

Introduction

Maximizing production from an oil field is a crucial task, given the enormous financial investment at stake in any large-scale field development. Careful planning with respect to the placement of new wells and control of injection and production rates at existing wells is essential, as these decisions can have a significant impact on production. Placing wells poorly – in regions of low permeability, for instance – may make it difficult to achieve good flow rates, while suboptimal control strategies may result in premature waterflooding at production wells. The vast number of potential development scenarios drives the need for efficient, computerized optimization approaches to assist in making these decisions.

Finding optimal well locations and determining optimal well control are often treated as separate problems (Ciaurri et al. (2011)). Well placement problems involve optimizing over parameters corresponding to the positions and orientations of injection and production wells. We limit ourselves in this paper to considering vertical wells, which are parameterized simply by the well's (x, y) co-ordinates. A simple control scheme is typically assumed in well placement problems; for instance, injection wells can be held at a fixed bottom hole pressure (BHP), while producers are held at a lower BHP in order to generate flow. Well control problems, on the other hand, focus on managing the injection and production rates at wells that are already in place. The optimization variables in this case are usually either the BHP or the flow rate for each well, which can be changed at specified time intervals.

The objective function that one wishes to maximize in both of these problems is typically either the total amount of oil extracted, or the net present value (NPV) of the extracted oil. The NPV function is related to the total amount of oil extracted, but emphasizes producing more oil early in the reservoir's lifetime (due to the time value of money) and also usually incorporates the costs of water injection and disposal of any water produced. Evaluating this objective function requires running a reservoir simulator, and is therefore a computationally expensive operation. The behaviour of the objective function is notably different in well control and well placement problems. Specifically, the function tends to vary smoothly as control parameters are perturbed, while in well placement problems, the function is usually nonsmooth, due to heterogeneous properties of the reservoir such as permeability. Thus, different optimization approaches have typically been used for addressing these two problems. Optimization studies on well placement have often focused on global algorithms with some stochastic element in order to avoid local optima, while well control problems have tended to make use of deterministic algorithms based on local search techniques (Ciaurri et al. (2011)).

A unified approach to optimizing well placement and well control at the same time has the potential to provide benefits over the treatment of these problems separately. In particular, the best well configuration when producers are held at some fixed BHP is not necessarily the same as the best configuration if the control can vary with time (Zandvliet et al. (2008)). Additionally, determining the optimal placement of new wells may also require adjusting the control parameters at wells already in place. The problem of simultaneous optimization of well placement and control has been largely unexplored in academic literature. Here, we investigate approaches to addressing this problem using a two optimization algorithms: particle swarm optimization (PSO) and generalized pattern search (GPS). We find that hybridizing these two algorithms and applying them simultaneously provides advantages in some simple experiments; in more complicated cases, applying them in sequential steps may work best.

Existing research

Well placement studies have tended to use stochastic optimization approaches aimed at exploring the solution space globally. Genetic algorithms (GAs) have received the widest use (Bittencourt and Horne (1997); Yeten et al. (2003); Güyagüler and Horne (2004); Artus et al. (2006); Ozdogan et al. (2005); Nogueira and Schiozer (2009); Emerick et al. (2009); Bukhamsin et al. (2010)). Other optimization algorithms that have been applied to the problem include simultaneous perturbation stochastic approxi-

mation (SPSA) (Bangerth et al. (2006)), covariance matrix adaptation (Bouzarkouna et al. (2010)), and particle swarm optimization (PSO) (Onwunalu and Durlofsky (2010, 2011)). These methods were found to outperform GAs in most cases, in that they found better solutions, or else required fewer function evaluations to match the performance of GAs. In addition to determining suitable algorithms, well placement papers have addressed other issues such as parametrization and optimal placement of non-conventional wells (Yeten et al. (2003); Bukhamsin et al. (2010)), consideration of geological uncertainty when determining optimal positions (Güyagüler and Horne (2004); Artus et al. (2006)), placement of well patterns rather than individual wells (Ozdogan et al. (2005); Onwunalu and Durlofsky (2011)), and inclusion of nonlinear constraints as part of the optimization (Zandvliet et al. (2008); Emerick et al. (2009)).

A popular optimization algorithm for well control problems, on the other hand, has been the adjoint method (Brouwer and Jansen (2002); Sarma et al. (2006); Zandvliet et al. (2007); Jansen et al. (2009); van Essen et al. (2011)). The adjoint method determines descent directions on the objective function surface by approximating the gradient of the function. This algorithm is well-suited to the optimal control problem due to the smoothly varying nature of the objective function. Formulating the gradient approximations requires in-depth knowledge of the workings of the reservoir simulator, however, and may be challenging as a result. This issue can be avoided by using “black box” optimization algorithms, which deal only with inputs and outputs to the simulator. Examples of black-box algorithms that have been applied to the well control problem include stochastic methods like SPSA (Wang et al. (2009)) and GAs (Yang et al. (2003)), as well as deterministic methods such as generalized pattern search (GPS) and Hooke-Jeeves directed search (Ciaurri et al. (2010, 2011)).

Several well control studies have developed the concept of closed-loop reservoir management (Jansen et al. (2009); Wang et al. (2009); Peters et al. (2009)), where the geology of the reservoir must be estimated based on measured observational data. Thus, the optimization approach must include a history-matching component, which builds an approximate model of the reservoir based on measurements taken during production. The updated model is then used to determine the optimal control scheme for the next time period. We assume in this study that an accurate model of the reservoir is available, and thus do not consider the problem of history matching.

The need to include nonlinear constraints during optimization is a key issue that arises in well control problems (Sarma et al. (2006); Zandvliet et al. (2007); Ciaurri et al. (2010)). When wells are controlled by BHP, for example, the flow rate at each well has a complicated, nonlinear dependence on not only the well’s BHP, but also on numerous other factors, such as the reservoir’s permeability field and the well’s proximity to other wells. In addition to straightforward upper and lower bounds on BHP, there may also be maximum or minimum permissible flow rates prescribed at each well, which are not directly controlled. It is not generally possible to determine whether these constraints are satisfied without performing a reservoir simulation. Any optimization approach must therefore include some mechanism for handling solutions that improve the value of the objective function, but violate these constraints. Constraints of this type have major implications on the optimal control. If no constraints are present then the optimal control is guaranteed to be of the “bang-bang” type, meaning that the BHP should be held only at either its maximum or minimum permissible value (Sudaryanto and Yortsos (2001); Zandvliet et al. (2007); Wang et al. (2009)). If the problem is constrained, then the optimal control may include BHP values from anywhere within the permissible range.

Optimization approach

Combining global and local optimization techniques should be advantageous when addressing well control and well placement in a unified way, given the differing nature of these two problems. We use PSO as a global optimizer in this study, and GPS for the local search. Our choice of these algorithms is motivated by the fact that both have performed well in previous production optimization studies (Onwunalu and Durlofsky (2010); Ciaurri et al. (2010)); both are black-box methods that do not require in-depth

knowledge of the simulator; and both are easily parallelized to help mitigate the expensive cost of function evaluations. We now give an overview of PSO and GPS, as well as of the specific optimization approaches used in this paper.

Particle swarm optimization

Particle swarm optimization (Kennedy and Eberhart (1995); Clerc (2006)) is an optimization algorithm based on modeling the behaviour of a herd of animals acting collectively. PSO utilizes a number of *particles* (typically 20 to 40) to explore solution space in a semi-random way. The position of particle i at iteration k , denoted $\mathbf{x}_i^{(k)}$, is a vector of size N , where N is the number of variables in the optimization problem. Every position is associated with the corresponding objective function value, and every particle remembers the best position it has found so far. Particles in the swarm also communicate with one another to share the best positions that have been found by the swarm on the whole. Given $\mathbf{x}_i^{(k)}$, the position of the particle at iteration $k + 1$ is:

$$\mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)} + \mathbf{v}_i^{(k+1)}, \quad (1)$$

where the particle's velocity vector $\mathbf{v}_i^{(k+1)}$ is given by

$$\mathbf{v}_i^{(k+1)} = \iota \mathbf{v}_i^{(k)} + \mu \mathbf{r}_1^{(k)} \otimes (\mathbf{p}_i^{(k)} - \mathbf{x}_i^{(k)}) + \nu \mathbf{r}_2^{(k)} \otimes (\mathbf{g}_i^{(k)} - \mathbf{x}_i^{(k)}). \quad (2)$$

The velocity is a combination of three terms. The first term models the tendency of the particle to continue traveling in the direction given by its current velocity. The second term represents the tendency of the particle to move toward the best position it has found so far, denoted by $\mathbf{p}_i^{(k)}$. Finally, the third term represents the tendency of the particle to move toward the best position found by any other particle with which it communicates, denoted by $\mathbf{g}_i^{(k)}$. The constants ι , μ and ν are parameters whose values are chosen to weight these three terms appropriately. To inject randomness into the particle movement, the N -vectors $\mathbf{r}_1^{(k)}$ and $\mathbf{r}_2^{(k)}$ are generated from the uniform distribution on $(0, 1)$ at every iteration, then multiplied componentwise with the terms in brackets by the \otimes operator. The PSO iteration continues until some convergence criterion is met; for example, until the velocities of the particles have become sufficiently small, until the particles are sufficiently close to one another, or simply until some maximum number of iterations have been performed.

