A New Well-Pattern-Optimization Procedure for Large-Scale Field Development

J.E. Onwunalu, SPE, and **L.J. Durlofsky,** SPE, Stanford University

Summary

The optimization of large-scale multiwell field-development projects is challenging because the number of optimization variables and the size of the search space can become excessive. This difficulty can be circumvented by considering well patterns and then optimizing parameters associated with the pattern type and geometry. In this paper, we introduce a general framework for accomplishing this type of optimization. The overall procedure, which we refer to as well-pattern optimization (WPO), includes a new well-pattern description (WPD) incorporated into an underlying optimization method. The WPD encodes potential solutions in terms of pattern types (e.g., five-spot, nine-spot) and pattern operators. The operators define geometric transformations (e.g., stretching, rotating) quantified by appropriate sets of parameters. It is the parameters that specify the well patterns and the pattern operators, along with additional variables that define the sequence of operations, that are optimized. A technique for subsequent well-by-well perturbation (WWP), in which the locations of wells within each pattern are optimized, is also presented. This WWP represents an optional second phase of WPO. The overall optimization procedure could be used with a variety of underlying optimization methods. Here, we combine it with a particle-swarm-optimization (PSO) technique because PSO methods have been shown recently to provide robust and efficient optimizations for well-placement problems.

Detailed optimization results are presented for several example cases. In one case, multiple reservoir models are considered to account for geological uncertainty. For all examples, significant improvement in the objective function is observed as the algorithm proceeds, particularly at early iterations. The use of well-by-well perturbation (following determination of the optimal pattern) is shown to provide additional improvement. Limited comparisons with results using standard well patterns of various sizes demonstrate that the net present values (NPVs) achieved by the new algorithm are considerably larger. Taken in total, the optimization results highlight the potential of the overall procedure for use in practical field development.

Introduction

Field-development optimization entails the determination of the number, type, location, trajectory, and drilling schedule for new wells such that an objective function is maximized. Examples of relevant objective functions include NPV for the project and cumulative oil produced. Computational optimization is commonly employed to address this problem, with recent applications involving hundreds of wells (Volz et al. 2008).

A straightforward and common approach for representing the solution parameters in field-development optimization is to consider a series of wells and to concatenate the well-by-well optimization parameters. For problems with many wells, however, the number of optimization variables can become excessive and the search space can become very large, thereby increasing the complexity of the optimization problem. This can lead to a degradation in algorithm performance. Additional complications may result when necessary constraints (e.g., minimum well-to-well distances) are incorporated, and this can also affect algorithm performance negatively.

In this work, we propose a new procedure, called the WPO algorithm, which can be used for optimization problems involving a large number of wells. WPO consists of a new WPD, followed by an optional WWP, with both procedures incorporated into a core optimization methodology. WPD represents solutions at the level of well patterns rather than individual wells, which leads to a significant reduction in the number of optimization variables for problems with large numbers of wells. In fact, using WPD, the number of optimization variables is independent of the number of wells considered. In WPD, each potential solution consists of three elements: parameters that define the basic well pattern, parameters that define so-called well-pattern operators, and the sequence of application of these operators. The well-pattern operators define pattern transformations that vary the size, shape, and orientation of the well patterns considered in the optimization. The optimum number of wells required, in addition to the producer/injector ratio, is obtained from the optimization. Optimized solutions based on WPD are always repeated patterns (i.e., the method does not lead to irregular well placements). The subsequent use of WWP allows (limited) local shifting of all wells in the model, which enables the optimization to account for local variations in reservoir properties.

There have been many previous studies addressing the optimization of well location and type. Most of these investigations, however, have focused on optimizations involving relatively few wells rather than large-scale field development. For a discussion of some of this literature, see Onwunalu and Durlofsky (2010). Here, we consider previous investigations that proposed optimization strategies that share similarities with our procedure. Pan and Horne (1998) applied kriging and least-squares techniques to determine optimal well scheduling in a pattern waterflood project. Ozdogan et al. (2005) used a fixed-pattern approach to optimize well count and location of production and injection wells using a hybrid genetic algorithm (GA). The procedure, which was applied to a field case, reduced the number of simulations required in the optimization. Emerick et al. (2009) presented a procedure for handling different well-placement constraints such as maximum well length and minimum distance between wells. They used a two-population binary genetic algorithm where each individual belongs to one of the two populations, depending on its feasibility. Litvak and Angert (2009) described a robust field-development-optimization procedure that has been applied successfully to giant fields (Litvak et al. 2007; Volz et al. 2008). They considered three pattern types—inverted five-spot, inverted seven-spot, and staggered line-drive—to reduce the number of optimization variables. These investigators considered different well types and a specified set of well spacings in their optimizations.

The optimization procedure introduced here differs from the work of Litvak and Angert (2009) in several respects. We use a different technique to represent potential field-development scenarios, and our algorithm considers very general well patterns. This is accomplished through use of pattern-transformation operations, which allow patterns to be rotated, stretched, or sheared to an

This paper (SPE 124364) was accepted for presentation at the SPE Annual Technical Conference and Exhibition, New Orleans, 4–7 October 2009, and revised for publication. Original manuscript received for review 4 July 2009. Revised manuscript received for review 22 July 2010. Paper peer approved 5 August 2010.

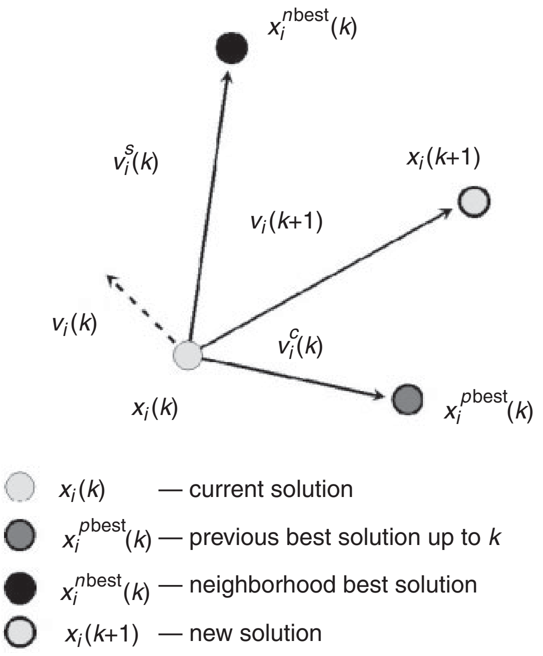

$x_i^{nbest}(k)$
$v_i^s(k)$
$x_i(k+1)$
$v_i(k+1)$
$v_i(k)$
$v_i^c(k)$
$x_i(k)$
$x_i^{pbest}(k)$
$x_i(k)$ — current solution
$x_i^{pbest}(k)$ — previous best solution up to k
$x_i^{nbest}(k)$ — neighborhood best solution
$x_i(k+1)$ — new solution

Fig. 1—Graphical illustration of PSO particle velocity and particle-position updates for a single particle, $\mathbf{x}_i(k)$, in a 2D search space.

optimal degree. This can be important, for example, in situations where there are preferential flow directions in the field. In addition to standard patterns, the algorithm accepts user-defined patterns. WPD also eliminates the need for well-to-well distance constraints in the optimization. This is useful because highly constrained optimization problems are generally more difficult to solve than less-constrained problems (Engelbrecht 2005). Finally, the use of WWP enables the local adjustment of well locations.

The well-pattern description and WWP developed in this work can be applied in conjunction with any number of underlying optimization algorithms. Here, we use the PSO algorithm (Kennedy and Eberhardt 1995; Eberhardt and Kennedy 1995), which is a relatively new global-optimization procedure, as our core algorithm. In recent work (Onwunalu and Durlofsky 2010), we applied PSO to several well-placement-optimization problems involving vertical, deviated, and dual-lateral wells. The PSO algorithm was found to provide better results, on average, than the binary GA for the problems considered. Our investigation of PSO for well-placement optimization was motivated by the study of Matott et al. (2006), who compared a variety of optimization algorithms for groundwater-remediation problems and found PSO to be generally superior to simulated annealing, conjugate gradient, and continuous GAs. PSO tends to require fewer iterations (or function evaluations) relative to other algorithms to produce solutions of similar quality (Abdelhalim and Habib 2009; Onwunalu and Durlofsky 2010), which makes it well suited for use in field-development optimization.

This paper proceeds as follows. We first provide a brief description of the PSO algorithm. Next, the overall well-pattern description and well-pattern operators are discussed in detail. We then describe our approach for WWP. Results for several examples, including a case with geological uncertainty, are then presented. For these examples, it is demonstrated that WPO enables a wide variety of patterns to be evaluated and provides significant improvement in the objective functions.

PSO Algorithm

The PSO algorithm is a procedure for global optimization originally developed in Kennedy and Eberhardt (1995) and Eberhardt and Kennedy (1995). The algorithm is based on social behaviors observed in animal groups such as schools of fish, bird flocks, and bees. Like GAs, the algorithm is population based. In PSO, individual solutions are called particles and the collection of particles is called a swarm (these terms are analogous to "individuals" and "population" within the context of GAs). The particles interact with each other and exchange information regarding the search space. The particles can be grouped into neighborhoods within which particles interact. The specific topologies of the neighborhoods can affect the performance of the algorithm.

The particle interactions within PSO lead to a cooperative search. At each iteration, particles move to new positions in the search space on the basis of their acquired experience (i.e., previous sampling of the search space) and the collective experience of other particles in their neighborhood. The particle velocity, which is recomputed at each iteration, determines how the particle moves through search space. We will briefly describe the variant of the PSO algorithm used here. The interested reader is referred to Engelbrecht (2005), Clerc (2006a), and Poli et al. (2007) for more details about the PSO algorithm.

The PSO variant applied in this work is called a "local best PSO." The approach used here entails neighborhoods that are unique to each particle in the swarm. We denote $\mathbf{x}$ as a potential solution in the search space of a d-dimensional optimization problem, $\mathbf{x}_i(k) = [x_{i,1}(k), \ldots, x_{i,d}(k)]$ as the position of the ith particle in Iteration k, $\mathbf{x}_i^{pbest}(k)$ as the previous best solution found by the ith particle up to Iteration k, and $\mathbf{x}_i^{nbest}(k)$ as the position of the best particle in the neighborhood of Particle i at Iteration k. Note that each particle is always a member of its own neighborhood. In addition, the neighborhoods can overlap because a particle can reside in multiple neighborhoods.

