

Applications of genetic algorithms in exploration and production

MICHAEL D. McCORMACK and RICHARD F. STOISITS, Optimization Associates, Plano, Texas, U.S.

DONALD J. MACALLISTER and KELLY D. CRAWFORD, ARCO Exploration and Production Technology, Plano, Texas, U.S.

The use of optimization algorithms to improve the performance of reservoir engineering systems and production operations has a long and successful history. Likewise, optimization has been applied to seismic processing problems for many years, generally via the least-squares method. More recently, investigations have focused on the use of a new class of advanced optimization methods for near-surface statics computations, velocity analysis, and seismic inversion.

In this paper we will discuss the operation and benefits of using one of these new optimization techniques—genetic algorithms (GA). A GA can solve very large and complex problems and overcome some shortcomings of traditional optimization methods. As the name suggests, the underlying model employed in a GA is based on biological genetics and Darwinian notions of *adaptation to the environment and survival of the fittest*.

After describing the operation of a GA, we present examples showing how a GA can solve a material-balance problem in production engineering and multiple suppression in seismic data processing.

Genetic algorithms. A GA is an optimization technique that arrives at a solution using a method that crudely mimics the process of evolution in biological systems. The underlying motivation for the GA methodology is grounded in the concept that living biological species represent optimum solutions to the problem of survival in a hostile environment.

In a biological community, those individuals with chromosomes that enable them to best survive in the environment in which they live will, on average, tend to dominate the species and pass on their superior characteristics to the next generation. Over many generations most individuals in a population will acquire these good genetic characteristics. Occasionally a mutation will occur in the genetic material of an individual. In most instances, this mutation is bad; but once in a while it benefits the organism. If so, then this mutated characteristic will be passed on to subsequent generations. Many optimization problems can be formulated in terms of this evolutionary paradigm. The values of a set of control parameters for the problem being optimized correspond to the chromosomes in an individual. Just as each individual in a population has a unique set of chromosomes representing one solution to the environmental survival “problem,” one can create a “population” of solutions each having its own unique set of values for the control parameters. And finally, one can create new solutions to the control problem by combining the existing population of solutions using rules borrowed from genetics.

The high-level methodology of a simple GA is summarized by the following steps:

- 1) Represent the variables in the problem as a string of concatenated numbers, called a chromosome. Collectively, the values of the members of a chromosome are one

potential solution to the problem being optimized. In the seismic data processing example that follows, the unknown variables are the digital samples of a wavelet. For illustrative purposes assume that the wavelet contains five samples. The goal of the GA optimization is to find values for each sample that satisfy some performance criteria. One possible chromosome could be written as:

0.12 1.04 -0.89 -0.34 0.03

- 2) Create a “population” of several hundred chromosomes using a random number generator. Each five-sample chromosome can be thought of as one individual in the population.

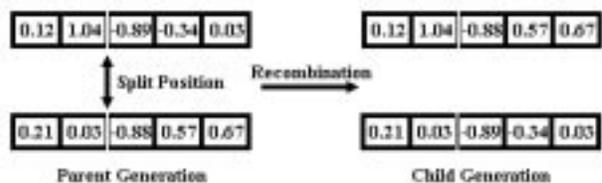
0.12 1.04 -0.89 -0.34 0.03

-0.45 0.23 -0.1 0.99 0.45

⋮

0.21 0.03 -0.88 0.57 0.67

- 3) Evaluate the fitness of each chromosome in the population using an appropriate *fitness function*. This function provides a measure of how “well” a chromosome (or wavelet) solves the problem. In the example above, the first wavelet might have a fitness of 23.5, the second 0.56, and so on.
- 4) Create two new “child” chromosomes by selecting two chromosomes from the original “parent” population and applying mutation and recombination operators. Chromosomes are selected for mating based on their fitness value—a chromosome with a higher fitness is more likely to be selected than one with a lower fitness value. A mutation operator will randomly alter the value of one or more of the members of the chromosome. Recombination involves splitting two parent chromosomes at a random location in the string and recombining the split strings to create two “child” chromosomes. In the example below, two chromosomes from the original parent generation are selected and “split” between the second and third numbers in the string. The right-hand substring of each chromosome is exchanged with the other chromosome to create two new child chromosomes. Each new child chromosome represents a new solution to the problem. Additional child chromosomes are created until the population of children equals that of the parents.



