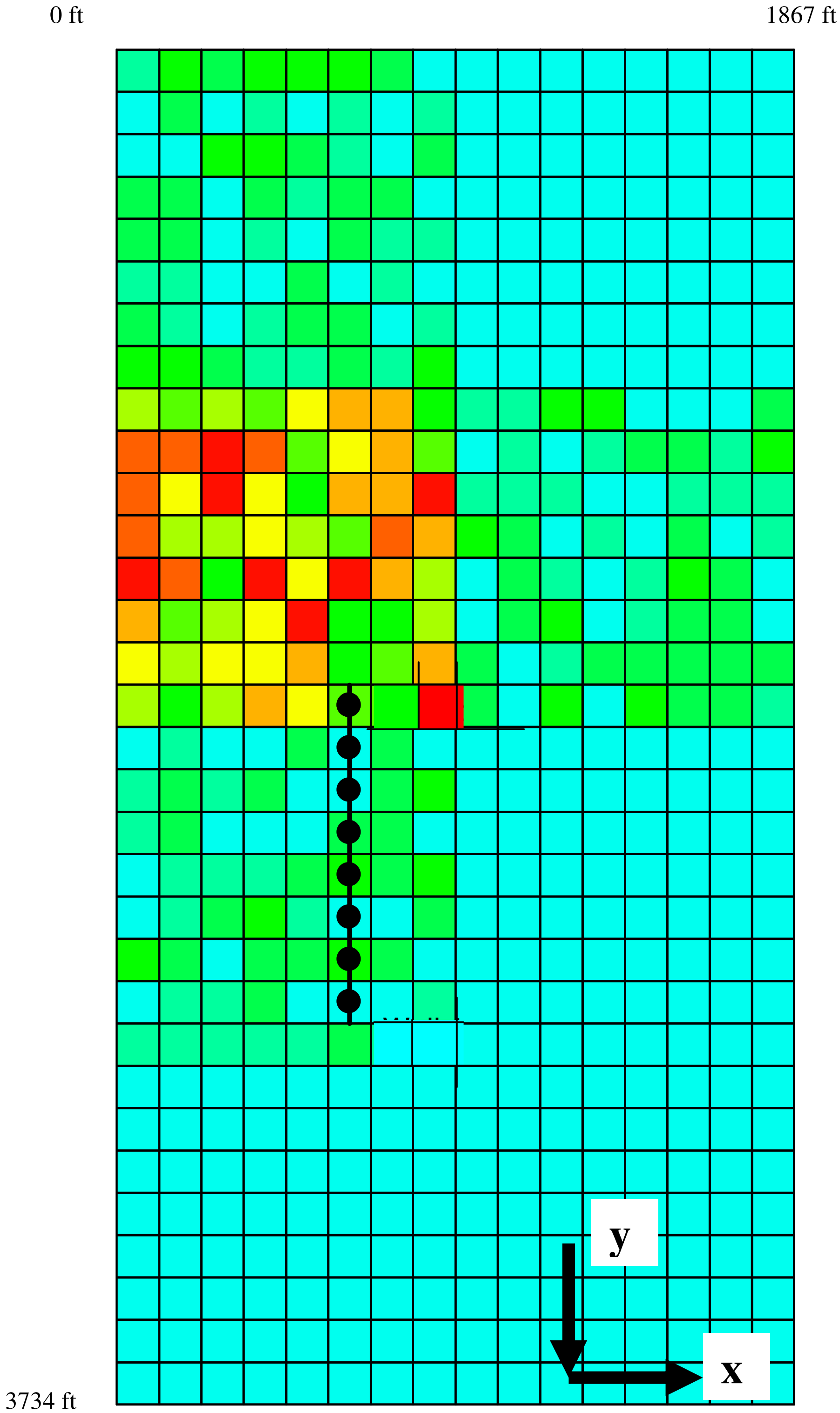

Fig. 19 – Optimal well locations for Case 4

Similar to Case 3, Case 4 can be viewed as having one set of wells. Well 1 begins where Well 2 ends, and Well 2 begins where Wells 1 ends.

The lowest fitness value in the reservoir is 2871.9 MMSCF, which is a 13.3 percentage difference between the optimal fitness values of 3312.2 MMSCF. The percent difference shows that location of a horizontal well in an anisotropic, heterogeneous gas reservoir can affect cumulative production significantly.