At each iteration, new position vectors are determined for each particle. The new position vector of Particle i, denoted $\mathbf{x}_i(k+1)$, is computed as (Kennedy and Eberhardt 1995; Eberhardt and Kennedy 1995)

$$\mathbf{x}_i(k+1) = \mathbf{x}_i(k) + \mathbf{v}_i(k+1)\cdot\Delta t, \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

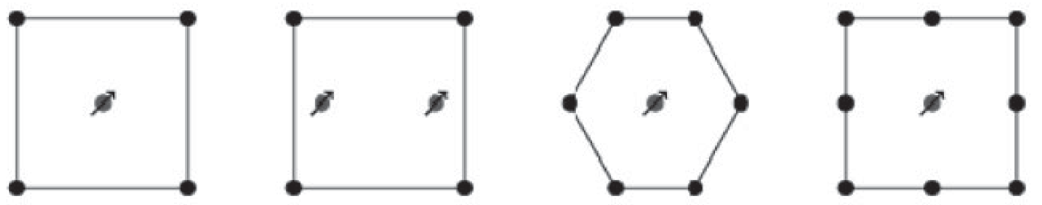
where $\mathbf{v}_i(k+1) = [v_{i,1}(k+1), \ldots, v_{i,d}(k+1)]$ is the velocity of Particle i at Iteration k+1 and Δt is a time increment. Here, consistent with standard PSO implementations, we set $\Delta t = 1$. It should be noted, however, that recent work has demonstrated improved results using variable Δt (Martinez and Gonzalo 2008, 2009), so this might be worthwhile to consider in future investigations. The elements of the velocity vector are computed as (Shi and Eberhardt 1998; Engelbrecht 2005)

$$\begin{aligned}\mathbf{v}_i(k+1) = \omega\cdot\mathbf{v}_i(k) + c_1\cdot\mathbf{D}_1(k)\times\left[\mathbf{x}_i^{pbest}(k) - \mathbf{x}_i(k)\right] \\ + c_2\cdot\mathbf{D}_2(k)\cdot\left[\mathbf{x}_i^{nbest}(k) - \mathbf{x}_i(k)\right]\end{aligned}, \quad \ldots\ldots (2)$$

where ω, c_1, and c_2 are weights; $\mathbf{D}_1(k)$ and $\mathbf{D}_2(k)$ are diagonal matrices whose diagonal components are uniformly distributed random variables in the range (0, 1); and j, $j \in \{1,2,\ldots,d\}$, refers to the jth optimization variable. In the optimizations performed in this paper, we set $\omega = 0.721$ and $c_1 = c_2 = 1.193$. These values were determined from numerical experiments performed by Clerc (2006b). We note that it is possible to optimize these parameters as part of the overall procedure, and this will be investigated in future work.

The velocity equation (Eq. 2) has three components, referred to as the inertia (term involving ω), cognitive (term involving c_1), and social (term involving c_2) components, respectively (Engelbrecht 2005). The inertia component provides a degree of continuity in particle velocity from one iteration to the next, while the cognitive component causes the particle to move toward its own previous best position. The social component, by contrast, moves the particle toward the best particle in its neighborhood. These three components perform different roles in the optimization. The inertia component enables a broad exploration of the search space, while the cognitive and social components narrow the search toward the promising solutions found up to the current iteration.

Fig. 1 shows the velocity computation and solution update in Iteration k+1 for a particle in a 2D search space. Here, $\mathbf{v}_i(k)$ is the particle's previous velocity, while $\mathbf{v}_i^c(k)$ is the velocity (cognitive)



(a) Inverted five-spot (b) Inverted six-spot (c) Inverted seven-spot (d) Inverted nine-spot

Fig. 2—Illustration of different types of well patterns. The solid black circles represent producers, and the circles with arrows represent injectors.

from the current position [$\mathbf{x}_i(k)$] to the particle's previous best position [$\mathbf{x}_i^{pbest}(k)$], and $\mathbf{v}_i^s(k)$ is the velocity (social) from the current position to the current neighborhood best position [$\mathbf{x}_i^{nbest}(k)$]. The velocity vectors $\mathbf{v}_i(k)$, $\mathbf{v}_i^s(k)$, and $\mathbf{v}_i^c(k)$ are used to compute $\mathbf{v}_i(k+1)$ according to Eq. 2. The new particle velocity, $\mathbf{v}_i(k+1)$, is added to the current position to obtain the new position vector, $\mathbf{x}_i(k+1)$, as shown in Eq. 1.

The PSO algorithm uses a cooperative search strategy that involves interaction (communication) between particles. These interactions occur between particles that reside in the same neighborhood. The grouping of particles into neighborhoods is referred to as the neighborhood topology (Engelbrecht 2005; Clerk 2006a), and various such topologies have been devised. Here we use the random-variable neighborhood topology (Clerk 2006a) in which the members of a neighborhood are determined probabilistically. The neighborhood topology is defined by a so-called adjacency matrix m_{ij}. This (nonsymmetric) matrix contains zeros and ones as entries, with an entry of one indicating that Particle i is contained in the neighborhood of Particle j. The matrix contains ones on the diagonal; otherwise, $m_{ij} = 1$ only when a generated random number is less than a specified probability p. The mean number of nonzero elements on any row (i.e., the mean number of neighborhoods each particle belongs to) is a user-specified value (the particular value determines the probability p). Here we take this value to be 3, as suggested by Clerc (2006a). The neighborhood topology is not updated if a better solution is found in the previous iteration. A particle will in general belong to multiple neighborhoods. Note that the locations of the particles in the search space do not affect the neighborhood topologies. The random-variable topology used here is robust and reduces the susceptibility of solutions to get trapped in local optima. More details on this approach can be found in Clerc (2006a) and Onwunalu (2010).

The steps of the PSO algorithm are summarized as follows:

1. Initialize all particle-position vectors randomly in the feasible space.
2. Initialize all particle-velocity components to zero.
3. Compute the objective function for all particles.
4. While not converged
   a. Reinitialize the neighborhood for all particles (if no improvement).
   b. Update the best particle position in each neighborhood.
   c. Update the previous best position of each particle.
   d. Compute the new particle positions using Eqs. 1 and 2.
   e. Compute the objective function for all particles.
5. Repeat Step 4 until termination criterion is reached.

Full details regarding the PSO algorithm used in this work can be found in Onwunalu and Durlofsky (2010) and Onwunalu (2010). We note, finally, that, rather than initialize all particle positions randomly at the start of the optimization, some number of user-defined particles (well patterns in the first phase of the optimization, new well locations in the second phase) can be specified. This enables engineering insight to be incorporated into the optimization.

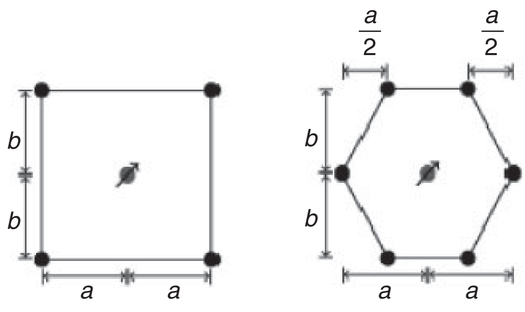
WPD

The overall WPO algorithm contains as key components the WPD, WWP, and the core optimization algorithm (PSO in this case). The WPD, which we now describe, treats well patterns (rather than individual wells) as the basic unit of representation. Thus, in our PSO implementation, each particle represents a repeated well pattern. WPD can encode representations for a wide variety of possible well patterns, in addition to the transformations that are used to manipulate these well patterns. It is the parameters that define the patterns and quantify the transformations that are optimized during the course of the optimization. As indicated in the Introduction, the WPD representation offers several benefits, including a reduction in the number of optimization variables, the ability to perform optimizations without well-to-well distance constraints, and the automatic determination of the optimum number of wells. The use of well patterns also presents some challenges. For a robust optimization, many different well-pattern types, shapes, and orientations must be considered, and there is a very large number of possible combinations of these attributes. Thus, a concise solution representation, coupled with an efficient and robust core optimization algorithm, is required for this problem.

In the WPD representation, each solution contains three groups of optimization parameters: the basic parameters associated with each of the different well patterns, parameters that quantify the pattern operations, and parameters that define the sequence of application of these operations. In the following sections, we will describe these three elements in detail.

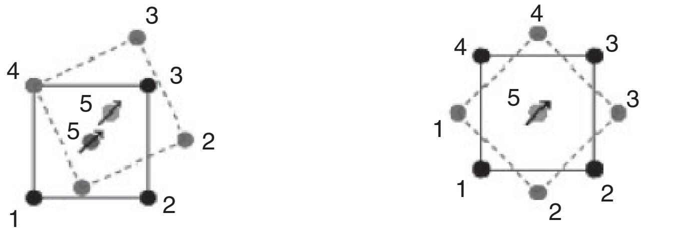
Basic Well-Pattern Parameters. In order to consider different well-pattern types in the optimization, a basic well-pattern representation is required. For this purpose, we extend the representation for inverted five-spot patterns described in Pan and Horne (1998). Our representation uses four variables to represent the well pattern, $\{\xi^0, \eta^0, a, b\}$, where (ξ^0, η^0) designate the areal location of the center of the pattern [a well may or may not be located at (ξ^0, η^0)] and a and b specify well spacings. We represent areal location using (ξ, η) rather than the usual (x, y) because $\mathbf{x}$ is used to designate PSO solutions. Our extended representation is as follows:

$$\mathcal{P} = \left\{ I^{wp}, \left[\xi^0, \eta^0, a, b \right] \right\}, \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$



(a) Inverted five-spot (b) Inverted seven-spot

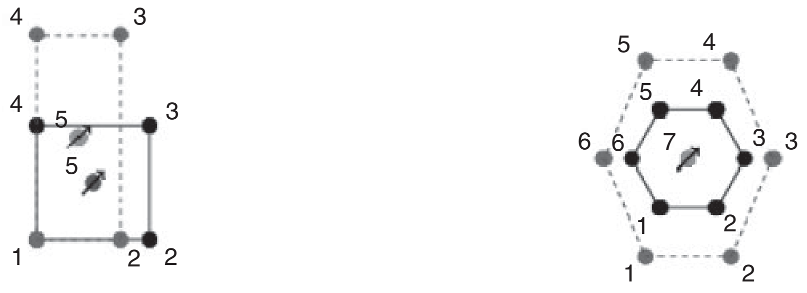
Fig. 3—Well-pattern-element representations for the inverted five- and seven-spot patterns.



(a) Rotation about Well 4 with θ=–25°

(b) Rotation about Well 5 with θ=45°

Fig. 4—Illustration of the rotation operator applied to an inverted five-spot pattern.



(a) Scaling with factors {0.75, 1.8} from Well 1

(b) Scaling with factors {1.5, 2.0} from Well 7

Fig. 5—Illustration of the scale operator for the inverted five- and seven-spot patterns.

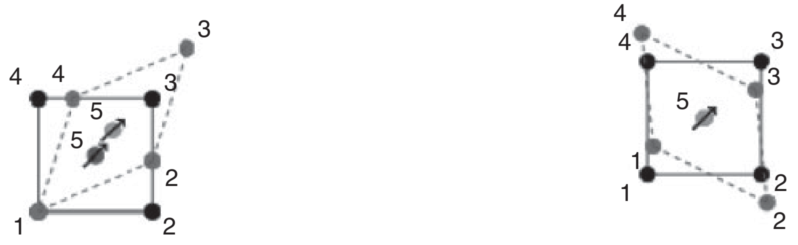
where I^{wp} is an integer variable that defines the pattern type (e.g., seven-spot, nine-spot). If we consider N_p different well-pattern types in the optimization, then $I^{wp} \in \{1,2,\ldots,N_p\}$, with each value corresponding to a different pattern type. The representation shown in Eq. 3 is quite simple and is applicable for many pattern types. This representation could be readily extended to account for more-complex well arrangements, such as 13-spot patterns.