- 5) Evaluate the new child chromosomes using the fitness function. Replace the parent population by the child population.
- 6) Iterate back to the fourth step above until a chromosome with an acceptably high fitness value is found, then STOP.

Surprisingly this straightforward procedure will find an excellent wavelet solution. The major strength of genetic algorithms is that they are ideally suited for solving highly nonlinear problems (i.e., situations where the fitness functions are multimodal and have discontinuities). On average, crossover and small mutations tend to exploit the local optima (local peaks or troughs), while larger mutations tend to explore the search space (locating a different peak or trough). Crossover is more generally a convergent operator, while mutation has both convergent and divergent properties. This combination of exploitation and exploration is what gives the genetic algorithm its power.

Surface multiple suppression. *TLE's* January 1999 issue contained a special section on theoretical and practical developments in the suppression of surface and interbed multiples. Guitton and Cambois summarized one approach to removing surface multiples using the equation:

$$P_0(x,t) = P(x,t) - W^{-1}(t) * P(x,t) * P(x,t) - W^{-1}(t) * W^{-1}(t) * P(x,t) * P(x,t) * P(x,t) + \dots$$

X and t are the offset distance and time, $P(x,t)$ is the input seismic data, $P_0(x,t)$ is the surface multiple-free data, $W^{-1}(t)$ is the inverse of the seismic wavelet, and $*$ indicates 2-D convolution. The second term of the right side subtracts out first-order surface multiples, the third term removes second-order multiples, and so on for higher order multiples. This expression is nonlinear because the second- and higher-order multiple terms involve two or more convolutions with the unknown inverse seismic wavelet, $W^{-1}(t)$.

We know $P(x,t)$ and can easily compute the autoconvolutions specified in each of the terms on the right of this equation. What we need to solve for, in order to use this equation for multiple removal, is the inverse seismic wavelet, $W^{-1}(t)$. To do this we can use a GA and follow the steps outlined in the last section. The chromosome for the GA will be the digital sample values of $W^{-1}(t)$. In the synthetic seismic data example that follows we used 21 samples at a 4-ms sample rate (rather than the five samples that were used in the previous discussion). Initially each sample in the chromosome is assigned a random value that was constrained between -2 and $+2$; 100 chromosomes were generated representing the initial parent population. A fitness function is also required to evaluate the performance of each chromosome solution. Since it seems physically reasonable that the multiple-free data contain less energy than the data containing primaries and multiples, the best solution for $W^{-1}(t)$ will be the one that minimizes the total energy in the $P_0(x,t)$ data set.

The input data and processed output shot records are displayed on the left and right, respectively, of Figure 1. The genetic algorithm has attenuated the water-bottom multiples by about 22 db. Total processing time to compute the inverse seismic wavelet was less than one minute on a Unix box.

The material-balance problem. In an oil field, oil, water, and gas are produced from wells by the natural pressure resulting from the weight of the overlying rocks. Because

the pressure declines as more and more fluids are taken from the reservoir, it is common practice to reinject pressurized water and gas back into the reservoir to maintain reservoir pressure. A key responsibility of a reservoir engineer is to develop a comprehensive picture of the flow of produced and injected fluids in the reservoir so that the maximum volumes of hydrocarbons can be recovered.

"Material balance" is a software tool used by reservoir engineers to develop and efficiently produce the hydrocarbon reserves of a field. In order to maximize oil recovery, it is necessary to know the amounts and distributions of oil, water, and gas fluids in the reservoir at any given time. Using the material-balance process, engineers estimate the relative amounts or saturations of oil, water, and gas in the reservoir, and pressure changes occurring in the reservoir as a result of injecting and producing fluids. These estimates are compared with actual field measurements of reservoir pressure to assess the accuracy of the material-balance calculation. A key constraint on this calculation is the conservation of total mass of injected and produced fluids—the so-called "material balance"—from which the name of the procedure is derived.

