Optimizing Cyclic-Steam Oil Production With Genetic Algorithms

The full-length paper details a project that applied a new technology, genetic algorithms, to the problem of scheduling oil production by cyclic steaming at an oil field in the San Joaquin Valley. The paper focuses on three themes: the successful solution of a production problem with new technology; the impact of that technology on oilfield personnel; and the potential of that technology to support other types of projects.

Introduction

The Antelope reservoir in the Cymric field, in the San Joaquin Valley, is a siliceous shale reservoir containing 12 to 13°API heavy oil. The reservoir consists primarily of diatomite, characterized by its high porosity, high oil saturation, and very low permeability. Approximately 430 wells are producing from this reservoir, with an average daily production of 23,000 bbl. The oil from the field is recovered using a Chevron-patented cyclic-steam process. A fixed amount of saturated steam is injected into the reservoir during a 3- to 4-day period. The high-pressure steam fractures the rock, and the heat from the steam reduces oil viscosity. The well is shut in during the next couple of days, known as the soak period. Condensed steam is absorbed by the diatomite, and oil is displaced into the fractures and wellbore. After the soak period, the well is returned to production. The flashing of hot water into steam at the prevailing pressure provides the energy to lift the fluids to the surface. The well flows for approximately 20 to 25 days. After the well dies, the same cycle is repeated. Cycle length is 26 to 30 days.

*This article, written by Assistant Technology Editor Karen Bybee, contains highlights of paper SPE 93906, "Optimizing Cyclic-Steam Oil Production With Genetic Algorithms," by **A. Patel,** SPE, ChevronTexaco Energy Technology Co; **D. Davis,** Nutech Solutions Inc.; **C. Guthrie,** SPE, and **D. Tuk,** ChevronTexaco Energy Technology Co.; and **Tai Nguyen,** SPE, and **J. Williams,** ChevronTexaco North America Upstream, prepared for the 2005 SPE Western Regional Meeting, Irvine, California, 30 March–1 April.*

Because there is no oil production during the steaming and soaking period, there is an incentive to minimize the steaming frequency and increase the length of the cycle. But because well production is highest immediately after returning to production and declines quickly thereafter, a case can be made for increasing the steaming frequency and reducing the length of the cycle. This suggests that there is an optimum cycle length for every well that results in maximum productivity during the cycle. Because there are more than 400 wells in the field, and there are constraints of steam availability and distribution system, as well as facility constraints, the result is a formidable scheduling problem.

Genetic Algorithms

Genetic algorithms (GAs) are global optimization techniques developed by John Holland in 1975. They are one of several techniques in the family of evolutionary algorithms—algorithms that search for solutions to optimization problems by "evolving" better and better solutions. A genetic algorithm begins with a "population" of solutions and then chooses "parents" to reproduce. During reproduction, each parent is copied, and then parents may combine in an analog to natural crossbreeding, or the copies may be modified, in an analog to genetic mutation. The new solutions are evaluated and added to the population, and low-quality solutions are deleted from the population to make room for new solutions. As this process of parent selection, copying, crossbreeding, and mutation is repeated, the members of the population tend to get better. When the algorithm is halted, the best member of the current population is taken as the solution to the problem posed.

One critical feature of a GA is its procedure for selecting population members to reproduce. Selection is a random process, but a solution member's quality biases its probability of being chosen. Because GAs promote the reproduction of high-quality solutions, they explore neighboring solutions in high-quality parts of the solution search space. Because the process is randomized, a GA also explores parts of the search space that may be far from the best individuals currently in the population.

In the last 20 years, GAs have been used to solve a wide range of optimization problems. There are many examples of optimization problems in the petroleum industry for which GAs are well suited. At ChevronTexaco, in addition to the cyclical-steam scheduling problem, well placement, rig scheduling, portfolio optimization, and facilities design have been addressed with GAs. At NuTech Solutions, GAs have been used in planning rig workover projects so that overall workover time is reduced, planning production across multiple plants to reduce costs, planning distribution from multiple plants to a large number of customers to reduce costs, and controlling pipeline operations to reduce costs while satisfying pipeline constraints.

All of these problems have common features: they cannot be formulated as linear programming problems, they involve a large number of complex constraints both hard and soft, and significant increases in profits can be obtained if the problems are solved better than they were being solved before the creation of a computerized optimization procedure.

Problem Formulation

The cyclic-steam scheduling problem is formulated as a GA optimization problem in which the objective is to maximize cumulative production over a 2-month period. The fitness function is calculated as the cumulative production minus the penalties for violating the soft constraints.

The problem has many constraints. The field-level constraints include steam availability and the maximum number of wells steaming. Gauge-station constraints include minimum amount of steam used and maximum number of wells on production. The header-level constraints include maximum number of wells steaming, and the individual-well constraints include maximum/minimum number of production days.

Additionally, there are operational constraints such as communication where multiple wells must be steamed together and wells blocked because of rig activities.