Several well patterns, containing different numbers of wells, are shown in **Fig. 2.** It is evident that, using the representation shown in Eq. 3, each pattern can be represented in terms of the five variables appearing in the equation **(Fig. 3).** Were we to represent each well individually, a five-well pattern would require 10 parameters (ξ and η locations of each well) and a nine-well pattern would require 18 parameters. Thus, the representation in Eq. 3 clearly leads to significant reduction in the dimension of the search space.

In our algorithm, well patterns are in all cases repeated to fill the entire reservoir domain. Each pattern has the same size and orientation as the base pattern. Wells that fall outside of the reservoir boundaries are eliminated from the total set. The well-pattern parameters a and b, in addition to the parameters connected to the operators, thus determine the total number of wells associated with each PSO solution (particle). In this way, WPO determines the optimal number of wells. We constrain the minimum and maximum values of a and b such that the patterns they define are of physically reasonable sizes. The bounds prescribed for a and b depend on the bounds used for the parameters associated with the pattern operators because these also affect the size of the patterns.

The well-pattern representation in Eq. 3 does not allow for the general orientation of well patterns. In fields with large-scale permeability features or correlations, this representation may be suboptimal because patterns cannot align themselves to take advantage of trends in flow. We now describe techniques that generalize the representation given in Eq. 3.

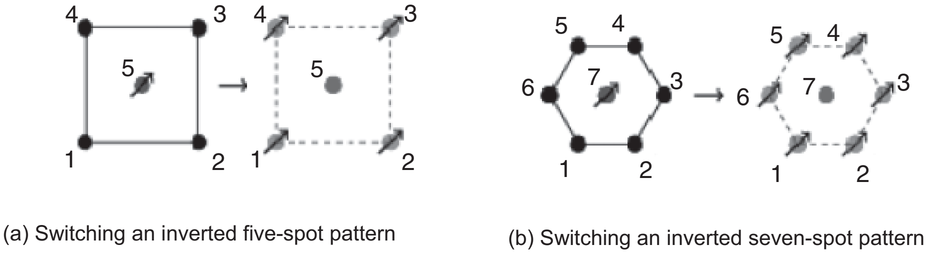
Well-Pattern Operators. Well-pattern operators define operations that can be performed on the encoded well patterns. When applied to a pattern, these operators can alter the pattern size, shape, orientation, and type (normal vs. inverted), and the location of the wells in the pattern. We developed four well-pattern operators: rotation, scale, shear, and switch operators. The rotation operator rotates a well pattern, the scale operator increases or decreases the size of a well pattern, the shear operator skews the shape of a well pattern, and the switch operator changes the pattern type from the normal to the inverted form by switching production wells to injection wells and vice versa. Other operators can be readily incorporated into the WPD representation. In general, application of these pattern operators requires the specification of several parameters, including the reference well. The reference well serves as the origin for the pattern operation, and its location remains unchanged after the operation is performed.



(a) Shearing with factors {0.45, 0.3} from Well 1

(b) Scaling with factors {−0.50, −0.10} from Well 5

Fig. 6—Illustration of the shear operator for the inverted five-spot pattern.



(a) Switching an inverted five-spot pattern

(b) Switching an inverted seven-spot pattern

Fig. 7—Illustration of the switch operator for the inverted five- and seven-spot patterns.

We now define the pattern operators and associated parameters. Each pattern has wells located at the vertices of a polygon (outer wells), and, except for the six-spot pattern, there is also a well at the center (Fig. 2). In our numbering convention, the outer wells are numbered consecutively in the counterclockwise direction followed by the interior wells.

Each pattern operator takes as input a well pattern and produces a new well pattern as output. We designate $\mathbf{W}_{\text{in}}$ to be an N×2 matrix representing the well locations in the input well-pattern element [these well locations are designated (ξ, η)] and $\mathbf{W}_{\text{out}}$ to be the corresponding matrix representing the output well locations [designated $\left(\hat{\xi},\ \hat{\eta}\right)$]. In both matrices, well locations are relative to the reference well, located at (ξ^{ref}, η^{ref}). The two matrices are given by

$$\mathbf{W}_{\text{in}} = \begin{bmatrix} \xi_1 - \xi^{\text{ref}} & \eta_1 - \eta^{\text{ref}} \\ \xi_2 - \xi^{\text{ref}} & \eta_2 - \eta^{\text{ref}} \\ \vdots & \vdots \\ \xi_n - \xi^{\text{ref}} & \eta_n - \eta^{\text{ref}} \\ \vdots & \vdots \\ \xi_{N_{wp}} - \xi^{\text{ref}} & \eta_{N_{wp}} - \eta^{\text{ref}} \end{bmatrix}$$

and

$$\mathbf{W}_{\text{out}} = \begin{bmatrix} \hat{\xi}_1 - \xi^{\text{ref}} & \hat{\eta}_1 - \eta^{\text{ref}} \\ \hat{\xi}_2 - \xi^{\text{ref}} & \hat{\eta}_2 - \eta^{\text{ref}} \\ \vdots & \vdots \\ \hat{\xi}_n - \xi^{\text{ref}} & \hat{\eta}_n - \eta^{\text{ref}} \\ \vdots & \vdots \\ \hat{\xi}_{N_{wp}} - \xi^{\text{ref}} & \hat{\eta}_{N_{wp}} - \eta^{\text{ref}} \end{bmatrix}, \quad \text{(4)}$$

where N_{wp} is the number of wells in the well pattern. Most of the well-pattern transformations can now be described through the following operation:

$$\mathbf{W}_{\text{out}}^T = \mathbf{M}\mathbf{W}_{\text{in}}^T, \quad \text{(5)}$$

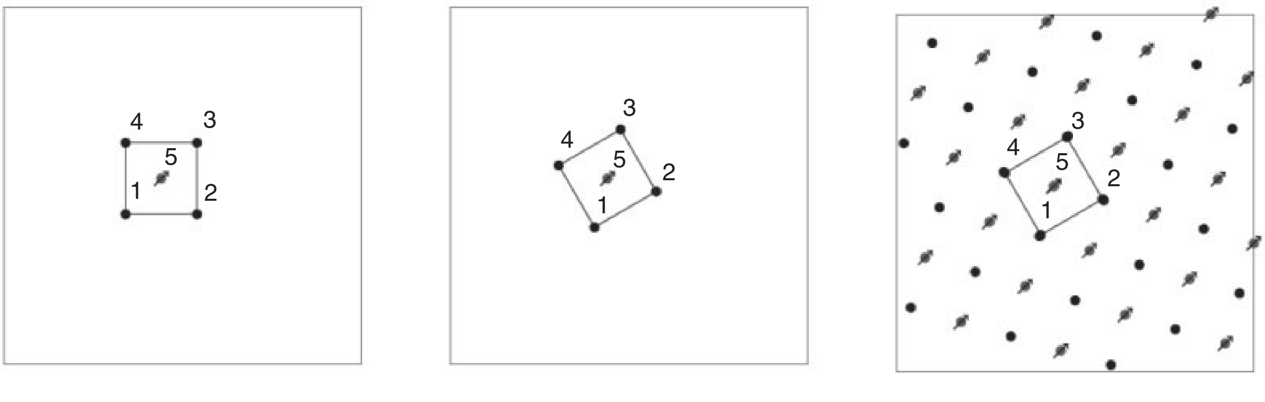
where $\mathbf{M}$ is a 2×2 transformation matrix. The specific forms of $\mathbf{M}$ for the relevant pattern operators are described later. We illustrate the well-pattern operators using the inverted five-spot and seven-spot well patterns, though the operators also apply to other well-pattern types.

Rotation Operator. The rotation operator, designated $\mathcal{O}_{\text{rot}}$, rotates a well pattern by an angle θ about a reference well, n^{ref}, $n^{\text{ref}} \in \left\{1, 2, \ldots, N_{wp}\right\}$. After the rotation, the locations of all wells other than n^{ref} are altered. The rotation operator does not change the size of the well pattern. The rotation of the pattern element is achieved through use of $\mathbf{M} = \mathbf{M}_\theta$ in Eq. 5, where

$$\mathbf{M}_\theta = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}. \quad \text{(6)}$$

This results in clockwise rotation for $\theta > 0$ and counterclockwise rotation for $\theta < 0$. **Fig. 4** illustrates the rotation operator applied to an inverted five-spot pattern. In Fig. 4a, the initial well pattern (solid lines) is rotated about Well 4 with $\theta = -25°$ (counterclockwise rotation). In the final pattern (dashed lines), the locations of Wells 1, 2, 3, and 5 differ from those in the initial pattern. Fig. 4b shows a 45° (clockwise rotation) about Well 5.

Scale Operator. The scale operator, $\mathcal{O}_{\text{scale}}$, increases or decreases the size of a well pattern. The scale operator requires as arguments the reference well in the pattern and axis scaling factors for the ξ and η directions. If the scale factor for an axis is greater than unity, the pattern is stretched in that direction. If the scale factor is less than unity, the well pattern is shrunk along that direction.

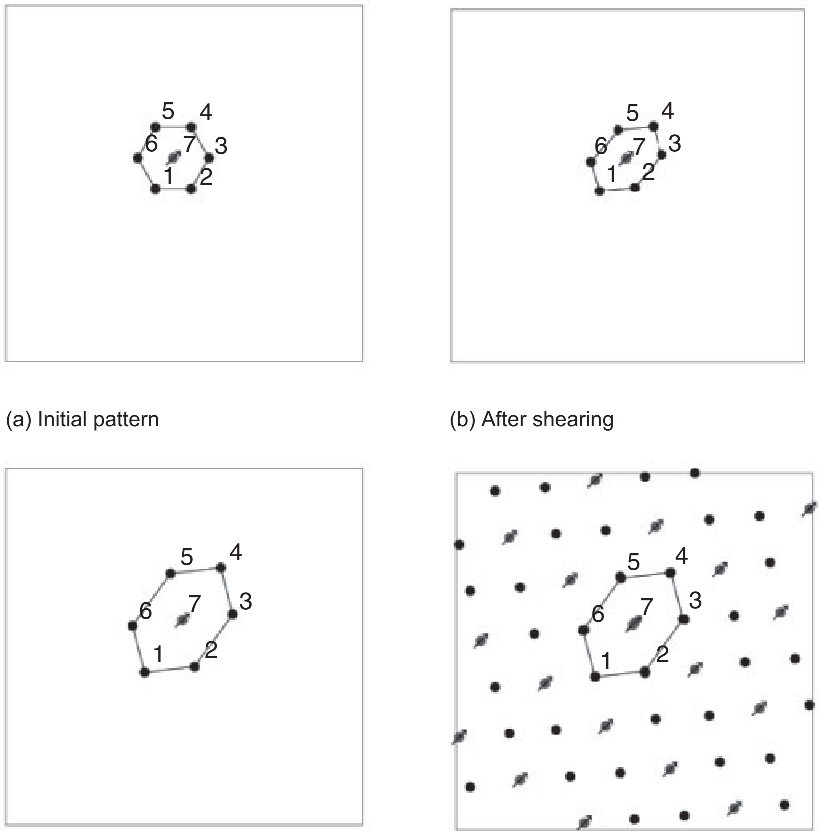


(a) Initial pattern

(b) After rotation

(c) Repeated final pattern

Fig. 8—Application of one pattern operator.