The field is divided into a series of patterns centered on producing wells with injection wells on the pattern borders (Figure 2). A separate material-balance calculation is done for each pattern. Because injection wells are on pattern boundaries, fluids from a single injection well must

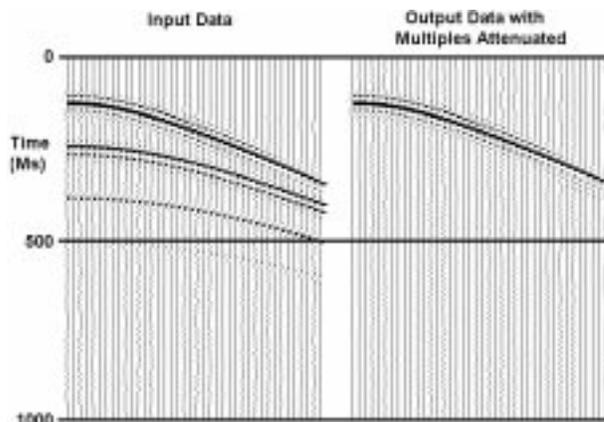


Figure 1. Input synthetic seismic data in the window 0-1000 ms (left) and the processed output data with the surface multiples attenuated (right).

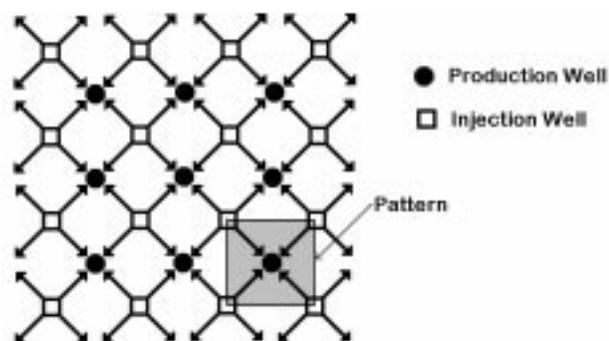


Figure 2. Plan view of production and injection wells in a field. Each square consists of a group of wells called a "pattern." Note that injection wells can be shared between adjacent patterns. The arrows illustrate the approximate flow directions of fluids in the reservoir, the fluid moving from injectors to production wells.

be allocated to more than one pattern. Because of inhomogeneities in the reservoir rocks, the injected fluids do not move out symmetrically in all directions, so that one cannot assume that a quarter of the injected fluid moves into each of the four contiguous patterns that an injection well contributes to. The central problem in a material-balance calculation is to determine how much of the fluid introduced into an injection well goes into each of the patterns of which that injection well is a member. The difference between the total injected and produced fluids in a pattern can be used to calculate the pressure that should exist in that pattern. A net decrease in fluid in a pattern results in a decrease in the pressure in that pattern; the converse is also true. Current practice in the industry is to manually "guess" the contribution of fluids from each injection well surrounding a producer until a "reasonable" pressure profile for all patterns in the field is achieved. This is clearly a very labor intensive and subjective process.

Past efforts to automate this allocation process using a least squares, linear programming approach have not been satisfactory because of the nonlinear pressure-volume-temperature (PVT) behavior of reservoir fluids, and the large size of the optimization that was being attempted. Because of these problems, we felt that a stochastic optimization technique such as genetic algorithms could be successfully applied to the material-balance procedure.