Case 4 has a total of 11 simulation runs per generation. The genetic algorithm found the optimum fitness value on the fifth run of the 73rd generation for a total of 797 simulations. The reduction in simulations results in a 42.1 percent difference from the 1,376 simulations it took to run the exhaustive search.

Figure 20 is a graph of the maximum function fitness value per generation for the fourth case.

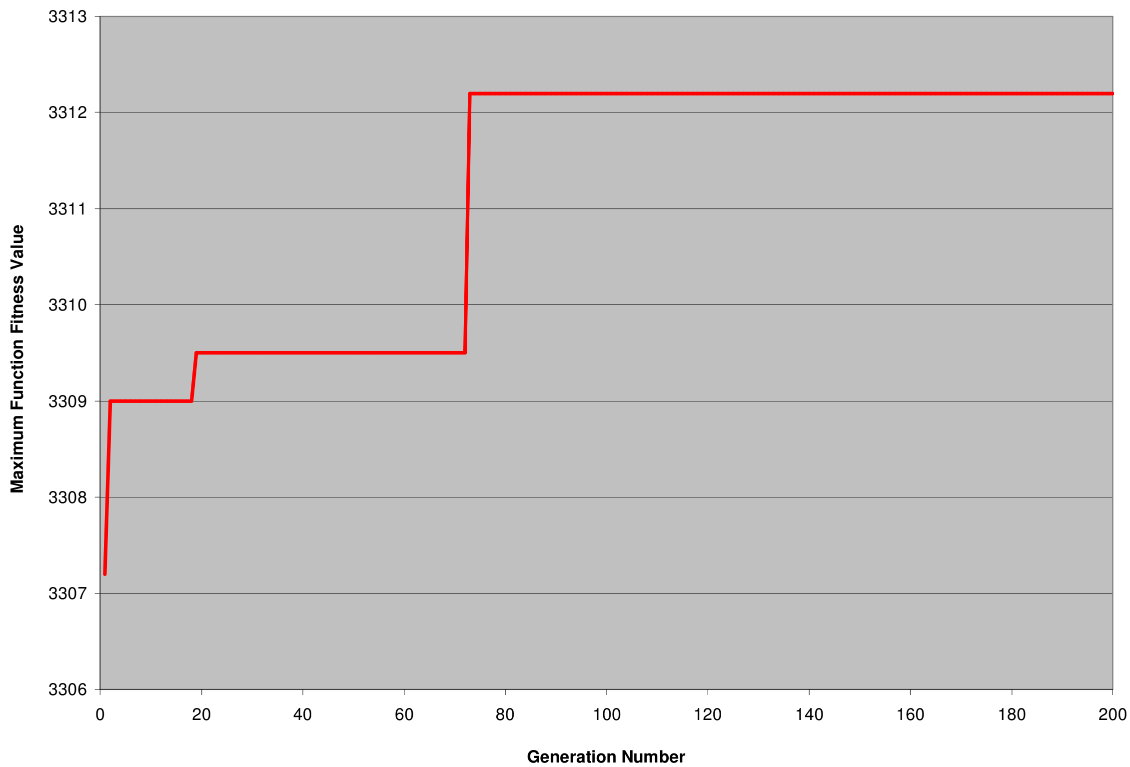

Fig. 20 – Maximum function fitness value vs generation number for Case 4

The maximum function fitness value for the first generation occurred at the eighth run at a value of 3307.2, which is 99.85% of the maximum function fitness value for the entire reservoir. It takes the genetic algorithm to 789 more simulations to find the maximum fitness value of the entire reservoir, again verifying the need for local-optima-searching techniques to be coupled with the genetic algorithm.

CASE 5

The exhaustive run for case five results in one optimum well placement for a single producer well. The location is placed at grid block (8,9) in which the location can act as either the head or toe of the well. The placement of the well results in a cumulative gas produced value of 1595.7 MMSCF. The total number of simulations required for the exhaustive search is 1,376.

Figure 21 shows the fitness value versus well-location for the third case. The well location is shown in Figure 22.