(a) Initial pattern

(b) After shearing

(c) After scaling

(d) Repeated final pattern

Fig. 9—Application of two pattern operators (shear and scale).

A nonuniform scale matrix, $\mathbf{M}_{\mathrm{sc}}$, is used to achieve the scaling of a well pattern:

$$\mathbf{M}_{\mathrm{sc}} = \begin{pmatrix} S_\xi & 0 \\ 0 & S_\eta \end{pmatrix}, \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

where S_ξ and S_η are axis scaling factors. **Figs. 5a and 5b** illustrate the scale pattern operator applied to the inverted five-spot and inverted seven-spot patterns, respectively. In Fig. 5a, the well pattern is scaled relative to Well 1 using scaling factors {0.75, 1.8}. In Fig. 5b, the inverted seven-spot pattern is scaled with factors {1.5, 2.0} relative to Well 7. Because the pattern is replicated over the entire field, it is clear that these scaling parameters will have a strong effect on the total number of wells. In the examples, the scaling factors are constrained to be between 0.5 and 2.

TABLE 1—ECONOMIC PARAMETERS FOR NPV COMPUTATION	
Well cost	3×10^6 (USD)
Oil price (Examples 1, 3, 4)	60 (USD/STB)
Oil price (Example 2)	80 (USD/STB)
Gas price (Example 2)	2.5 (USD/MSCF)
Water-production cost (Examples 1, 3, 4)	5 (USD/STB)
Water-injection cost (Examples 1, 3, 4)	5 (USD/STB)
Water-production cost (Example 2)	10 (USD/STB)
Water-injection cost (Example 2)	10 (USD/STB)
Discount rate, *r*	0.10

Shear Operator. The shear operator, $\mathcal{O}_{\mathrm{shear}}$, alters the shape of a well pattern by shearing (skewing) the well pattern in the ξ and η directions. The shear operator requires three arguments: a reference well and axis shearing factors for the ξ and η directions. These factors indicate the amount of shearing in each direction relative to the other direction. The shearing of the pattern element is achieved using a shear matrix, $\mathbf{M}_{\mathrm{sh}}$:

$$\mathbf{M}_{\mathrm{sh}} = \begin{pmatrix} 1 & H_\xi \\ H_\eta & 1 \end{pmatrix}, \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

where H_ξ and H_η are axis shearing factors. Care must be taken in defining the minimum and maximum values of H_ξ and H_η because

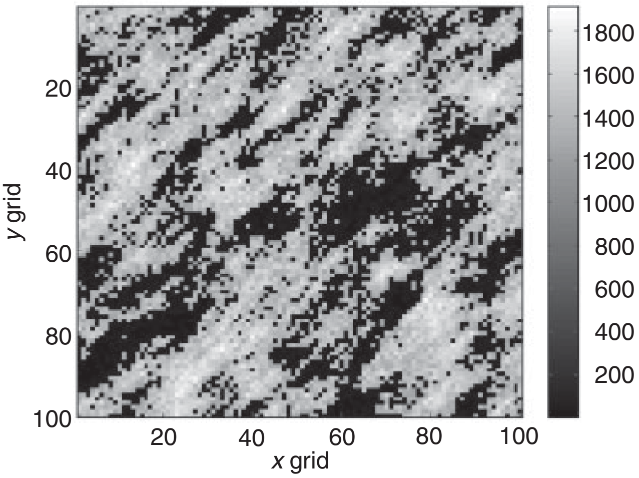
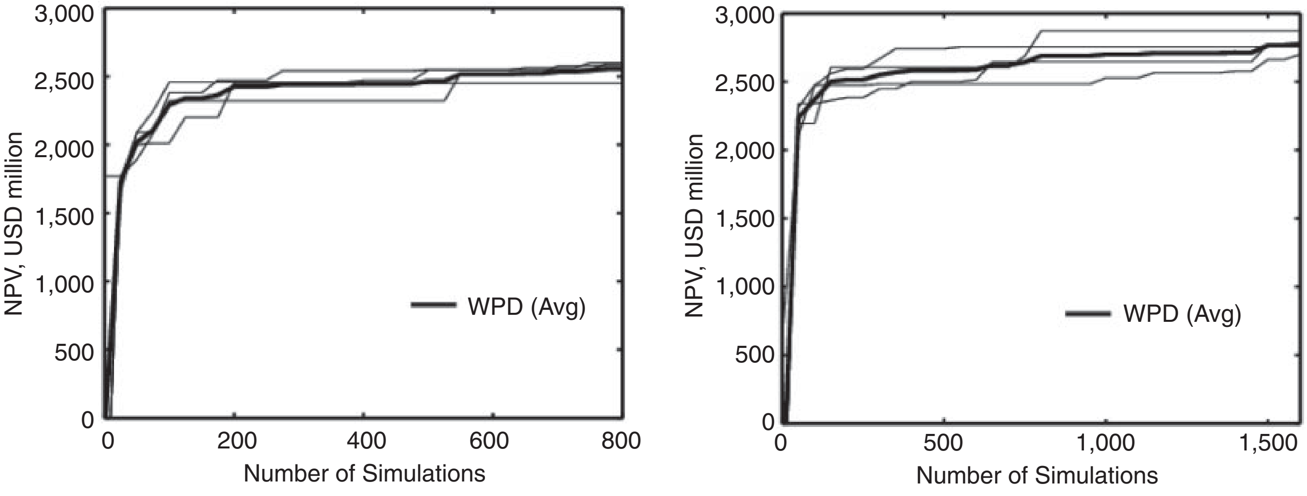


Fig. 10—Permeability field (Example 1).



(a) One pattern operator

(b) Four pattern operators

Fig. 11—NPV of the best solutions vs. number of simulations for the four optimization runs using one and four pattern operators (Example 1).

well locations become colinear if H_ξ and H_η approach −1 or 1. In the examples, the shearing factors are constrained to be between −0.5 and 0.5. **Fig. 6** illustrates the shear operator applied to an inverted five-spot pattern.

Switch Operator. The switch operator, $\mathcal{O}_{\text{switch}}$, switches a well pattern from the normal to the inverted form and vice versa. This is achieved by switching the type (producer, injector) of all the wells in the pattern. The switch operator does not require any arguments.

The switch operator offers some benefits for the overall WPO algorithm. It enables the algorithm to consider both normal and inverted forms of the patterns without increasing the number of patterns that need to be defined in the algorithm. It also allows the algorithm to consider different producer-/injector-well ratios for the same well-pattern parameters. For example, the normal seven-spot and inverted seven-spot patterns have producer/injector ratios of 1:2 and 2, respectively (Craig 1971). **Figs. 7a and 7b** illustrate the switch pattern operator applied to the inverted five-spot and inverted seven-spot patterns, respectively.

Representation of Well-Pattern Operators. As described, each pattern operator (except the switch operator) requires the specification of a reference well in the pattern and at most two additional operator arguments. This allows us to use a simple generic representation for these pattern operators that can be readily extended to other operators that may be introduced.

Let $\mathcal{O}_j$ represent the jth pattern operator, where $\mathcal{O}_j \in \{\mathcal{O}_{\text{rot}}, \mathcal{O}_{\text{scale}}, \mathcal{O}_{\text{shear}}, \mathcal{O}_{\text{switch}}\}$. In the WPD representation, $\mathcal{O}_j$ is represented as

$$\mathcal{O}_j = \left[\underbrace{\{n_j^{\text{ref}}\}}_{\text{reference well}}, \underbrace{\{\arg_{j,1}, \arg_{j,2}\}}_{\text{operator arguments}} \right], \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

where n_j^{ref} is the reference well for Operator j and $\{\arg_{j,1}, \arg_{j,2}\}$ is the list of arguments for Operator j.

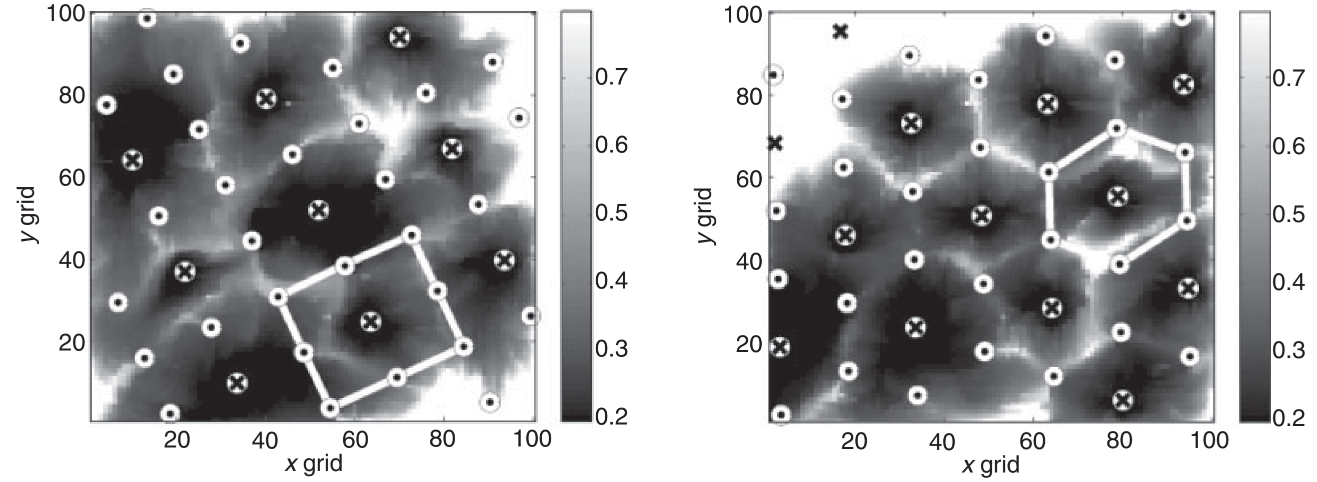
In our implementation, the arguments appearing in Eq. 9 are represented as normalized variables between zero and unity. Each variable is then rescaled as required before being used in the

TABLE 2—OPTIMIZATION RESULTS USING WPD WITH ONE PATTERN OPERATOR (EXAMPLE 1)

			Well Count	
Run	Best Pattern	NPV (million USD)	Producers	Injectors
1	Inverted 9-spot	2,591	26	10
2	Inverted 7-spot	2,449	28	14
3	Inverted 6-spot	2,575	28	9
4	Inverted 9-spot	2,597	30	9
Average		2,553		

TABLE 3—OPTIMIZATION RESULTS USING WPD WITH FOUR PATTERN OPERATORS (EXAMPLE 1)

			Well Count	
Run	Best Pattern	NPV (million USD)	Producers	Injectors
1	Inverted 9-spot	2,754	30	9
2	Inverted 7-spot	2,872	28	14
3	Inverted 9-spot	2,698	33	9
4	Inverted 9-spot	2,773	28	8
Average		2,774		



(a) One pattern operator

(b) Four pattern operators

Fig. 12—Well locations of the best solutions using one operator and four operators (Example 1). Oil saturation at *t* = 500 days shown as background.

pattern operation. This enables appropriate exchange of information between particles that are acted on by different operators during the course of the particle-swarm optimization.