If every particle communicates with every other particle in the swarm, then one can replace the term $\mathbf{g}_i^{(k)}$ in (2) with a single value $\mathbf{g}^{(k)}$, representing the overall global best solution found so far. Under this *global* neighbourhood topology, PSO may quickly converge to a minimum before the solution space is fully explored. Thus, it is usually recommended that each particle communicate only with 2 to 4 other particles at any one time (Clerc (2006)). At every iteration of the algorithm, the neighbourhood of particles with which each particle communicates can be chosen randomly. This *random* neighbourhood topology was used for this study, as well as a swarm size of 20 particles, and parameter values of $\iota = 0.721$, $\mu = \nu = 1.193$. These parameter values have been found to provide good convergence results in many numerical experiments (Clerc (2006)).

Generalized Pattern Search

Generalized Pattern Search (Lewis and Torczon (1999); Audet and Dennis (2004)) is an optimization algorithm that begins from a single *incumbent point* and consists of a series of *search* and *poll* steps. At every iteration k , a discrete mesh, centred at the current incumbent $\mathbf{x}^{(k)}$, is defined by:

$$M^{(k)} = \left\{ \mathbf{x}^{(k)} + \Delta^{(k)} D \mathbf{z} : \mathbf{z} \in \mathbb{N}^{n_D} \right\}, \quad (3)$$

where $\Delta^{(k)}$ is the resolution of the mesh at iteration k , D is a matrix whose columns form the *search directions*, \mathbb{N} is the set of natural numbers, and n_D is the number of search directions. The search

directions must form a *positive spanning set* in solution space; i.e., one must be able to specify any point in solution space by adding together only positive scalar multiples of these directions. Two common choices of search directions are:

$$\mathcal{D} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N, -\mathbf{e}_1, -\mathbf{e}_2, \dots, -\mathbf{e}_N\}, \text{ or} \quad (4)$$

$$\mathcal{D} = \left\{ \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N, -\sum_{n=1}^N \mathbf{e}_n \right\}, \quad (5)$$

where the \mathbf{e}_n are the canonical basis vectors $(1, 0, 0, \dots, 0)^T$, $(0, 1, 0, \dots, 0)^T$, etc. Here \mathcal{D} refers to the set of search directions, which form the columns of the matrix D .

The search step consists of selecting a finite number of points on $M^{(k)}$ and evaluating the objective function at each one. If any of those points improves the objective function value, the point with the best value becomes the new incumbent. The search step can employ any strategy in selecting points, and may even be omitted, if desired. If none of the points selected in the search step are better than the incumbent, then the algorithm proceeds to the poll step. The poll step consists of evaluating the objective function at all the points that are immediate neighbours of the incumbent point on the mesh $M^{(k)}$. These points are given by:

$$\{\mathbf{y}_j^{(k)}\} = \left\{ \mathbf{x}^{(k)} + \Delta^{(k)} \mathbf{d}_j \mid \forall \mathbf{d}_j \in \mathcal{D} \right\}. \quad (6)$$

If the poll step finds one or more points with a better objective function value than the incumbent, then the point with the best value becomes the new incumbent. Optionally, $\Delta^{(k)}$ may be increased for the next iteration. If the poll step is unsuccessful, then $\Delta^{(k)}$ is reduced and another iteration begins, using the same incumbent point as before. The algorithm is considered to have converged once $\Delta^{(k)}$ is reduced beyond a specified threshold, which indicates that the current point is at least close to a local optimum.

Bound and general constraints

Broadly speaking, there can exist two types of constraints on the optimization vector \mathbf{x} : *bound* and *general* constraints. Bound constraints are simple componentwise inequality constraints of the form

$$\mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u, \quad (7)$$

where \mathbf{x}^l and \mathbf{x}^u are the lower and upper bounds on \mathbf{x} , respectively. In the context of a reservoir optimization problem, these could be the minimum and maximum grid indices (for well placement) or upper and lower limits on the control parameters.

Both PSO and GPS can easily incorporate bound constraints. In PSO, any particles that travel outside of the bounds are projected back onto the boundary of search space. For instance, if component d of particle i 's position exceeds the maximum value \mathbf{x}_d^u after being updated, then the particle's position and velocity are modified as follows:

$$\begin{aligned} \mathbf{x}_{i,d} &= \mathbf{x}_d^u \\ \mathbf{v}_{i,d} &= 0 \end{aligned} \quad (8)$$

The velocity component is set to zero to ensure that the particle does not continue to travel in the direction that led it out of bounds. Bound constraints are treated similarly in GPS; namely, points which lie outside of search space are projected back onto the boundary during the poll step (Lewis and Torczon (1999)).

General constraints refer to any constraints on the input parameters other than simple bound constraints. Input that violates general constraints (which we refer to as *infeasible* input) can sometimes be identified prior to evaluating the objective function; for instance, if the input specifies placing two wells at the same location. Other constraints, such as an upper limit on the flow rate for wells controlled by BHP, require

running the reservoir simulator to determine if they are satisfied. A simple mechanism for PSO to handle general constraints is to allow particles to move to infeasible positions, but not store these positions in the particle's memory (Hu and Eberhart (2002)). Thus, particles can explore search space freely, but are only attracted to positions that are feasible, in addition to providing good objective function values. This strategy requires that every particle be initialized to a feasible position, so that the particle always has at least one feasible position stored in its history. To handle general constraints in GPS, one can simply ignore any infeasible points during polling, and thus only accept feasible points which also reduce the objective function value. This approach is not ideal for general-purpose optimization, as it may prevent the algorithm from traveling through the infeasible region to find the true optimum; alternative approaches such as *filtering* are recommended instead (Audet and Dennis (2004); Ciaurri et al. (2010)). We found that this first approach was sufficient for this study, however, possibly because GPS was used in conjunction with PSO, rather than as a stand-alone optimizer.

Hybrid algorithm

An optimization algorithm that hybridizes PSO and GPS has previously been proposed in (Vaz and Vicente (2007, 2009)). This algorithm, denoted PSwarm, is essentially a GPS algorithm that uses PSO as the *search* step. Thus, the algorithm behaves exactly like PSO for as long as the search step continues to find points that improve the objective function value. When this step fails to improve the solution, polling takes place around the current best position found. If the poll step finds a better solution, the current best position is updated and a new iteration of PSO begins; otherwise, the polling stencil size is reduced, as described in the section on GPS. The algorithm proceeds until the convergence criteria for both PSO and GPS are met; i.e., the velocity of the particles is sufficiently small, and the polling stencil size is reduced beyond a specified threshold.

In this paper we have made the following modifications to PSwarm in order to adapt it to the simultaneous well placement and control problem:

1. We have extended the PSO and GPS components of the algorithm to handle general constraints, as described in the previous section. The PSwarm algorithm as described in (Vaz and Vicente (2009)) handles linear constraints, but not general constraints of the types seen in this problem.
2. PSwarm uses the global network topology as the communication model during the PSO step. We have replaced this model with the random variable neighbourhood topology, as discussed in the section on PSO. Each particle's communication neighbourhood consists of itself and two other particles, which are selected randomly at each iteration.
3. The PSwarm algorithm, as originally proposed, performs a poll step every time the PSO (search) step fails to improve the objective function value. We have relaxed this condition so that the PSO step was allowed to fail up to five times consecutively before GPS was applied. This choice was made in light of the fact that evaluating our objective function requires running the reservoir simulator, and is thus more computationally expensive than in most optimization problems. The cost of polling is therefore greater. The failure tolerance of five iterations was chosen after some experimentation showed that it provided comparable results to when polling was performed more frequently, despite requiring far fewer function evaluations on average.
4. We have chosen specially selected direction vectors \mathcal{D} to use during the poll step.