Although all of the field constraints could have been incorporated in the problem formulation as hard constraints, constraints that absolutely cannot be violated, the decision was made to make many of the constraints soft constraints, constraints that can be violated but with an associated penalty. An example of a hard constraint is the total steam available on a given day for the whole field, whereas some soft constraints are maximum amount of steam used by a well group and minimum number of wells steaming in a header.

The optimization is stopped when one of the following criteria is satisfied.

• A specified number of generations have been created.

• A specified amount of time has elapsed.

• The fitness function has not improved over a specified number of generations.

The GA used multiple heuristics to enhance its performance and speed up its search for high-quality solutions. To begin with, when it created the initial population of solutions, "the seed," it used heuristics based on those that the well operators and steam operators used at the oil field. It also used some heuristics developed for the project to find good initial schedules. An example of such a heuristic is "attempt to steam high-production wells at their optimal cycle length—the length of time between steaming at which a well's average daily production is maximized." The constraints of the problem made it impossible to steam all wells at their optimal cycle length, but inserting schedules based on this as a goal into the initial population gave the algorithm some high-quality solutions that could be mutated and crossbred with other types of solutions to find even better solutions.

The technique used for representing solutions was not the approach commonly found in GA textbooks. An indirect encoding approach was used in which each solution was a permutation list of wells, with multiple entries allowed for the same well. Then a decoding procedure was used that simulated the effects of various schedules to translate the permutation list into an actual schedule. The schedule builder looks at the first well on the list and simulates steaming it on Day 1. If this process violates no hard constraints, then the well is scheduled for steaming on Day 1. The schedule builder then looks at the second well on the list. It simulates the effects of steaming that well together with the first well on Day 1. If no hard constraints are violated, this well is added to the schedule for Day 1. If hard constraints are violated, the well is not added to the schedule. The process continues, considering each well for steaming on Day 1, and adding each well, in order, that can be steamed without violating a hard constraint. Then the process continues with Day 2, considering each well, in order, that was not already steamed on Day 1. The process is repeated for Days 3 and 4. The critical point is that the schedule-building process will not build a schedule that violates a hard constraint. Also, this schedule-building process uses some clever heuristics and a simulator to transform a list of wells into a feasible steaming schedule. Once a schedule is built, it can be evaluated, and its "score" is returned to the GA as the evaluation of the original solution, the list of wells.

The optimization process uses heuristics to initialize the population, as well as randomly generated solutions to fill out the initial population. The process includes intelligent heuristics in the procedures used to modify new solutions. Also used are crossbreeding procedures appropriate to combining different permutations to combine two parents to produce a child. The process includes a good deal of domain knowledge in the schedule builder to produce feasible schedules. A post-processor is included that checks to find simple changes that could be made to the best solution found to improve its quality.

The interface to the optimizer gives the well operators and steam operators at the field a great deal of power and flexibility in their interactions with the system. The operators can edit the well data that are entered into the optimizer. They can select optimization heuristics and procedures used in a run. They can parameterize the objective function that specifies the goals of the run. They can activate, deactivate, and parameterize the hard and soft constraints. They also can edit the solutions found by the optimizer in cases in which there is a constraint known to the operators that is not reflected in the databases available to the optimizer.

Application

A pilot test was conducted to demonstrate the feasibility, and to determine the potential benefit, of the approach. A well group comprising 21 wells was selected, and the test was conducted for 60 days. During this period, the wells in this group were operated only on the recommendations of the scheduler without any adverse effects. The work process was changed to facilitate the scheduler needs. The operators successfully adopted the new process, which relied much more on production data that were updated daily, and used the visualizer to determine well performance parameters. Daily logs were kept of the recommendations that were followed. At the end of the test period, a comparison was done against the baselines.

Cumulative production from this well group during the 60-day period increased 11%. The number of active wells was the same as in the base period, implying that well productivity increased. The scheduler recommended shorter cycles for many of the wells compared to field practice. Cumulative steam during the same period increased 17%. Because the total steam available for the field is constrained, it would not be possible for the scheduler to increase the steam usage the way it did in the pilot test. As a result, the increase in production resulting from higher steam use was backed out. The final conclusion was that use of the scheduler would increase production 3 to 5%.

The pilot test helped identify issues that needed to be addressed before implementing the process fieldwide. The full-length paper describes each individual part of the optimization tool.

Results

As a risk-mitigation measure, the plan for developing the scheduler with all of its functionality was broken down into three releases. Each subsequent release would be approved to move forward after the previous release was considered to be successful. Release 1 of the scheduler for the whole field was deployed in January 2003. From 8 January 2003 to 8 March 2003, production increased 6.4% compared to the baseline. Not all of this increase was a result of the scheduler because new wells were drilled, and the production from these wells was a part of the increased production.

To estimate the economic benefit of the scheduler, the production gain attributable to factors other than the scheduler had to be deducted. Of the total 6.4% increase, there was a 1.4% gain that cannot be attributed to anything other than the scheduler. A preliminary examination of the project indicates a payout period of approximately 75 days. The scheduler has been in daily use since August 2003, and the operators like its consistency and fast response. JPT

For a limited time, the full-length paper is available free to SPE members at www.spe.org/jpt. The paper has not been peer reviewed.