Solution Representation in WPD. We now describe the overall representation of potential solutions using the well-pattern description. Each solution (particle within our PSO implementation) consists of the basic well-pattern definition and parameters (Eq. 3) and the representation for the pattern operators (Eq. 9). In addition, each solution contains a set of variables that defines the sequence of pattern operations when multiple operators are applied. The *i*th PSO particle, $\mathbf{x}_i$, is thus encoded as

$$\mathbf{x}_i = \left[\underbrace{\left\{I_i^{wp},\ \left[\xi_i^0,\eta_i^0,a_i,b_i\right]\right\}}_{\text{pattern parameters}}\underbrace{\left\{S_{i,1},S_{i,2},\ldots,S_{i,\mathcal{N}_o}\right\}}_{\text{operator sequence}}\underbrace{\left\{\mathcal{O}_{i,1},\mathcal{O}_{i,2},\ldots,\mathcal{O}_{i,\mathcal{N}_o}\right\}}_{\text{pattern operators}}\right], \qquad (10)$$

where $\left\{I_i^{wp},\left[\xi_i^0,\eta_i^0,a_i,b_i\right]\right\}$ are the pattern parameters for Particle i, $\mathcal{N}_o$ is the number of pattern operators, $\left\{\mathcal{O}_{i,1},\mathcal{O}_{i,2},\ldots,\mathcal{O}_{i,\mathcal{N}_o}\right\}$ provides the list of pattern operators, and $\left\{S_{i,1},S_{i,2},\ldots,S_{i,\mathcal{N}_o}\right\}$ represents the sequence of application of the pattern operators. Each $S_{i,j}$, $S_{i,j} \in \left\{0,1,2,\ldots,\mathcal{N}_o\right\}$, is an integer variable representing the index of a pattern operator. For example, if $S_{i,1} = 1$ and $S_{i,2} = 2$, then the pattern operator $\mathcal{O}_{i,1}$ is applied first and pattern operator $\mathcal{O}_{i,2}$ is applied second (using the well pattern generated from $\mathcal{O}_{i,1}$). If $S_{i,j} = 0$, then the jth pattern operator ($\mathcal{O}_{i,j}$) is skipped.

All components of any particle $\mathbf{x}_i$, which represents a PSO solution, are treated as real numbers. Some of the optimization parameters, such as n_j^{ref} and $S_{i,j}$, however, are integers. Where necessary, we determine integer values from real values by simply rounding to the nearest integer.

Examples of using one pattern operator and two pattern operators in sequence are illustrated in **Figs. 8 and 9,** respectively. In Fig. 8, the rotation operator is applied to the initial well pattern (Fig. 8a) and the resulting pattern (Fig. 8b) is repeated over the entire domain (Fig. 8c). In the example in Fig. 9, the shear and scale operators are applied in sequence to an inverted seven-spot pattern. As indicated, wells that fall outside the reservoir boundaries are eliminated. It is evident from these figures that a wide variety of well patterns can be generated (and thus evaluated in the WPO procedure) using the WPD representation.

WWP

Optimization using WPD produces solutions that consist of repeated well patterns. It is possible to improve the solution further by performing optimizations that involve perturbing the

TABLE 4—NPV OF UNOPTIMIZED STANDARD WELL PATTERNS FOR DIFFERENT WELL SPACINGS*

			Well Count	
Spacing	Pattern	NPV (million USD)	Producers	Injectors
30 acres	Inverted 5-spot	1,432	25	25
	Inverted 6-spot	749	25	45
	Inverted 7-spot	−991	65	30
	Inverted 9-spot	−1,321	75	25
40 acres	Inverted 5-spot	1,895	16	16
	Inverted 6-spot	1,557	48	16
	Inverted 7-spot	245	44	20
	Inverted 9-spot	805	48	16
50 acres	Inverted 5-spot	2,151	16	16
	Inverted 6-spot	1,780	16	28
	Inverted 7-spot	467	40	20
	Inverted 9-spot	707	16	28

* The well patterns are aligned with the reservoir boundaries (Example 1).

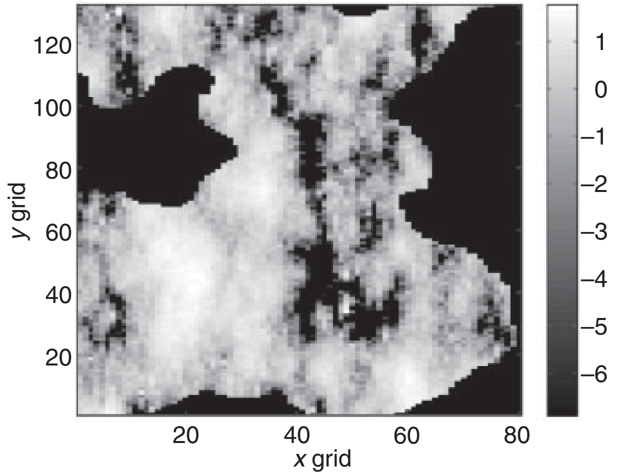


Fig. 13—Logarithm of permeability field (Example 2).

well locations determined using WPD. We refer to this as WWP. Following WWP, the basic patterns remain essentially intact but the well locations within the patterns are shifted to improve the objective function.

We again use the PSO algorithm for the WWP optimization. Here, however, each PSO particle $\mathbf{x}_i$ contains a concatenation of the perturbations to be applied to the well locations determined from WPD:

$$\mathbf{x}_i = \left\{ \underbrace{\Delta\xi_1, \Delta\eta_1}_{\text{Well 1}}, \underbrace{\Delta\xi_2, \Delta\eta_2}_{\text{Well 2}}, \ldots, \underbrace{\Delta\xi_n, \Delta\eta_n}_{\text{Well } n}, \ldots, \underbrace{\Delta\xi_N, \Delta\eta_N}_{\text{Well } N} \right\}, \qquad (11)$$

where N is the total number of wells as determined in the WPD optimization and $\Delta\xi_n$ and $\Delta\eta_n$ represent the perturbations of the spatial locations of Well n. Before simulation, the actual well locations are obtained by adding the perturbations to the corresponding well location from the WPD solution. In our implementation, the minimum and maximum values of $\Delta\xi_n$ and $\Delta\eta_n$ are constrained to keep wells from shifting from one pattern to another. We note that WWP could also be used for other determinations, such as the completion interval for each well in 3D problems. Elimination of wells could also be considered through inclusion of an active/inactive optimization parameter for each well. Neither of these options was considered in this work, though they could be incorporated easily into the WWP procedure (the dimension of the search space, however, will increase).

The WWP procedure introduces an efficient local search, which leads to improved solutions (as will be demonstrated). Improved solutions are achieved because the optimized well locations now account for local variations in porosity, permeability, and other properties. The two procedures—WPD and WWP—are complementary because WPD provides the ability to search efficiently on the large scale and to optimize well count, while WWP enables local adjustments.

Although the dimension of the search space in WWP is the same as that using well-by-well concatenation (for cases where the number of wells is specified), WWP has several advantages over well-by-well concatenation. Specifically, in WWP, wells are allowed to move only a limited number of gridblocks in each direction. This constraint can be incorporated easily into the optimization, in contrast to the general well-distance constraints required when using well-by-well concatenation. In addition, because wells can move only locally in WWP, the size of the search space is much smaller than that for well-by-well concatenation (despite the fact that the dimension of the search space is the same in both cases). Finally, in WWP the number of wells (as determined from the WPD optimization) is fixed and does not need to be determined as part of the optimization. Using well-by-well concatenation, however, the number of wells should also be an optimization variable, which further complicates the optimization.

It is important to note that the use of WPD followed by WWP cannot be expected to provide the overall global optimum that could (theoretically) be achieved through use of well-by-well concatenation. This is because well-by-well concatenation entails a broader search, which should ultimately lead to a better global optimum than that resulting from the use of WPD plus WWP. However, the WPD-plus-WWP search is much more efficient, so this approach is expected to provide better solutions given practical computing resources.

Examples

We now apply the WPO procedure to four example problems. In all cases, we maximize NPV using the procedures described earlier. The economic parameters used for the computation of NPV are provided in **Table 1.** Simulation runs for Examples 1, 3, and 4 are performed using Stanford's General Purpose Research Simulator, GPRS (Cao 2002; Jiang 2007). The 3DSL streamline simulator (3DSL 2006) is used for Example 2. Because of the stochastic nature of the PSO algorithm, we perform multiple optimization runs with the same set of input parameters. This enables us to gauge the degree of variability in the optimization results. In the figures in this section, *x*-grid and *y*-grid refer to gridblock locations in the simulation models.

Example 1: WPD Optimizations Using Different Numbers of Operators. In this example, we perform optimizations using either one or four WPD operators per particle. We optimize using only WPD; the WWP optimization is not applied for this case.

We consider a synthetic, heterogeneous, 2D reservoir model containing 100×100 gridblocks, with each block being 100×100×40 ft. The permeability field, shown in **Fig. 10,** was generated geostatistically using an exponential variogram model with oriented correlation lengths of 1,000 and 5,000 ft. Porosity varies from block to block and is correlated with permeability. The reservoir model initially contains oil and water ($S_{o,i} = 0.80$, $S_{w,i} = 0.20$). The oil viscosity μ_o is 1.20 cp, and oil compressibility c_o is 2.0×10^{-5} psi^{-1}. For water, we specify $\mu_w = 0.31$ cp and $c_w = 2.9 \times 10^{-6}$ psi^{-1}. Relative permeability endpoints for oil and water are 0.85 and 0.30, respectively. The initial reservoir pressure is 5,000 psi. The production and injection wells operate under bottomhole pressure

TABLE 5—OPTIMIZATION RESULTS USING WPD WITH FOUR PATTERN OPERATORS (EXAMPLE 2)

			Well Count	
Run	Best Pattern	NPV (million USD)	Producers	Injectors
1	Inverted 5-spot	1,377	16	15
2	Inverted 5-spot	1,459	15	15
3	Inverted 5-spot	1,460	15	15
4	Inverted 5-spot	1,372	15	15
5	Inverted 5-spot	1,342	13	15
Average		1,402		

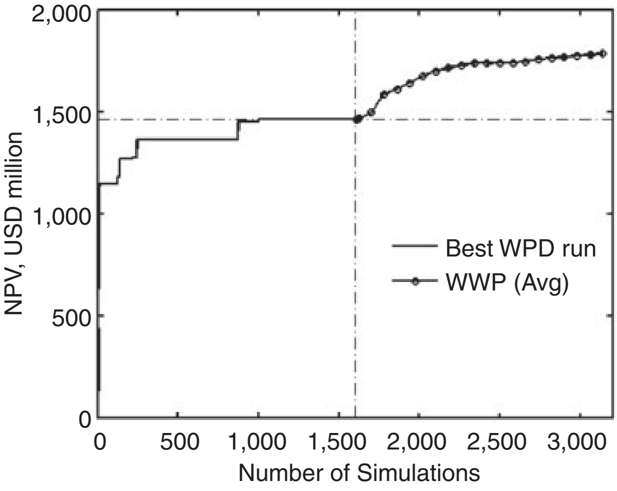


Fig. 14—NPV of best result from WPD and average NPV of the WWP solutions vs. number of simulations (Example 2).