Genetic algorithm for the material-balance system. The first task was to develop an appropriate chromosome representation for the material-balance problem. Figure 3 is a plan view of four injection wells and one center producer. In this simple diagram, the fluid being injected into injection well 1 splits into four "streams" in the formation, going to four different producing wells. For example, the fraction of fluid flowing to the southeast could be 0.15, to the northeast 0.37, to the northwest 0.22 and to the southwest 0.26. Note that these four fractions, or allocation factors, must sum to 1.0. For injection well 1 we create a chromosome representation that consists of the four fractional allocations concatenated to form a string: [0.15][0.37][0.22][0.26]. This chromosome can be expanded by further appending a similar four-value string that represents the four allocation factors for injection well 2. The resulting chromosome would be: [0.15][0.37][0.22][0.26][0.23][0.12][0.00][0.65]. The entire chromosome is constructed in an analogous manner to include the allocation factors for all injection wells and all producing wells that we wish to consider in the optimization. The length of a typical chromosome string for a real-world oil field problem is very long—typically 3000-7000 floating point values will need to be determined.

Because the representation we described above took far too much computer time, we employed an alternate chromosome representation for this problem. We replaced each

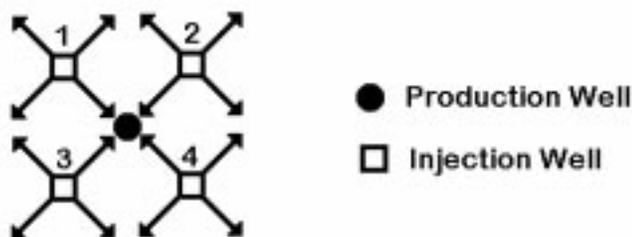


Figure 3. Plan view of injectors and producer wells, showing fluid outflow from each injection well into the formation.

floating-point value in the chromosome by a 10-bit binary string equivalent. This new representation allowed us to take advantage of a fast, powerful binary bit genetic algorithm that reduced the convergence time to acceptable levels.

The other key step in developing a GA solution is to create a fitness (or cost) function to evaluate the performance of each chromosome. Our fitness function used the sum of the squared differences between the estimated and measured values of pattern pressure to measure the performance of a chromosome. A physically reasonable constraint was also applied that forced the estimated allocation factors to vary slowly with time.

Results of the material balance system. Extensive tests were conducted using both synthetic data sets and data from an operating field. The synthetic data studies demonstrated that pattern pressures could be matched within a fraction of a percent, and the allocation factors could be matched within a few percent.

In actual field data studies, the "correct" answers for the allocation factors are not directly known, but we can get a feeling for the accuracy of the GA-based procedure in terms of the matches that were obtained between the material-balance estimates of pattern pressures and actual field measurements of the same quantities. Most patterns pressure matches were within ± 50 psi. To achieve this level of accuracy took 12-18 hours of computing on a high-end SGI Unix box. By comparison, when the same material-balance calculations are performed manually by reservoir engineers, it takes 9-12 man-months to obtain pattern pressure matches that are at best within ± 200 psi.

The above system was recently transferred into operating groups of a major oil company and is in production use. At one field it is estimated that the improved accuracy of the GA-based material-balance system will result in better reservoir management practices and will ultimately result in the recovery of an additional 12 000 000 barrels of oil.

Conclusions. While the examples described in this article represent only a small sampling of existing and potential applications of GA technology, we hope that the reader takes away some important concepts:

- Genetic algorithm technology is not a laboratory or academic curiosity. It can and is being applied to serious, real-world business problems.
- GAs can be used to solve exceptionally large and complex problems that cannot be resolved in a timely or economic fashion using more conventional optimization methods. This new technology should encourage us to revisit old problems that have not been successfully solved in the past.
- GAs have been applied in other areas of our industry with substantial economic returns. Examples include product scheduling at coal mines, gasoline tanker truck, and pipeline product routing.

Suggestions for further reading. Textbooks that describe details of the theory and implementation of genetic algorithms include *Handbook of Genetic Algorithms*, edited by Davis (Van Nostrand Reinhold, 1991), *Genetic Algorithms + Data Structures = Evolution Programs* (second edition) by Michalewicz (Springer-Verlag, 1992), and *Genetic Algorithms in Search, Optimization and Machine Learning*, by Goldberg (1989). E

Corresponding author: M. McCormack, m.d.mccormack@worldnet.att.net