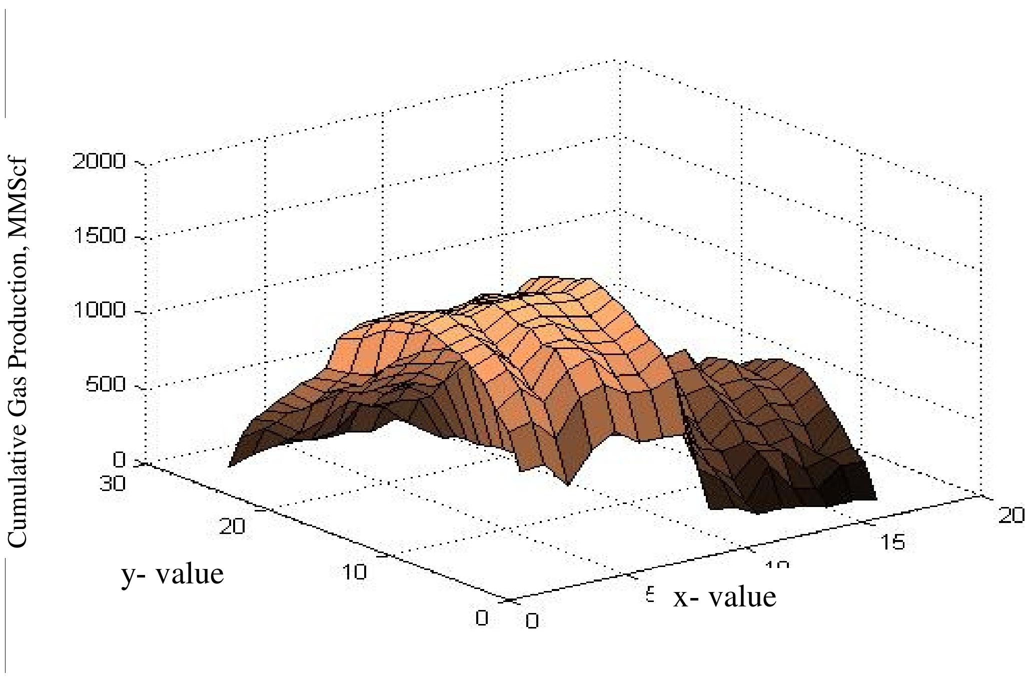

Fig. 21 – Case 5 fitness value vs. well location

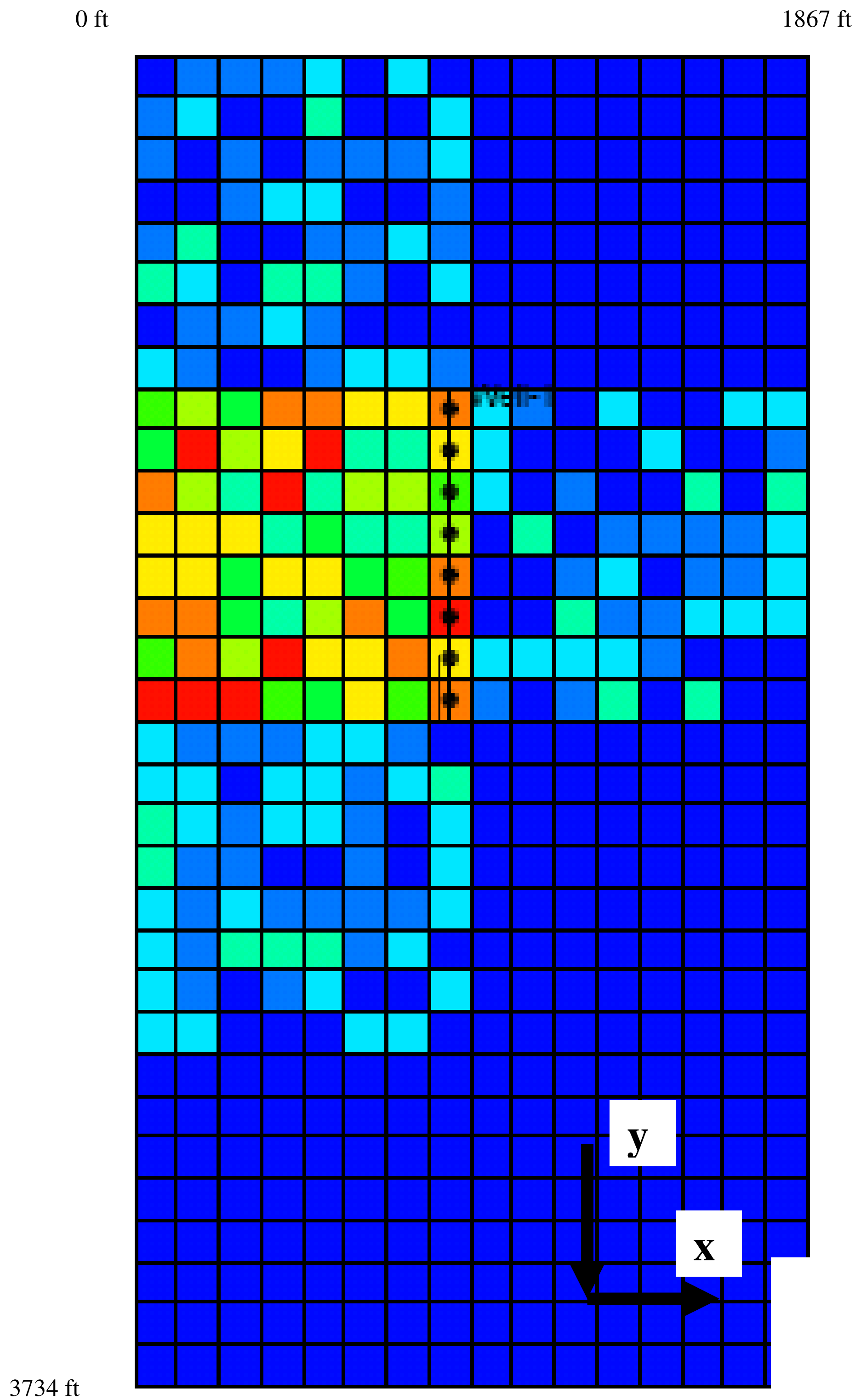

Fig. 22 – Optimal well locations for Case 5

The lowest fitness value in the reservoir is 2.9595 MMSCF, which is a 99.8 percentage difference between the optimal fitness values of 1595.7 MMSCF. The percent difference shows the location of a horizontal well in an anisotropic, heterogeneous gas reservoir can affect cumulative production significantly within the first year of production.

Case 5 has a total of 11 simulation runs per generation. The genetic algorithm found the optimum fitness value on the fourth run of the 18th generation for a total of 202 simulations. The reduction in simulations results in a 85.3 percent difference from the 1,376 simulations it took to run the exhaustive search.

Figure 23 is a graph of the maximum function fitness value per generation for the fourth case.