Regarding the fourth item, a common choice for GPS search directions are the canonical vectors given in (4). In the context of the well placement and control problem, each of these directions corresponds to incrementally perturbing either the x or y component of one well's position, or a single BHP value at one well for one time interval. We instead chose a specialized set of search directions that are likely to

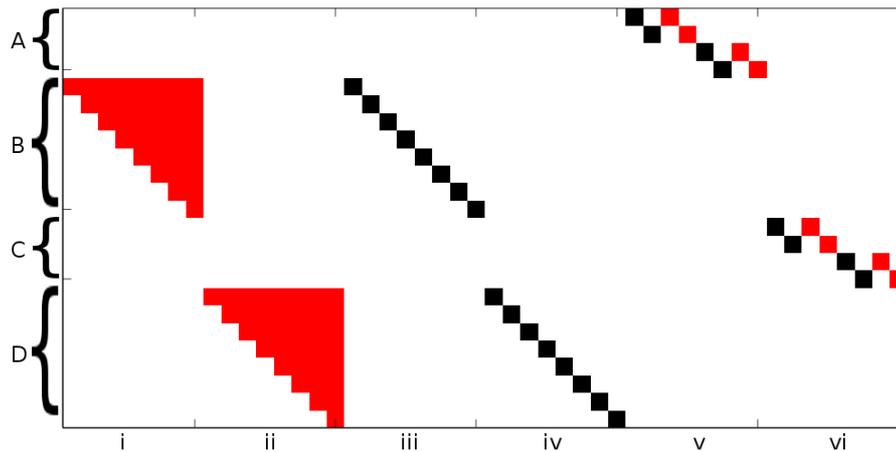


Figure 1 Specialized search directions used during the GPS algorithm for the mock problem. A red entry corresponds to a value of -1 , and a black entry to $+1$.

result in a greater improvement to the objective function (the Net Present Value, in this case) in a single step. Specifically, the NPV is more likely to be improved by lowering the BHP at production wells, since doing so generates a higher flow rate. The opposite is true for injection wells. Furthermore, we gain more by increasing production in early years, due to the discounting rate applied in the NPV calculation. The search directions were chosen with these two facts in mind.

We illustrate this second set of search directions in Figure 1 for a mock problem, involving the placement and control for a group of four vertical wells, consisting of two injectors and two producers. Each well's position is determined by its (x,y) co-ordinates, and the BHP at each producer can be changed every year over an 8-year production period. The BHP at injectors is held fixed. Thus, there are 24 variables; the (x,y) co-ordinates of each of the four wells, and the 8 BHP values for each of the two producers. The four components along the y -axis in Figure 1 (labeled A) correspond to the x and y co-ordinates of the first injector-producer pair, and the next 8 components B to the BHP specified at the first producer in each year of production. Labels C and D correspond to the equivalent components for the second injector-producer pair.

Each column of the matrix represents one search direction. The first 8 directions (labeled i) correspond to lowering the BHP at the first producer, with the first direction corresponding to lowering BHP in the first year, the second direction to lowering it in years 1 and 2, until the 8th direction lowers the BHP at the well for all 8 years. The directions labeled ii do the same for the second producer. Directions iii and iv correspond to raising the BHP at the first and second producer, respectively. These ensure that directions $i - iv$, taken together, form a positive spanning set over the control parameters. Finally, directions v and vi alter the x and y co-ordinates of the first and second injector-producer pairs, respectively. This last group of directions are scaled independently of the control variables during the polling step, so that we only ever alter the x or y co-ordinate of a well by one grid space during polling. The idea is that the optimization of the well positions is primarily achieved by the PSO step. Well positions should only need to be perturbed slightly during the poll step, which is aimed mainly at optimizing the controls.

Experiments

We now describe several experiments that were used to test the performance of the different optimization approaches. All experiments were performed using the Matlab Reservoir Simulation Toolbox (MRST) (Lie et al. (2011); SINTEF Applied Mathematics (2011)) as the reservoir simulator. MRST

Table 1 Economic parameters used in all experiments.

Parameter	Value
c_o	\$80/bbl
$c_{w,disp}$	\$12/bbl
$c_{w,inj}$	\$8/bbl
r	10%

is an open-source simulator implemented in Matlab, which includes routines for processing and visualizing unstructured grids, as well as several solvers for single and two-phase flow. The flow and transport equations are solved in alternating steps in order to determine the phase pressures, flow rates and saturations at every time point. Modeling of simple vertical and horizontal wells is provided using the Peaceman model (Peaceman (1978)).

The objective function we used in these experiments was the NPV over the entire production period $[0, T]$. The NPV was computed as in (Bangerth et al. (2006)):

$$NPV(\mathbf{x}) = \int_0^T \left\{ \sum_{n \in prod} [c_o q_{n,o}^-(t) - c_{w,disp} q_{n,w}^-(t)] - \sum_{n \in inj} c_{w,inj} q_{n,w}^+(t) \right\} (1+r)^{-t} dt, \quad (9)$$

The parameters c_o , $c_{w,disp}$ and $c_{w,inj}$ represent the price per barrel of produced oil, disposal cost per barrel of produced water, and cost per barrel of injected water, respectively. The functions $q_{n,o}^-(t)$ and $q_{n,w}^-(t)$ are the production rates (barrels/day) of oil and water, respectively, at well n , while $q_{n,w}^+(t)$ is the water injection rate at well n . These rates are implicitly functions of the optimization vector \mathbf{x} , since they depend on the prescribed bottom hole pressures. The yearly interest rate is specified by r . We used the parameter values provided in Table 1 for all experiments. This choice of values meant that production became unprofitable once the water cut at a well reached roughly 78%. This threshold value is often as high as 90 or 95% in practice; a lower value was chosen to ensure that shutting in a well was the optimal choice in some experiments.

Experiment 1

The first experiment used a simple 2D reservoir model, consisting of 50×60 grid cells measuring 25 metres per side (total field size: $1250 \times 1500 \times 25$ m, or $4100 \times 4920 \times 82$ ft). The small size of this reservoir block allowed us to run many iterations of each algorithm and study the convergence behaviour. The permeability field of the reservoir is shown in Figure 2. This field contained several regions of high permeability where we would expect optimally configured wells to be placed. A uniform porosity value of 20% was assumed throughout the reservoir. The initial saturation of the reservoir was assumed to be 100% oil.

We considered four problems using this reservoir model. In the first two problems, we placed a single vertical injector/producer pair. The injector was held at a fixed BHP of 350 bars (5076 psi), while the producer BHP was permitted to lie between 175 and 350 bars (2538–5076 psi), and could be changed every year over a twelve-year production period. Thus, there were 16 variables in this problem; 4 positional variables and 12 control variables. We also considered two situations: one where there were no general constraints on the optimization (denoted Problem 1A), and one where a maximum flow rate of $600 \text{ m}^3/\text{day}$ (3774 bpd) was imposed on both wells (denoted 1B). The final two problems (1C and 1D) involved placing two injectors and two producers in the same reservoir, under the same conditions. This doubled the size of the problem to 32 variables. Again, we considered both the unconstrained problem (Problem 1C) and the problem where the maximum flow rate was constrained at $600 \text{ m}^3/\text{day}$ (Problem 1D). The experimental parameters of these four problems are summarized in Figure 3 (top image).

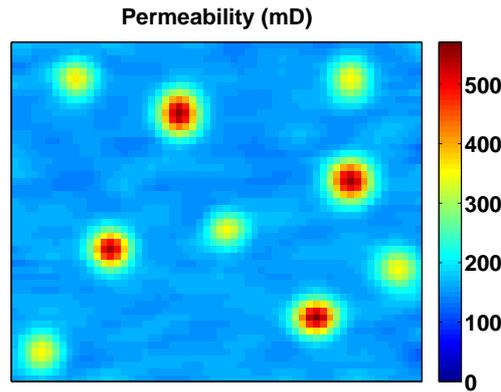


Figure 2 Permeability field of 2D reservoir used in first experiment.