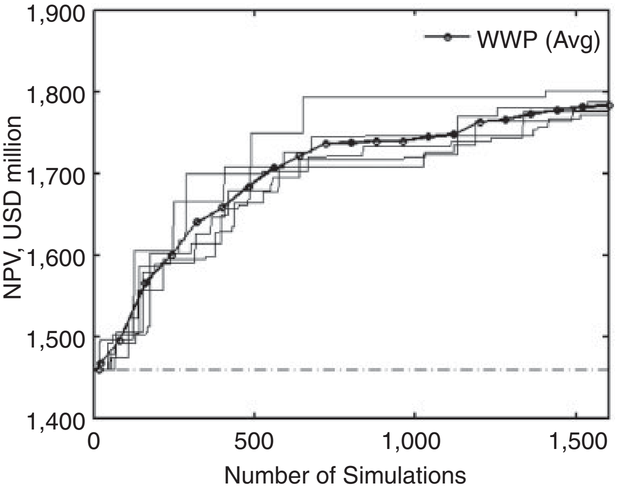


Fig. 15—NPV of the best WWP solutions vs. number of simulations (Example 2).

(BHP) constraints of 1,000 and 6,000 psi, respectively. The total production time is 3,650 days.

The four well-pattern types shown in Fig. 2 are considered in the optimizations. We perform optimizations using one and four operators. For the optimizations that apply one operator per particle, we use 20 PSO particles. In these optimizations, although each particle has only one pattern operation performed on it, the particular operator varies with particle and iteration. For the optimizations that apply four pattern operators per particle, we use 40 particles. More particles are used in the optimizations with four operators because the number of variables is approximately twice that of the optimizations with one operator. Four optimizations are performed in each case, and each optimization is run for 40 iterations. Function evaluations are performed in parallel using a cluster of up to 40 processors.

Figs. 11a and 11b show the evolution of the NPV of the best development scenario vs. number of simulations for the optimizations using one and four operators. Each thin curve corresponds to a different optimization run, and the heavy curve depicts the average of the best solutions from the four runs. NPV clearly improves with iterations, with the largest improvement coming at early iterations. **Tables 2 and 3** summarize the results for the optimization runs with one and four operators, respectively. In the optimizations with one operator, the inverted nine-spot is the best pattern, with an NPV of USD 2,597 million (Table 2). This development scenario has 30 producers and nine injectors. For the optimizations with four operators, the inverted seven-spot pattern, containing 28 producers and 14 injectors, gives the best scenario with an NPV of USD 2,872 million (Table 3).

Although the results in Tables 2 and 3 suggest that the use of four operators provides generally better NPVs than those using one operator, it must be kept in mind that twice as many simulations are performed in the four-operator cases than in the one-operator cases. However, assessment of NPVs for the four-operator runs after 800 simulations indicates that these, on average, are superior to those from the one-operator runs after 800 simulations (average NPV of USD 2,688 million for the four-operator runs vs. USD 2,553 million for the one-operator runs). The maximum NPV for the four-operator runs after 800 simulations (USD 2,872 million) is also considerably higher than that from the one-operator runs (USD 2,597 million). Therefore, in all subsequent WPD optimizations, four operators will be used.

Figs. 12a and 12b show the well locations from the best solutions in the optimizations with one and four operators, respectively. The black dots inside white circles represent producer wells, while a white circle with a black X designates an injection well. In these figures, the basic pattern element is depicted by the white lines. The oil saturation at 500 days is shown as background. It is evident that the best patterns, in both cases, are rotated with respect to the reservoir boundaries. This results from the effect of reservoir heterogeneity on the flow field. We note that, based on the oil-saturation maps, both patterns appear to provide efficient sweep (recovery factor is very high at the end of the runs in both cases). Because the optimizations seek to maximize NPV rather than cumulative oil produced, a number of other factors, such as well cost, the costs associated with water injection and production, and the timing (because of discounting) of costs and revenues, have a strong effect on the determination of the optimal scenario.

Next, we compare these results with those obtained using standard well patterns (no optimization is performed). Results for standard patterns aligned with the reservoir boundaries are presented in **Table 4.** We consider well patterns with spacings from 20 to 50 acres. This range was determined on the basis of the bounds specified for a and b and for the pattern operator parameters. Results for standard patterns with 20-acre spacings give negative NPVs and are not presented in the table. It is clear that the optimization results are significantly better than those for the standard patterns, which highlights the potential benefits of using our procedure for large-scale optimizations.

TABLE 6—OPTIMIZATION RESULTS USING THE WWP PROCEDURE (EXAMPLE 2)

		Increase Over WPD	
Run	NPV (million USD)	(million USD)	%
1	1,777	317	21.7
2	1,787	327	22.4
3	1,776	316	21.6
4	1,801	341	23.4
5	1,771	311	21.3
Average	1,782	322	22.1

Example 2: WPD and WWP Optimizations in Reservoir With Irregular Boundary. Here, we apply the WPD and WWP procedures to maximize NPV in a reservoir with irregular boundaries. The simulator used for this case is 3DSL (3DSL 2006). The 2D synthetic reservoir model contains 80×132 gridblocks, with each block being 250×200×10 ft. **Fig. 13** depicts the logarithm of the permeability field. Net/gross ratio varies from block to block, with blocks outside the boundary (feasible region) having zero net/gross. The reservoir initially contains oil, gas, and water ($S_{o,i}$ = 0.80, $S_{g,i}$ = 0.01, $S_{w,i}$ = 0.19). For fluid viscosities, we specify μ_o = 1.2 cp, μ_g = 0.01 cp, and μ_w = 0.31 cp. Relative permeability endpoints for oil and water are 1.0 and 0.1, respectively. The initial reservoir pressure is 2,700 psi. The production and injection wells operate under BHP constraints of 1,200 and 2,900 psi, respectively. The total production time is 1,825 days.

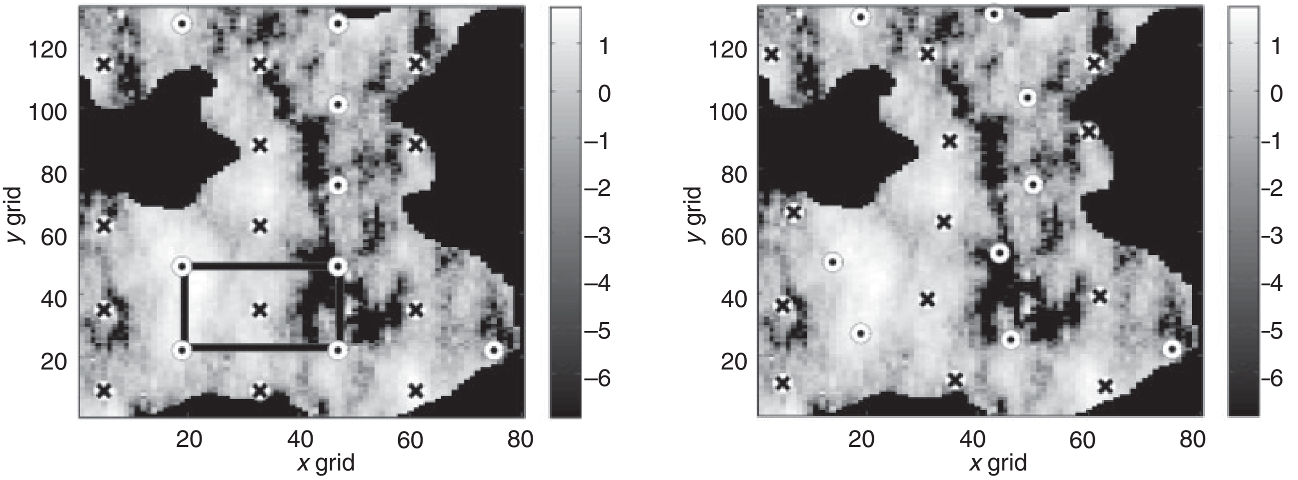


Fig. 16—Well locations for the best WPD and WWP solutions. Logarithm of permeability field is shown as background (Example 2).

All of the pattern types shown in Fig. 2 are considered in the WPD optimizations. Each optimization run uses 40 particles with four pattern operators applied. The runs proceed for 40 iterations. Five such WPD runs are performed. As is evident in Fig. 13, the region in which wells can be placed is irregular. In the WPD optimizations, well patterns are still replicated throughout the reservoir. Wells that fall outside the boundaries are eliminated from the total set, so only those wells located in the reservoir are included in the simulations.

Following the five WPD runs, we perform five WWP optimizations. The WWP optimizations are based on the best solution from the WPD runs (meaning the perturbations are computed with respect to the best configuration determined from the five WPD optimizations). In the WWP runs, the optimization parameters are defined such that the wells always fall within the feasible region after the perturbations.

Table 5 presents the results for the five WPD optimizations. The results for the five runs are quite consistent, with the inverted five-spot found to be the best pattern in all cases. The maximum NPV for the five runs is USD 1,460 million (Run 3).

Using the best pattern from Run 3, five WWP optimizations are then performed. The results are presented in **Fig. 14,** where the best WPD solution is shown (thick solid line, corresponding to the first 1,600 simulations) along with the average of the five WWP runs (solid line with circles). It is clear that NPV increases in both phases (WPD and WWP) of the optimization and that WWP provides clear improvement in the WPD results. Results from all five WWP runs are shown in **Fig. 15,** where the NPV of the best scenario in each run (thin lines) is displayed along with the average curve. It is evident that all of the WWP runs provide an increase in NPV relative to the best WPD solution (dot-dash line). Results from the five WWP runs are summarized in **Table 6.** The maximum NPV is USD 1,801 million, which represents an increase of USD 341 million (23.4%) over the best WPD result (Run 3 in Table 5).

Figs. 16a and 16b show the well locations from the best WPD and WWP optimization runs. Although the perturbations evident in Fig. 16b do not appear that dramatic, this configuration results in a significant improvement in NPV over the unperturbed configuration.

Finally, we note that solving this optimization problem using a traditional approach (i.e., through use of concatenation of well parameters) will present some difficulties. For example, constraints must be introduced to keep wells within the feasible region and to satisfy minimum well-to-well distance requirements. Incorporation of these constraints into the optimization may limit the effectiveness of standard algorithms, particularly for large numbers of wells.