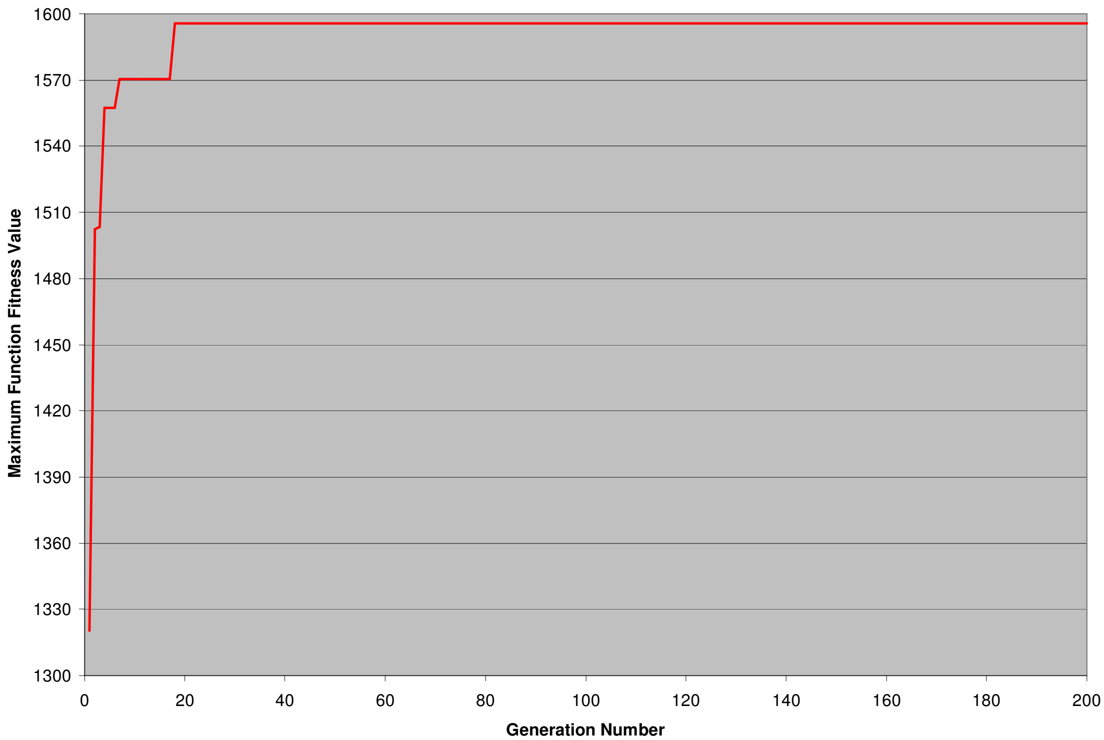

Fig. 23 – Maximum function fitness value vs generation number for Case 5

The maximum function fitness value for the first generation occurred at the tenth run at a value of 1320.3, which is 79.1% of the maximum function fitness value for the entire reservoir. It takes the genetic algorithm to 192 more simulations to find the maximum fitness value of the entire reservoir, again verifying the need for local-optima-searching techniques to be coupled with the genetic algorithm.

RESULTS SUMMARY

Table 4 summarizes the results in regards to the number of simulations for all five cases.

Table 4 – Number of Simulations for Each Method

Number of Simulations			
	Exhaustive	**SGA**	**% Difference**
Case 1	512	82	84.0
Case 2	1376	105	92.4
Case 3	1376	496	64.0
Case 4	1376	797	42.1
Case 5	1376	202	85.3

As evident from the table, the required number of simulations for the GA is dependent upon the complexity of the reservoir model. Genetic algorithms are extremely efficient in locating the area of the global optima, but there is room for improvement when finding THE global optima. There are local-optima-searching proxies that can be coupled with the genetic algorithm to further decrease the required number of simulations for the GA, thus increasing the percent difference even greater. The proxies will be discussed as future work.

CHAPTER IV

CONCLUSION

SUMMARY

An automated exhaustive search method was developed to determine if well placement had an effect on cumulative gas production in a gas reservoir. A simple genetic algorithm was developed and applied to five test cases involving gas reservoirs. The exhaustive search served as the basis to which the genetic algorithm was compared.

The exhaustive search verified that well placement in a gas reservoir does matter in horizontal well cases, especially during the first year of production, where economic value plays a major role, but not so much in vertical wells. Lateral orientation of the horizontal well is also of concern. The first three horizontal well cases (homogeneous gas reservoir, anisotropic gas reservoir with $k_x = k_y \neq k_z$, and anisotropic gas reservoir with $k_x \neq k_y \neq k_z$) show increases in production value between 6.61% and 16.32%. The last horizontal case, Case 5, showed a massive 99.8% increase in cumulative production value within the first year dependent upon well location. The jump in cumulative gas produced is significant since this research assumes no extra cost by drilling the well in different locations within the reservoir. In the vertical well case for a homogeneous reservoir, the location of the well had negligible effect on cumulative gas production.

The genetic algorithm reduced the number of necessary simulations to determine the highest fitness value in the reservoir. Each case showed at minimum a 42.1 percent decrease in the required number of simulations, saving both time and money.

FUTURE RECOMMENDATIONS

This research provides a solid foundation for future work on similar topics. The results presented in this research are problem specific. For generality, several recommendations for future research are discussed.

The stopping criterion needs to be determined to allow for more efficiency and real-world applications. This research uses a maximum generation value as the stopping criterion. Using the maximum generation number as a stopping criterion is beneficial for this research since a foundation is being built for further work, but it is unnecessary. Now that the implementation of the GA has been confirmed to work properly, the GA can now be stopped once it initially reaches the maximum fitness value because no further analysis needs to be taken. Ultimately, a stopping criterion that does not allow for prior knowledge of the answer must be determined to allow for the GA implementation of real-world problems.

A hybrid genetic algorithm (HGA), implementing polytope and proxy methods, should be developed to further enhance the computational benefits of the GA. The polytope and proxy methods enhance the genetic algorithms local optima-searching capabilities. The HGA takes advantage of the genetic algorithms global search, and the polytope and proxy methods local searching abilities to create a more streamlined and robust search optimization technique.