We applied three different optimization approaches to these problems. The fact that every optimization approach that we considered included a stochastic component necessitated performing multiple runs of each approach, in order to assess the average performance. Each approach was, therefore, applied 20 times to the appropriate problems. Figure 3 (bottom image) summarizes which optimization approaches (described below) were applied to each of the four problems.

The first approach was simply to apply PSO. PSO was run for a maximum of 200 iterations for Problems 1A and 1B, and up to 300 iterations for Problems 1C and 1D, with the algorithm terminated early if the average velocity of the swarm decreased beyond a certain threshold. We subsequently applied GPS to the best solution found in each run of PSO. This step consisted only of polling, starting from a stencil size ($\Delta^{(0)}$) of 32 bars for the control variables. The stencil size was reduced by a factor of 2 any time that polling did not find a better incumbent point, and the algorithm was terminated once $\Delta^{(k)}$ was equal to 1 bar. The application of GPS was not considered to be part of the optimization approach, but rather as a test to see how close the solutions found by PSO were to being locally optimal.

The second approach was to apply the hybrid algorithm (modified PSwarm) described in the previous section. This algorithm was run for a maximum of 200 iterations for all four test cases, but could be terminated early if the convergence criteria for both PSO and GPS were satisfied. We then applied GPS to the best solution found by each run in order to test its optimality, as we did with the solutions found by PSO. Solutions found by the hybrid algorithm could only be improved by this step if the maximum number of iterations had been reached, since the convergence criterion for GPS had to have been satisfied already for the algorithm to terminate early.

The third approach that we considered was to decouple the placement and control components of the problem. The first step of this approach consisted of treating the problem strictly as a well placement problem, by assuming that the producers were held at some fixed BHP throughout the 12-year production period. We used up to 200 iterations of PSO to determine the optimal well positions under these assumptions. Once optimal positions had been found, we allowed the controls to vary year-by-year and optimized the control using GPS. The positions could also be incrementally adjusted in this second step. This second step ensured that the solutions found by the decoupled approach were guaranteed to be locally optimal. The advantage of the decoupled approach is that it splits the problem into two smaller problems which are easier to solve than the full problem. A potential disadvantage is that we may find suboptimal solutions by not optimizing over all variables at the same time. We applied this approach only to Problems 1B and 1D, since the optimal control was expected to be “bang-bang” in Problems 1A and 1C, and could thus be found relatively easily. When initially placing the wells, the producers were held at a BHP of 200 bars, which was slightly higher than the minimum value of 175 bars, in order to

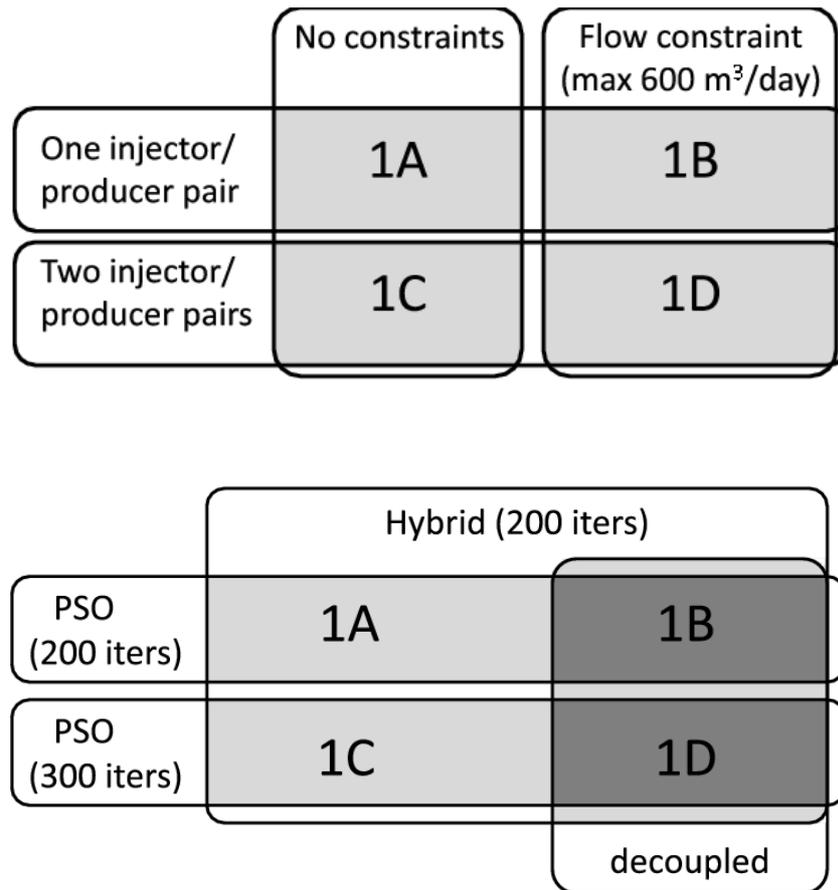


Figure 3 Top image: Experimental parameters that were changed for the four problems in Experiment 1. Bottom image: Optimization approaches that were applied to each of the four problems in Experiment 1.

allow the control to be iteratively improved by GPS in the second step.

Experiment 2

The second experiment used a reservoir model provided by the Norwegian University of Science and Technology (NTNU) as part of the Norne benchmark case (NTNU IO Centre (2012)). The full model of the Norne field is a $46 \times 112 \times 22$ grid consisting of 44,927 active cells. The reservoir model is subdivided into four different formations from top to base, denoted *Garn*, *Ile*, *Tofte* and *Tilje*. In order to reduce simulation time, we extracted the seven layers corresponding to the *Ile* formation to provide a smaller reservoir model, consisting of 15,004 active cells. The porosity of the reservoir ranged between 25–30% and the permeability from 20 to 2500 mD. The reservoir geometry is shown in Figure 4 (left image). The initial saturation was assumed to be 100% oil, as in Experiment 1.

The reservoir’s irregular shape meant that wells whose positional co-ordinates fell within the bounds prescribed by the grid might not correspond to valid locations in the reservoir. Thus, any positional co-ordinates in the (x,y) plane which did not correspond to a valid reservoir location were projected onto the nearest active cell during the optimization. This process is illustrated in Figure 4 (right image). Black cells indicate grid locations which pass through at least one active cell in the z -direction. The three red \times symbols indicate positions that are invalid, which were projected onto the nearest valid location (indicated by the green \times symbols).

Experiment 2 consisted of placing seven wells (three injectors, four producers) in this field, and opti-

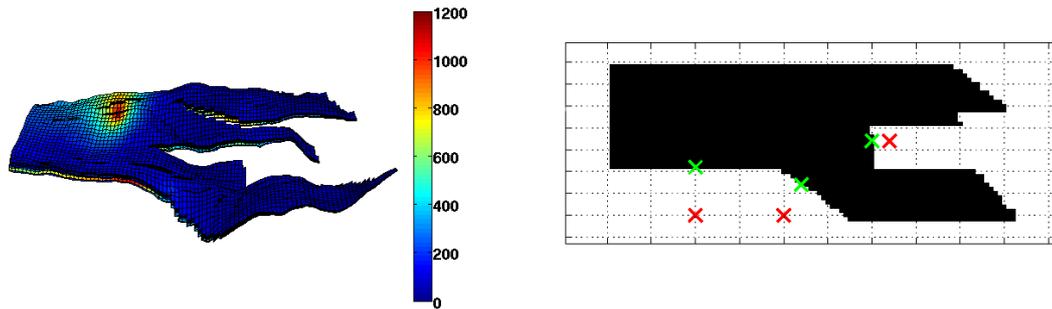


Figure 4 Left side: reservoir geometry of the Norne field. Permeability in the $x - y$ directions (in mD) is shown. Right side: Projection of invalid vertical well locations in the (x, y) plane onto the nearest valid co-ordinates.

mizing production over a 16-year time period. As in Experiment 1, the BHPs at injection wells were held fixed, this time at 450 bars (6527 psi). The BHPs at production wells could take any value between 150 and 450 bars (2176–6257 psi), and were allowed to be changed every year. All wells were assumed to be vertical and perforated in all seven layers of the field. Thus, the optimization variables consisted of 14 positional parameters (the (x, y) co-ordinates of every well) and 16 control parameters for each of the four producers (64 total), for a total of 78 variables. We again considered both the constrained and unconstrained cases; in Problem 2A, there were no constraints on production, while in Problem 2B, there was a maximum flow constraint of 3,500 m³/day (22,014 bpd) at each injector and 2,500 m³/day (15,725 bpd) at each producer.