Example 3: WPD and WWP Optimizations Over Multiple Reservoir Models. In this example, we account for geological uncertainty by performing the optimizations over five realizations of the reservoir model. In each phase of the optimization, we optimize expected NPV, which is simply the average of the NPVs over the five models. The reservoir model consists of 63×63 blocks, and each gridblock is 100×100×30 ft. The permeability fields were generated geostatistically using an exponential variogram model with correlation length of 1,000 ft. Porosity varies from block to block and is correlated with permeability.

We perform four WPD optimizations with four pattern operators. Using the best solution from the WPD runs, we then perform four WWP optimizations. Each optimization run contains 40 particles and is run for 40 iterations.

Table 7 shows the optimization results from the WPD runs. The best pattern is consistently a normal nine-spot pattern. Note that, although WPD encodes only the inverted forms of the well patterns, the optimization consistently switches from the inverted to the normal nine-spot. The best scenario (Run 2) has an expected NPV of USD 832 million and contains seven producers and 29 injectors.

Fig. 17 shows the evolution of NPV in the best WPD run and the average NPV of the four subsequent WWP runs. Again, NPV increases with iteration during both phases of the optimization. The increase in NPV using WWP is substantial in the first few iterations. **Fig. 18** shows the evolution of NPV for the best scenarios

TABLE 7—OPTIMIZATION RESULTS USING WPD WITH FOUR PATTERN OPERATORS (EXAMPLE 3)

			Well Count	
Run	Best Pattern	NPV (million USD)	Producers	Injectors
1	Normal 9-spot	705	8	30
2	Normal 9-spot	832	7	29
3	Normal 9-spot	723	9	27
4	Normal 9-spot	757	7	30
Average		754		

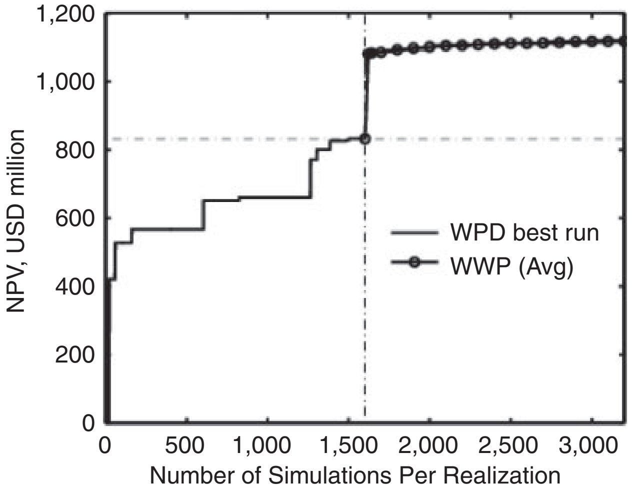


Fig. 17—NPV of best result from WPD and average NPV of the WWP solutions vs. number of simulations per realization (Example 3).

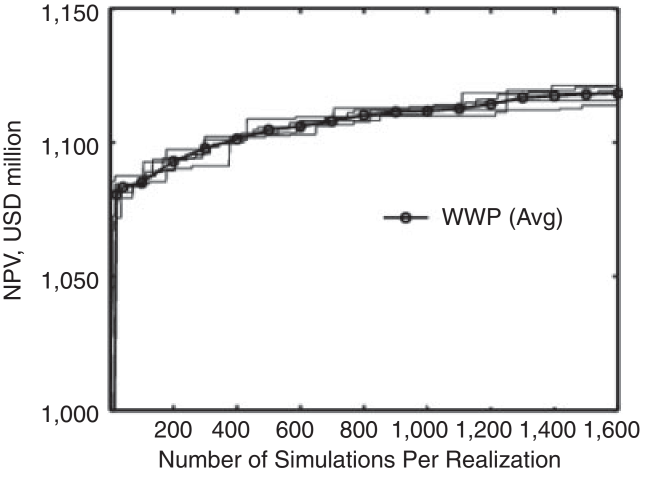


Fig. 18—NPV of the best WWP solutions vs. number of simulations per realization (Example 3).

in the four WWP optimizations (thin lines) along with the average result. There is very little variation between the four WWP optimizations. **Table 8** summarizes the results of the WWP optimizations. The WWP procedure improves the NPV in each case, with a maximum increase in NPV of USD 290 million (34.9%) over the best WPD result.

Figs. 19a and 19b show the well locations for the best solutions from the two phases of the optimization, with the permeability field of one of the realizations shown in the background. The degree of perturbation of the well locations, evident in Fig. 19b, is greater than that observed in Example 2. **Fig. 20** shows the NPVs for each of the five realizations. We see that the use of WWP provides improvement for all realizations.

Example 4: Comparison of WPO to Well-by-Well Concatenation. In this example, we compare optimizations using WPO to optimizations using concatenated well parameters. We use the same reservoir model and economic parameters as in Example 1. We first perform five WPD optimizations using four pattern operators. Then, as in Examples 2 and 3, we perform five WWP optimizations using the best WPD solution. We also perform five optimizations with concatenated well parameters using the same number of wells as in the best WPD solution.

In the optimizations using the well-by-well concatenation, we determine both well type and well locations. Well-to-well distance constraints are not included in the optimizations using concatenation. This is because, in problems with well-to-well distance constraints and relatively large numbers of wells, a significant fraction of the solutions can be infeasible, which reduces the effective swarm size and, thus, limits the performance of the PSO algorithm (we note that this issue could be addressed through the implementation of better constraint-handling techniques). It is not clear how the lack of well-to-well distance constraints will affect the well-by-well concatenation results. On the one hand, the actual global optimum for the unconstrained problem cannot be less than that for the constrained problem. On the other hand, however, the use of constraints may enable a more efficient search. The approach used here does not require the determination of the optimal number of wells in the well-by-well concatenation runs, so, in this sense, the problem setup provides an advantage to this approach. Other comparisons between the two techniques are, of course, also possible.

TABLE 8—OPTIMIZATION RESULTS USING THE WWP PROCEDURE (EXAMPLE 3)

		Increase Over WPD	
Run	NPV (million USD)	(million USD)	%
1	1,116	284	34.1
2	1,122	290	34.9
3	1,119	287	34.5
4	1,120	288	34.4
5	1,115	283	34.0
Average	1,184	286	34.4

In all optimizations, we use the PSO algorithm with 40 particles in each iteration. We run each WPD optimization for 20 iterations and then, using the best WPD result, run WWP for 20 iterations. This results in a total of 1,600 simulations for each full WPO optimization. Although we use the same parameters here as in Example 1 for the WPD portion of the runs, this case was run separately, and five optimization runs were performed (in contrast to four optimizations used in Example 1). Thus, these WPD results differ slightly from those presented earlier. In the optimizations using the concatenated well parameters, we run the optimizations with the specified number of wells until we have performed 1,600 simulations of feasible scenarios (i.e., cases with invalid well configurations are not included in the simulation count).

Fig. 21 shows the evolution of the NPV of the best WPD solution and the average of the five subsequent WWP optimizations. The average of the five well-by-well concatenation runs is also presented. The WPO procedure provides, on average, a better solution than the well-by-well concatenation approach. We note that the best result (of the five runs) using the concatenation approach gave an NPV of USD 3,575 million. The five WWP runs (starting from the best WPD optimization) provided NPVs ranging from USD 3,667 million to USD 4,022 million. Thus, for this particular example, we observe that WPO consistently outperforms well-by-well concatenation. This demonstrates that WPO indeed provides a useful and concise solution representation.

Conclusions

In this paper, we developed and applied a new procedure for optimizing well placement in large-scale field developments involving many wells. The new algorithm, called WPO, consists of a WPD incorporated into a core optimization algorithm. In the well-pattern description, each solution consists of a representation of a particular well pattern along with pattern operators that alter the size, shape, and orientation of the pattern. Many different well patterns can be considered within WPD. It is the parameters associated with the pattern descriptions and operators that are determined during the optimizations. The encoded well patterns are repeated across the field, which enables the optimum number of wells to be determined as part of the solution. A desirable feature of WPD is that the computational complexity of the optimization is essentially independent of the number of wells introduced.

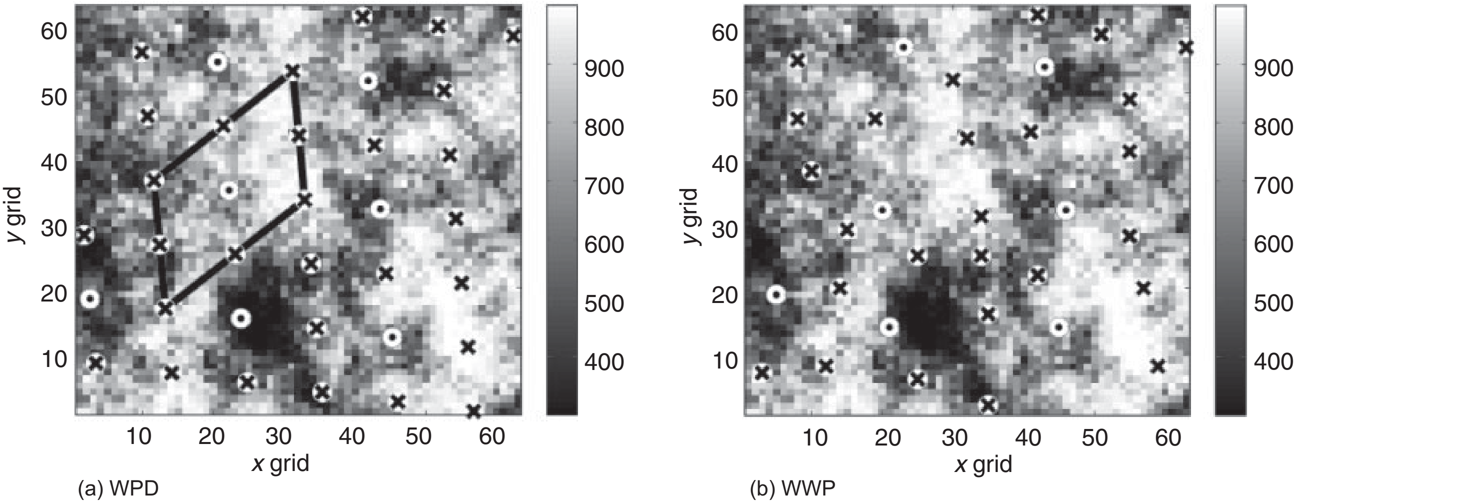


Fig. 19—Well locations for the best WPD and WWP solutions. Permeability field for one of the realizations is shown as background (Example 3).

A WWP procedure was also developed. WWP, which can be applied as an optional second phase of the optimization, entails a local perturbation of the well locations obtained from the WPD optimization. For the underlying (core) optimization algorithm, we used PSO. PSO has recently been demonstrated to be an efficient algorithm for optimizing well locations in subsurface-flow applications (Matott et al. 2006; Onwunalu and Durlofsky 2010).