A multi-well horizontal project needs to be undertaken to further enhance the capabilities of the work. Implementation of multiple wells allows for a wider range of problems to be studied, and is also closer to real world applications.

Specific to the research grant that this project falls under, all of the above information should be implemented into a gas condensate reservoir model.

REFERENCES

Beckner, B.L., and Song, X. (1995). Field Development Planning Using Simulated Annealing – Optimal Economic Well Scheduling and Placement. Presented at the *SPE 70th Annual Technical Conference and Exhibition*, Dallas, TX, October 22 – 25. SPE 30650.

Bittencourt, A.C., and Horne, R.N. (1997). Reservoir Development and Design Optimization, Presented at the *SPE 72nd Annual Technical Conference and Exhibition*, San Antonio, Texas, October 5-8. SPE 38895.

Carroll, D. *D.L. Carroll's FORTRAN Genetic Algorithm Driver*. FORTRAN Genetic Algorithm (GA) Driver, http://www.cuaerospace.com/carroll/ga.html. Downloaded 25 January 2009.

Goldberg, D.E. (1989). *Genetic Algorithm in Search, Optimization, and Machine Learning*. Addison-Wesley. Reading, MA.

Gűyagűler, B., and Gűmrah, F. (1999). Comparison of Genetic Algorithm with Linear Programming for the Optimization of an Underground Gas-Storage Field. *"In Situ"*, 23, p. 131-149.

Gűyagűler, B., and Horne, R.N. (2001). Uncertainty Assessment of Well-Placement Optimization, Presented at the *76th SPE Annual Technical Conference and Exhibition*, New Orleans, Louisiana, September 30 - October 3. SPE 87663.

Gűyagűler, B. (2002). *Optimization of Well Placement and Assessment of Uncertainty*. Ph.D. Dissertation, Stanford University.

Mohaghegh, S. (1998). Candidate Selection for Stimulation of Gas Storage Wells Using Available Data With Neural Networks and Genetic Algorithms. Presented at the *73rd SPE Easter Regional Meeting,* Pittsburgh, PA, November 9-11. SPE 51080.

Montes, G., and Bartolome, P. (2001). The Use of Genetic Algorithms in Well Placement Optimization. Presented at the *2nd Annual SPE Latin American and Caribbean Petroleum Engineering Conference,* Buenos Aires, Argentina, March 25-28. SPE 69439.

Özdoğan, U., and Horne, R.N. (2004). Optimization of Well Placement with a History Matching Approach. Presented at the *79th SPE Annual Technical Conference and Exhibition*, Houston, Texas, September 26-29. SPE 90091.

Pan, Y., and Horne, R.N. (1998). Improved Methods for Multivariate Optimization of Field Development Scheduling and Well Placement Design. Presented at the *73rd SPE Annual Technical Conference and Exhibition*, New Orleans, Louisiana, September 27-30. SPE 49055.

Sarich M.D. (2001). Using Genetic Algorithms to Improve Investment Decision Making. Presented at the *5th Annual SPE Asia Pacific Oil and Gas Conference and Exhibition*, Jakarta, Indonesia, April 17-19. SPE 68725.

Yeten, B., Durlofsky, L.J., and Aziz K. “Optimization of Nonconventional Well Type, Location, and Trajectory.” SPE Journal Volume 8, Number 3 (September 2003): 200-210. SPE 86880 received May 14, 2003.

VITA

Trevor Howard Gibbs received his Bachelor of Science degree in Physics from Stephen F. Austin State University in Nacogdoches, Texas in 2007. He began his graduate studies in the petroleum engineering program at Texas A&M University in College Station, Texas during the fall semester of 2007, and after spending one year in College Station taking courses, he moved to the Texas A&M University Qatar campus in Doha, Qatar to complete his research. He officially received his Master of Science degree in May 2010. His research interest includes horizontal well placement in gas reservoirs using genetic algorithms. Mr. Gibbs may be reached at his mailing address seen below. His email address is gibbsth@gmail.com.

Harold Vance Department of Petroleum Engineering
Texas A&M University
3116 TAMU – 507 Richardson Building
College Station, TX 77843-3116
c/o Ding Zhu or Hadi Nasrabadi