The same optimization methodology was applied as in Experiment 1. Five trials of each experiment were run, with each trial consisting of up to 200 iterations of PSO or 150 iterations of the hybrid algorithm. Pattern search to the best solution found in each trial to assess its optimality. For the decoupled approach, up to 150 iterations of PSO were performed during the well placement step, assuming that producers were held at a fixed BHP of 175 bars.

Results

The results of both experiments are shown in Table 2. The central part of the table shows the average, best and worst NPV values over the multiple runs of each approach that were performed for each approach (twenty runs for Experiment 1, and five runs for Experiment 2). The rightmost section shows the average NPV after the GPS algorithm was applied to each solution found by PSO and the hybrid algorithm, as well as the percentage improvement ($\Delta\%$) compared to the original average. These values indicate how close to being locally optimal the solutions found by either algorithm were. Plots of the convergence of the respective algorithms versus the number of function evaluations (fevals) for both experiments are shown in Figure 5.

Discussion

Experiment 1

Table 2 shows that overall, there was little difference in the final NPV values obtained by the hybrid algorithm versus PSO. The average, best and worst solutions found by either algorithm were generally within 5% of one other. The lone exception was Problem 1D, where the worst solution found by PSO had roughly a 10% lower NPV than the worst solution found by the hybrid algorithm. The results in the rightmost column indicate that the solutions found by PSO were less likely to be locally optimal than those found the hybrid algorithm (particularly in Problems 1B and 1D), since they were more often

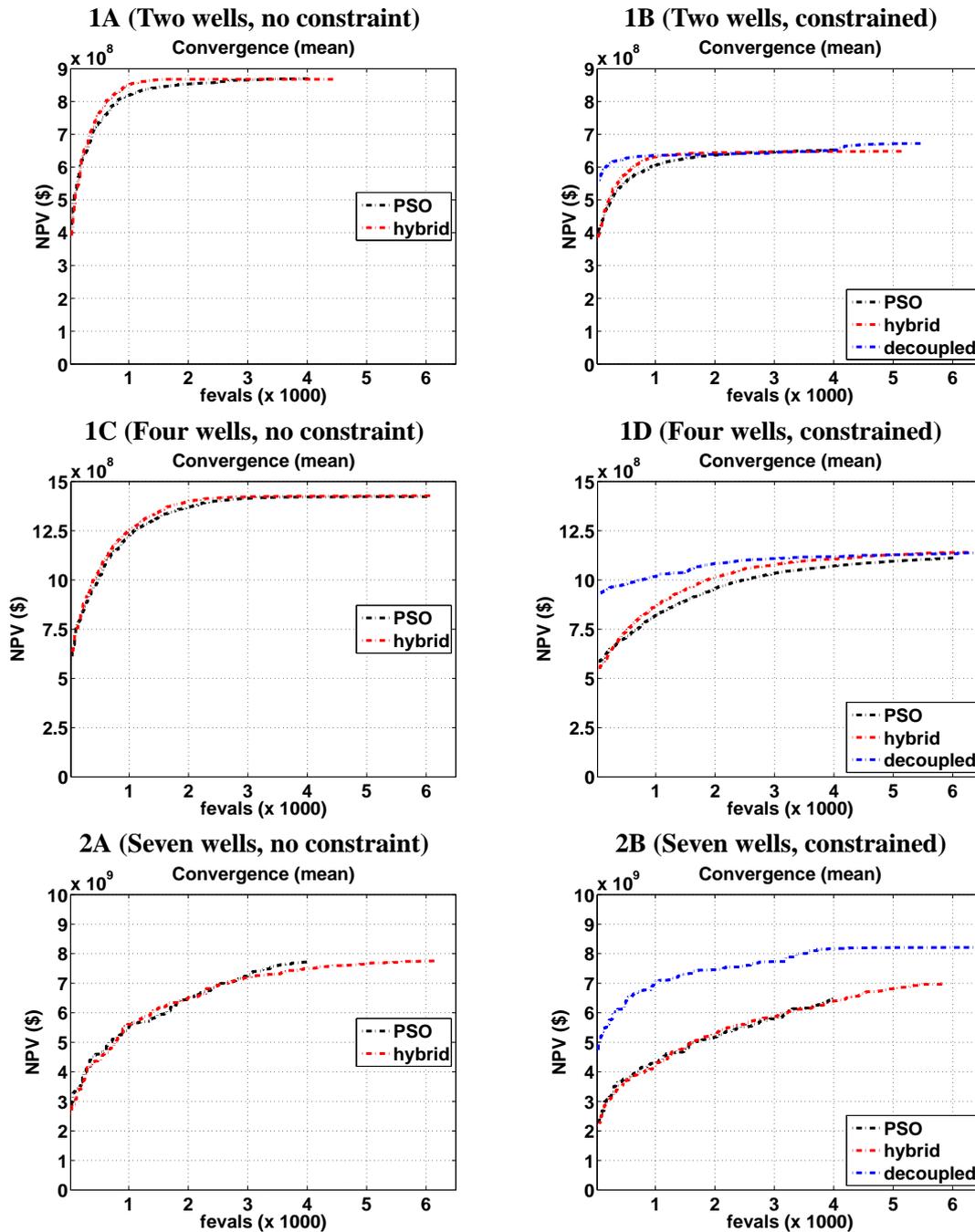


Figure 5 Convergence plots for the four problems of Experiment 1 and the two problems considered in Experiment 2, showing best NPV found as a function of the number of reservoir simulations (fevals). Convergence of PSO shown in black, hybrid algorithm in red, decoupled approach (where applicable) in blue. The vertical axis scale is the same across each row of plots.

Table 2 Results of first and second experiments.

Problem	Algorithm	First run			After GPS	
		Avg. (\$ $\times 10^8$)	Best (\$ $\times 10^8$)	Worst (\$ $\times 10^8$)	Avg. (\$ $\times 10^8$)	$\Delta\%$
1A	PSO	8.70	8.79	8.60	8.71	0.25
	hybrid	8.68	8.79	8.47	8.68	0.00
1B	PSO	6.52	6.67	6.36	6.58	1.20
	hybrid	6.48	6.59	6.40	6.49	0.20
	decoupled	6.72	6.92	6.37	—	—
1C	PSO	14.2	14.6	13.0	14.3	0.08
	hybrid	14.3	14.6	12.4	14.3	0.08
1D	PSO	11.1	12.2	9.61	11.4	2.64
	hybrid	11.4	12.2	10.6	11.5	0.75
	decoupled	11.3	12.1	10.3	—	—
2A	PSO	7.72	8.19	7.23	8.74	13.1
	hybrid	7.75	8.28	7.11	8.34	7.6
2B	PSO	6.47	7.05	6.09	7.63	18.0
	hybrid	6.99	7.89	5.81	7.61	8.9
	decoupled	8.21	8.73	7.72	—	—

improved by the subsequent application of GPS. This result was expected since the hybrid algorithm included a polling step to provide local optimization.

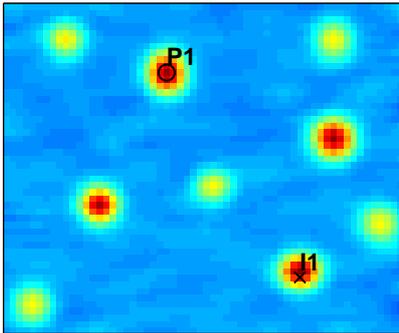
The main difference in the performance of the two algorithms is apparent from plotting the convergence of each algorithm against the number of function evaluations (fevals), as shown in Figure 5. Note that for PSO, the cost of each iteration was fixed at 20 fevals (the number of particles), while for the hybrid algorithm, it varied depending on whether or not the poll step was performed in that iteration. The convergence plots for Problems 1A and 1B show that the hybrid algorithm had typically found an optimal solution after roughly 1500 fevals, as indicated by the plateau in the convergence plot. PSO required 2500 to 3000 fevals to attain the same quality of solution. The gap in performance was even more pronounced for Problem 1D, where the performance of PSO lagged behind the hybrid algorithm's even after 6000 fevals. Problem 1C was the only case out of the four where the performance of the two algorithms was essentially the same.