The optimization procedure was applied to four example cases. Several variants were considered, including the use of one vs. four operators for each potential solution and the use of WWP following optimization using WPD. The overall optimization was shown to result in significant increases in the objective function, particularly at early iterations, in all cases. In one example, the WPO results for NPV were compared to those for standard well patterns of various sizes. The NPVs using WPO were seen to be significantly larger than those for standard well patterns, highlighting the potential benefit of the algorithm for identifying promising development scenarios. Significant improvement in NPV was obtained by performing WWP optimizations on the best solution obtained using WPD. For the two examples in which WWP was applied, average improvements in NPV of 22 and 34% over the best WPD solutions were achieved. We also compared WPO results to those obtained from optimizations using concatenated well parameters and found that the WPO procedure provided better solutions.

In future work, it will be useful to improve the efficiency of the WPO algorithm through use of surrogate (proxy) simulation models. This will act to reduce the number of time-consuming simulations required. It will also be of interest to investigate the use of specialized PSO particle neighborhoods and to consider other variants of the two-phase (WPD followed by WWP) optimization strategy applied here. Use of a hybrid optimization approach involving the combination of PSO with a local optimizer may also prove effective. Finally, it may be useful to apply a metaoptimization procedure to determine the best PSO parameters to use in the optimizations.

Nomenclature

a, b = well-spacing parameters
c = compressibility, psi^{-1}
c_1, c_2 = weight of cognitive and social components in velocity equation
d = number of optimization variables
$\mathbf{D}_1$, $\mathbf{D}_2$ = diagonal matrices of random numbers between 0 and 1
$\mathbf{H}$ = axis shearing factor
I^{wp} = index of well pattern
k = index of iteration

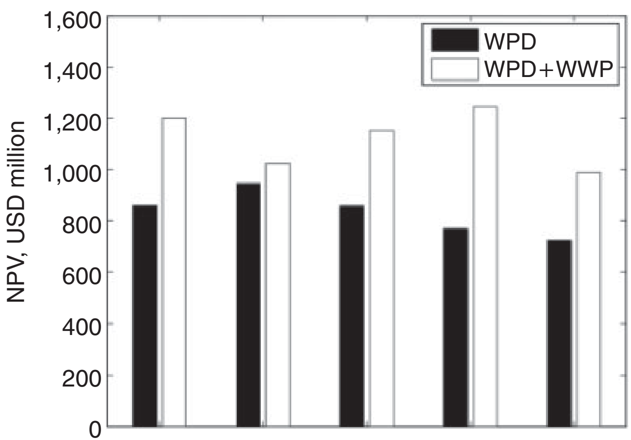


Fig. 20—NPV of the WPD and WWP optimizations for each realization (Example 3).

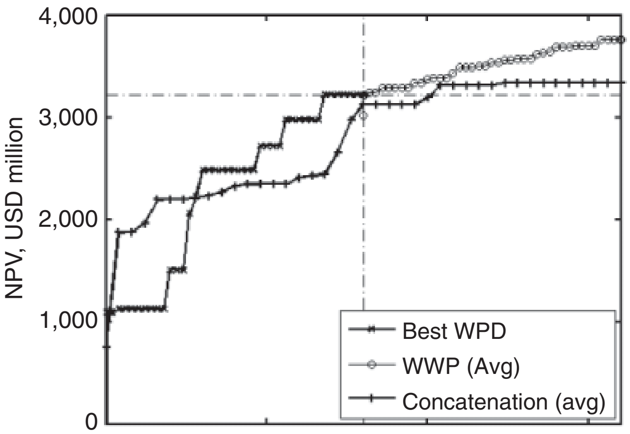


Fig. 21—The NPV of the best WPD solution and the average NPV of the five WWP solutions compared to well-by-well concentration (Example 4).

$\mathbf{M}$ = transformation matrix

$\mathbf{M}_\theta$, $\mathbf{M}_{\mathrm{sc}}$, $\mathbf{M}_{\mathrm{sh}}$ = rotation, scale, and shear transformation matrices, respectively

n = index of a well

N_p = number of different well patterns

N_{wp} = number of wells in a well pattern

$\mathcal{N}_o$ = number of pattern operators

$\mathcal{O}$ = pattern operator

$\mathcal{P}$ = pattern representation

S = axis scaling factor, operator sequence

S_{gi}, S_{oi}, S_{wi} = initial gas, oil, and water saturations, respectively

$\mathbf{v}$ = PSO particle velocity

$\mathbf{W}$ = matrix of well locations in a well pattern

$\mathbf{x}$ = PSO particle position

Δt = time increment

$\Delta\xi$, $\Delta\eta$ = spatial perturbations of a well location

θ = rotation angle

μ = viscosity, cp

ξ, η = areal location of a well

ξ^0, η^0 = center of a well pattern

ω = inertia weight

Subscripts

g, o, w = gas, oil, and water phases, respectively

i = index of PSO particle

j = index of pattern operator, index of optimization variable

Superscripts

c = cognitive

$n\mathrm{best}$ = neighborhood best

$p\mathrm{best}$ = previous best

ref = reference

s = social

wp = well pattern

Acknowledgments

We are grateful to the industrial affiliates of the Stanford University Advanced Wells and Smart Fields consortia for funding this research. We also thank the Stanford Center for Computational Earth and Environmental Science for providing the computational resources used in this work. We are also grateful to an anonymous reviewer for his/her useful comments on the original version of this paper.

References

3DSL User Manual v2.30. 2006. San Francisco, California: Streamsim Technologies.

Abdelhalim, M.B. and Habib, S.E.-D. 2009. Particle Swarm Optimization for HW/SW Partitioning. In *Particle Swarm Optimization*, ed. A. Lazinica, 49–76. InTech—Open Access Publisher, http://www.intechopen.com/articles/show/title/particle_swarm_optimization_for_hw_sw_partitioning.

Cao, H. 2002. Development of Techniques for General Purpose Simulators. PhD thesis, Stanford University, Stanford, California.

Clerc M. 2006a. *Particle Swarm Optimization*. London, UK: Wiley-ISTE.

Clerc, M. 2006b. Stagnation Analysis in Particle Swarm Optimisation or What Happens When Nothing Happens. Technical Report CSM-460, Department of Computer Science, University of Essex, Essex, UK (August 2006).

Craig, F.F. Jr. 1971. *The Reservoir Engineering Aspects of Waterflooding*. Monograph Series, SPE, Richardson, Texas 3: 29–77.

Eberhardt, R.C. and Kennedy, J. 1995. A New Optimizer Using Particle Swarm Theory. *Proc.,* Sixth International Symposium on Micro Machine and Human Science (MHS '95), Nagoya , Japan, 4–6 October, 39–43. doi: 10.1109/MHS.1995.494215.

Emerick, A.A., Silva, E., Messer, B., Almeida, L.F., Szwarcman, D., Pacheco, M.A.C., and Vellasco, M.M.B.R. 2009. Well Placement Optimization Using a Genetic Algorithm With Nonlinear Constraints. Paper SPE 118808 presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, 2–4 February. doi: 10.2118/118808-MS.

Engelbrecht, A.P. 2005. *Fundamentals of Computational Swarm Intelligence*. West Sussex, UK: Wiley.

Fernández Martínez, J.L. and García Gonzalo, E. 2008. The Generalized PSO: A New Door to PSO Evolution. *Journal of Artificial Evolution and Applications* (2008): 861275. doi: 10.1155/2008/861275.

Fernández Martínez, J.L. and García Gonzalo, E. 2009. The PSO Family: Deduction, Stochastic Analysis and Comparison. *Swarm Intelligence* **3** (4): 245–273. doi: 10.1007/s11721-009-0034-8.

Jiang, Y. 2008. Techniques for Modeling Complex Reservoirs and Advanced Wells. Ph.D dissertation, Stanford University, Stanford, California (June 2008).

Kennedy, J. and Eberhart, R. 1995. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks* **4** (December): 1942–1948. doi: 10.1109/ICNN.1995.488968.

Litvak, M. Gane, B., McMurray, L., and Skinner, R. 2007. Field Development Optimization in a Giant Oil Field in Azerbaijan and a Mature Oil Field in the North Sea. Paper OTC 18526 presented at the Offshore Technology Conference, Houston, 30 April–3 May. doi: 10.4043/18526-MS.

Litvak, M.L. and Angert, P.F. 2009. Field Development Optimization Applied to Giant Oil Fields. 2009. Paper SPE 118840 presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 2–4 February. doi: 10.2118/118840-MS.

Matott, L.S., Rabideau, A.J., and Craig, J.R. 2006. Pump-and-Treat Optimization Using Analytic Element Method Flow Models. *Advances in Water Resources* **29** (5): 760–775. doi: 10.1016/j.advwatres.2005.07.009.

Onwunalu, J.E. 2010. Optimization of Field Development Using Particle Swarm Optimization and New Well Pattern Descriptions. PhD thesis, Stanford University, Stanford, California.

Onwunalu, J.E. and Durlofsky, L.J. 2010. Application of a Particle Swarm Optimization Algorithm for Determining Optimum Well Location and Type. *Computational Geosciences* **14** (1):183–198. doi: 10.1007/s10596-009-9142-1.

Ozdogan, U., Sahni, A., Yeten, B., Guyaguler, B., and Chen, W.H. 2005. Efficient Assessment and Optimization of a Deepwater Asset Using Fixed Pattern Approach. Paper SPE 95792 presented at the SPE Annual Technical Conference and Exhibition, Dallas, 9–12 October. doi: 10.2118/95792-MS.

Pan, Y. and Horne, R.N. 1998. Improved Methods for Multivariate Optimization of Field Development Scheduling and Well Placement Design. Paper SPE 49055 presented at the SPE Annual Technical Conference and Exhibition, New Orleans, 27–30 September. doi: 10.2118/49055-MS.

Poli, R., Kennedy, J., and Blackwell, T. 2007. Particle Swarm Optimization. *Swarm Intelligence* **1** (1): 33–57. doi: 10.1007/s11721-007-0002-0.

Volz, R., Burn, K., Litvak, M., Thakur, S., and Skvortsov, S. 2008. Field Development Optimization of Siberian Giant Oil Field Under Uncertainties. Paper SPE 116831 presented at the SPE Russian Oil and Gas Technical Conference and Exhibition, Moscow, 28–30 October. doi: 10.2118/116831-MS.

Y. Shi and R. C. Eberhardt. 1998. A Modified Particle Swarm Optimizer. *Proc.,* IEEE International Conference on Computational Intelligence—Evolutionary Computation, Anchorage, 4–9 May, 69–73.

Jerome E. Onwunalu is a reservoir engineer at the Exploration, Production and Technology Group at BP America in Houston. Onwunalu holds a BEng degree from the University of Port Harcourt and MS and PhD degrees from Stanford University, all in petroleum engineering. He currently works in the Advanced Reservoir Performance Prediction team developing efficient algorithms and workflows in the area of field development planning and optimization. **Louis J. Durlofsky** is the Otto N. Miller professor and chairman in the Department of Energy Resources Engineering at Stanford University. He was previously affiliated with Chevron Energy Technology Company. Durlofsky holds a BS degree from The Pennsylvania State University and MS and PhD degrees from the Massachusetts Institute of Technology, all in chemical engineering. He codirects the Stanford University Industrial Affiliate Program on Reservoir Simulation and the Stanford Smart Fields Consortium.