The decoupled approach was applied only to Problems 1B and 1D, and was effective in both cases. This approach produced better solutions on average than either the hybrid or PSO algorithms for Problem 1B, while its performance for Problem 1D was comparable (see Table 2). The convergence of the decoupled approach for these two experiments are shown in Figure 5. We note that the decoupled approach began from a better initial solution than the other two methods, because the BHPs at producers were initially held near the low end of the range at 200 bars, as opposed to being randomly initialized between the minimum and maximum values of 175 and 350 bars. The hybrid algorithm was eventually able to “catch up” with the decoupled approach in both problems, despite this initial disadvantage; however, in Problem 1B, the decoupled approach eventually improved the solution further during the control optimization step.

The best well positions found over all optimization runs for each of the four problems are shown in Figure 6. Production wells in the two unconstrained problems (1A and 1C) were always placed in the high-permeability regions in order to generate the highest possible flow rate. The injection well

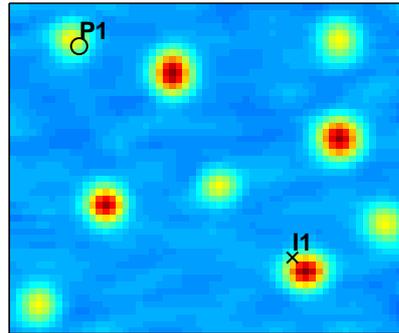
1A (Two wells, no constraint)

$$\text{NPV} = \$8.79 \times 10^8$$



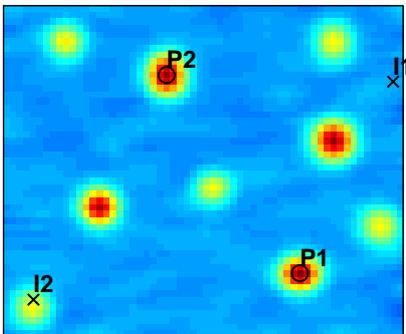
1B (Two wells, constrained)

$$\text{NPV} = \$6.92 \times 10^8$$



1C (Four wells, no constraint)

$$\text{NPV} = \$14.6 \times 10^8$$



1D (Four wells, constrained)

$$\text{NPV} = \$12.5 \times 10^8$$

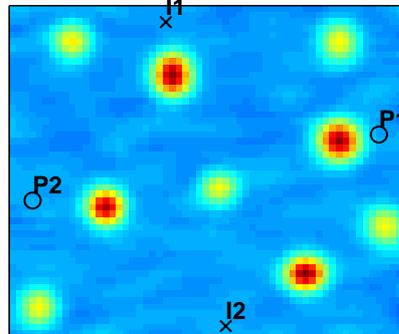


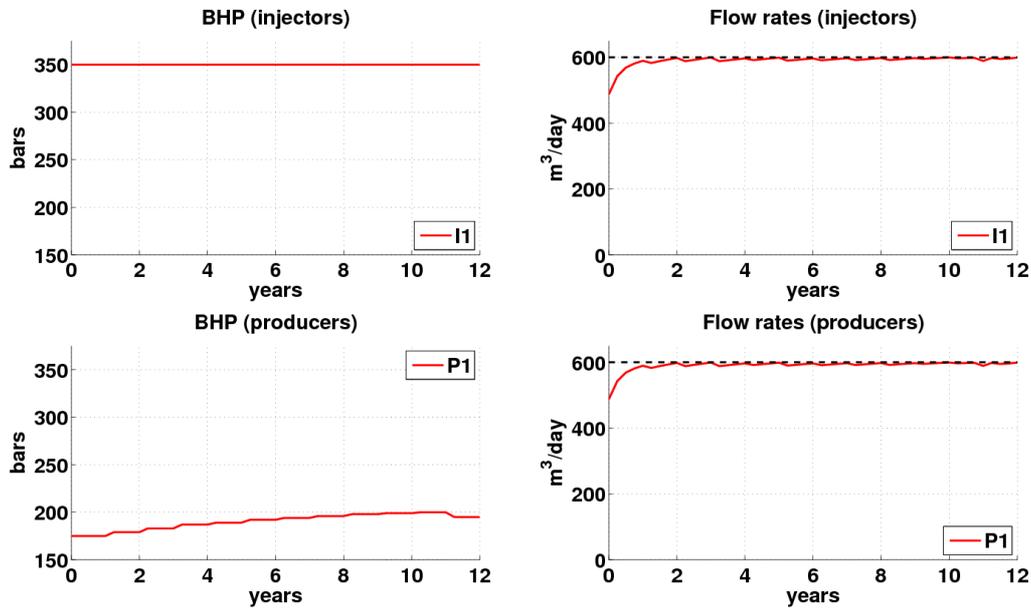
Figure 6 Best well placements found in each of the four problems in Experiment 1. Injection wells shown as black \times and producers as black \circ .

in Problem 1A was also placed in a high-permeability region, while the two injectors in Problem 1C had to be placed farther away from the producers in order to delay breakthrough of water. The best positions found in Problems 1B and 1D are noticeably different from the unconstrained problems, due to the effect of the maximum flow constraint of $600 \text{ m}^3/\text{day}$. While generating a high flow rate was the primary consideration in the unconstrained problems, in the constrained problems it was important to delay the water breakthrough for as long as possible, while also reaching the maximum flow rate quickly. The optimal solutions for Problems 1B and 1D therefore tended to place the wells farther away from one another than did the solutions to the corresponding unconstrained problems.

The optimal control strategies for Problems 1A and 1C were simply to hold producers at the lowest BHP (175 bars) until the water cut exceeded the profitable point of 78%, after which point the wells were shut in. This result is consistent with previous studies (e.g. Zandvliet et al. (2007)) which indicate that “bang-bang” control is the optimal choice in this case. Shutting in the producer was not necessary in Problem 1A, but in Problem 1C the second producer (denoted P2 in Figure 6, plot 1C) was shut in after 10 years. The optimal control in Problems 1B and 1D, on the other hand, required eventually raising the BHP at the producers, in order to maintain a flow rate below the maximum of $600 \text{ m}^3/\text{day}$. The optimal controls for these two problems, corresponding to the well placements shown in Figure 6, are shown in Figure 7.

1B (Two wells, constrained)

NPV = $\$6.92 \times 10^8$



1D (Four wells, constrained)

NPV = $\$12.5 \times 10^8$

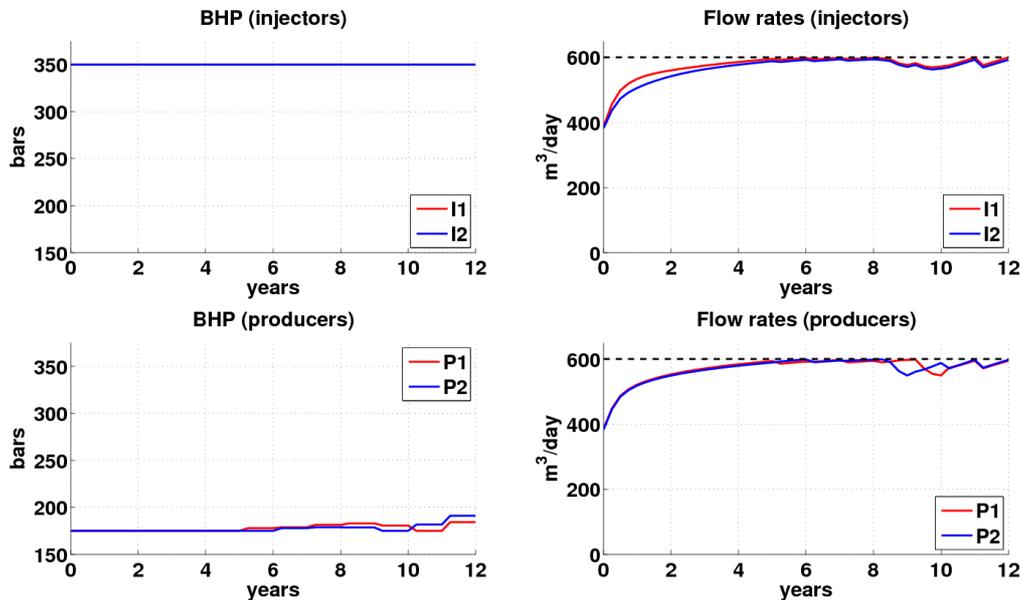


Figure 7 Bottom hole pressures (left plots) and flow rates (right plots) at the wells for the best solutions found in Problems 1B and 1D. Black dashed line on plots of flow rates indicates the maximum flow constraint of 600 m³/day. Some curves overlap.

Experiment 2

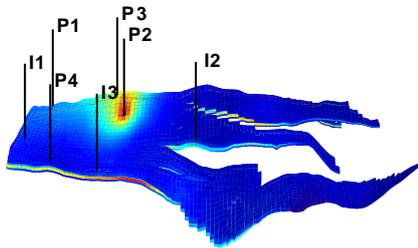
The results of the second experiment indicate very little difference between the performances of PSO and the hybrid algorithm for this problem. The quality of the solutions eventually found by either algorithm (Table 2) was essentially the same for Problem 2A, while for Problem 2B, the solutions found by the hybrid algorithm were roughly 8% better, on average. The number of function evaluations (fevals) used by the hybrid algorithm, however, was significantly larger, even though only 150 iterations of the hybrid algorithm were performed, compared to 200 iterations of PSO. This is due to the fact that polling was a very expensive operation in this experiment, requiring 156 fevals per polling step (twice the total number of variables). Thus, when convergence is plotted against the number of fevals, as shown in Figure 5, it is apparent that the performance of the two algorithms was essentially the same in both experiments. This is in contrast to the results of Experiment 1, where the performance of the hybrid algorithm was generally better. We note as well that in all optimization runs for Experiment 2, the algorithms were terminated early due to reaching the maximum number of iterations. Thus, the solutions found were not locally optimal and could usually be improved significantly by performing GPS afterwards, as indicated by the $\Delta\%$ values shown in Table 2. These values were significantly smaller for the hybrid algorithm than for PSO in both cases, however, indicating that the solutions found were closer to being locally optimal, as in Experiment 1. While it would have been desirable to run both algorithms for more iterations, this was not feasible due to the high computational cost of the reservoir simulations in this experiment.

The decoupled approach was far more effective than either of the other two methods when applied to Experiment 2B, as indicated by the results in Table 2 and by the convergence plot in Figure 5 (bottom right corner). It should be noted again that the decoupled approach was started from a much better initial solution than the other two approaches, since the BHPs at producers were held fixed at 175 bars, rather than initialized randomly in the range of 150-450 bars. Had the PSO and hybrid algorithms been given a better initialization, the gap in performance would likely have been smaller. This factor does not totally account for the improved performance of the decoupled approach, however. The convergence plot shows that after roughly 1500 fevals, the solutions found by the PSO and hybrid approaches were generally on par with the solutions used to initialize the decoupled method. As those algorithms proceeded, however, the convergence was clearly slower than that of the decoupled approach. The results indicate that in this larger-scale problem, there was a clear advantage gained from reducing the size of the problem from 78 variables to only 14 (the positions of the seven wells) by initially assuming a fixed control scheme.

The increased cost of polling in this experiment reduced the efficiency of the hybrid algorithm considerably. A possible solution to this issue would be to use a different local optimization method, such as SPSA (Bangerth et al. (2006)), which requires only two function evaluations per iteration, regardless of the number of variables involved in the problem. It should be noted that although we assessed the performance of the algorithms by measuring convergence versus the number of fevals, this may be a misleading evaluation when a large number of parallel resources are available. If 40 simulations can be performed in parallel, for instance, then the actual computational time to perform up to 160 fevals is only four times as great as that of performing two. Thus, the increased number of fevals required for GPS-style polling is not as significant an issue.

The best well positions and well control schemes found in Experiment 2 are shown in Figures 8 and 9, respectively. The effect of imposing flow constraints in this experiment is comparable to that observed in Experiment 1. The best solution found for Problem 2A was one that produced high flow rates by placing wells in areas of high permeability, even though that meant that the wells were clustered together and that water breakthrough at the producers occurred fairly early. Well P1 was eventually shut in when the water cut exceeded the profitable threshold. The best solution for Experiment 2B was one that maintained the flow rates at most of the wells near the maximum allowed value, meaning that wells did not have to be placed in the regions of highest permeability. The BHPs at the production wells also had to be adjusted slightly in later years in order to satisfy the constraints. The fact that three out of four producers and two

2A (Seven wells, no constraint)
 NPV = $\$9.21 \times 10^9$



2B (Seven wells, constrained)
 NPV = $\$8.73 \times 10^9$

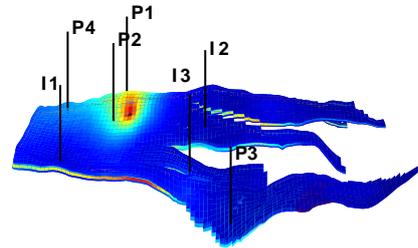


Figure 8 Best well positions found in Problems 2A and 2B.

out of three injectors were eventually close to the maximum flow rate indicates that the solution shown is a good one, although there is no guarantee that it is optimal.

Conclusions

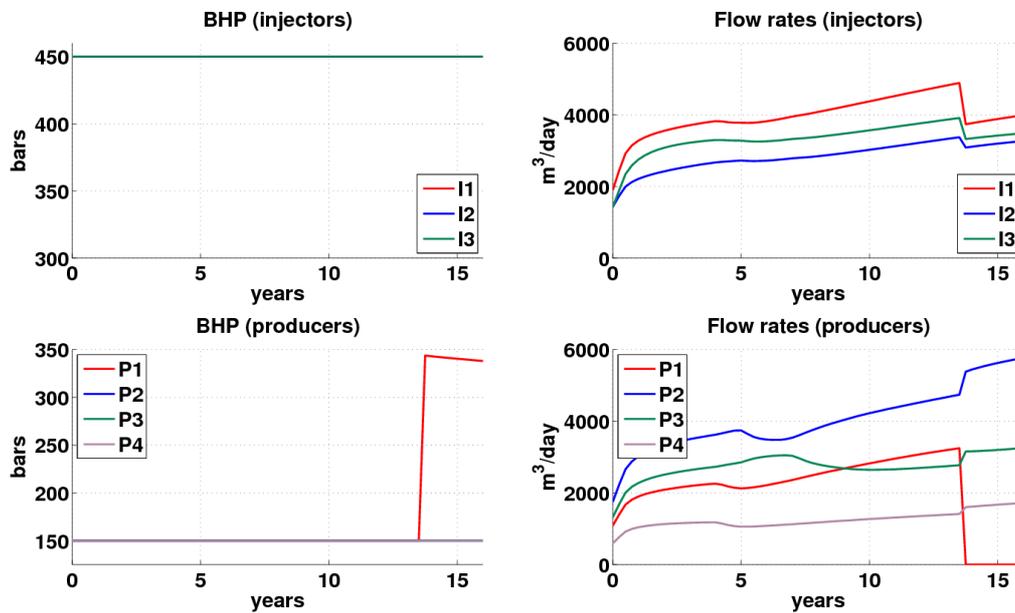
We have examined two approaches to simultaneous optimization of well placement and control, which combine particle swarm optimization (PSO) with pattern search (GPS). The first approach is a hybrid algorithm based on the previously proposed PSwarm algorithm (Vaz and Vicente (2009)), which acts on all variables of the problem simultaneously. The second approach is a decoupled method where PSO is applied initially to a well placement problem (assuming a fixed control scheme), and GPS is applied to the controls afterwards, once suitable positions have been established. These approaches were compared with the basic PSO algorithm in two sets of experiments, both of which consisted of placing vertical wells and controlling them using BHP. The first set of experiments involved placing and controlling up to four wells in a simple rectangular 2D reservoir model, while the second used a more realistic 3D reservoir model, and required placing and controlling seven wells. The objective in all the experiments was to maximize the NPV of the oil produced.

In the first set of experiments, the hybrid algorithm was found to have faster convergence than PSO in three out of the four test cases. (In the one other case, the performance of the two algorithms was essentially the same). The advantage of using the hybrid approach was especially noticeable in the fourth test case, which involved placing four wells and included nonlinear constraints on production. In the second set of experiments, there was little advantage to using the hybrid approach, largely as a result of the large numbers of variables in the problem, which increased the cost of polling. The decoupled approach was found to be the superior method for this problem, as it significantly outperformed both of the other approaches. In the first experiment, this approach provided slightly better results than the other two algorithms when applied to one problem, and comparable results for the other.

The results of our experiments suggest that the sequential, decoupled approach to optimizing well placement and control may be preferable to an approach that attempts to optimize over all variables simultaneously, especially if the number of variables is large. These experiments dealt with a fairly specific situation (placement of vertical wells controlled by BHP), however, and further studies are necessary to see if the same is true in more general problems. We also note that it is important that an appropriate fixed control scheme be chosen during the well placement step of the decoupled approach. When maximum flow constraints are present, for example, then holding production wells at the minimum BHP may result in suboptimal solutions, since wells may be placed farther apart than is necessary to avoid violating the constraints. Choosing a BHP near the high end of the range, on the other hand, may result in placing

2A (Seven wells, no constraints)

$$\text{NPV} = \$9.21 \times 10^9$$



2B (Seven wells, constrained)

$$\text{NPV} = \$8.73 \times 10^9$$

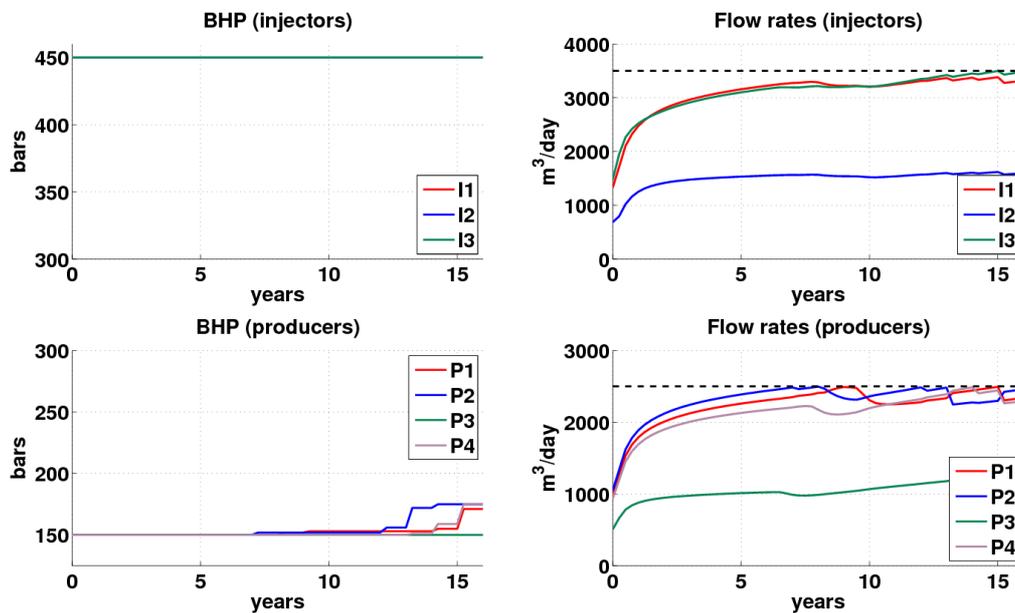


Figure 9 Bottom hole pressures (left plots) and flow rates (right plots) at the wells for the best solutions found for Problems 2A and 2B. Black dashed line on plots of flow rates for Experiment 2B indicate the maximum flow constraints of 3500 m³/day for injectors and 2500 m³/day for producers. Some curves overlap.

wells too close to one another, leading to solutions that suffer from early water breakthrough when BHPs are allowed to drop during the second stage of the optimization. We obtained good results in this study by choosing a fixed BHP near the lower end of the range for producers, but above the minimum value.

Acknowledgments

The authors acknowledge funding from the Natural Sciences and Engineering Research Council of Canada (NSERC), the Atlantic Canada Opportunities Agency (ACOA) and Research & Development Corporation of Newfoundland and Labrador (RDC). The authors thank the developers of MRST at SINTEF Applied Mathematics for making their software freely available and for providing support. The authors also thank StatoilHydro (operator of the Norne field) and its license partners ENI and Petoro for the release of the Norne data used in Experiment 2. Further, the authors acknowledge the Center for Integrated Operations at NTNU for cooperation and coordination of the Norne cases. The views expressed in this paper are the views of the authors and do not necessarily reflect the views of StatoilHydro and the Norne license partners.

References

- Artus, V., Durlofsky, L., Onwunalu, J. and Aziz, K. [2006] Optimization of nonconventional wells under uncertainty using statistical proxies. *Comput. Geosci.*, **10**(4), 389–404.
- Audet, C. and Dennis, J. [2004] A pattern search filter method for nonlinear programming without derivatives. *SIAM J. Optim.*, **14**(4), 980–1010.
- Bangerth, W., Klie, H., Wheeler, M., Stoffa, P. and Sen, M. [2006] On optimization algorithms for the reservoir oil well placement problem. *Comput. Geosci.*, **10**, 303–319.
- Bittencourt, A. and Horne, R. [1997] Reservoir development and design optimization. *Paper SPE 38895 presented at the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, 5-8 October.*
- Bouzarkouna, Z., Ding, D. and Auger, A. [2010] Using evolution strategy with meta-models for well placement optimization. *12th European Conference on Mathematics of Oil Recovery (ECMOR XII), Oxford, UK, 6-9 September.*
- Brouwer, D. and Jansen, J. [2002] Dynamic optimization of water flooding with smart wells using optimal control theory. *Paper SPE 78278 presented at the SPE 13th European Petroleum Conference, Aberdeen, Scotland, UK, 29-31 October.*
- Bukhamsin, A., Farshi, M. and Aziz, K. [2010] Optimization of multilateral well design and location in a real field using a continuous genetic algorithm. *Paper SPE 136944 presented at the SPE/DGS Annual Technical Symposium, Al-Khobar, Saudi Arabia, 4-7 April.*
- Ciaurri, D., Isebor, O. and Durlofsky, L. [2010] Application of derivative-free methodologies to generally constrained oil production optimization problems. *Procedia Computer Science*, **1**(1), 1301 – 1310.
- Ciaurri, D., Mukerji, T. and Durlofsky, L. [2011] Derivative-free optimization for oil field operations. In: Yang, X.S. and Koziel, S. (Eds.) *Computational Optimization and Applications in Engineering and Industry*. Springer Berlin / Heidelberg, vol. 359 of *Studies in Computational Intelligence*, 19–55.
- Clerc, M. [2006] *Particle Swarm Optimization*. iSTE, London.
- Emerick, A. et al. [2009] Well placement optimization using a genetic algorithm with nonlinear constraints. *Paper SPE 118808 presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, 2-4 February.*
- Güyagüler, B. and Horne, R. [2004] Uncertainty assessment of well-placement optimization. *SPE Reservoir Evaluation and Engineering*, **7**(1), 24–32.
- Hu, X. and Eberhart, R. [2002] Solving constrained nonlinear optimization problems with particle swarm optimization. In *Proceedings of the Sixth World Multiconference on Systemics, Cybernetics and Informatics*.
- Jansen, J., Douma, S., Brouwer, D., Van den Hof, P., Bosgra, O. and Heemink, A. [2009] Closed-loop reservoir management. *Paper SPE 119098 presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 2-4 February.*
- Kennedy, J. and Eberhart, R. [1995] Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, 1942–1948.
- Lewis, R. and Torczon, V. [1999] Pattern search algorithms for bound constrained minimization. *SIAM J. Optim.*, **9**(4), 1082–1099.
- Lie, K.A., Krogstad, S., Ligaarden, I., Natvig, J., Nilsen, H. and Skaflestad, B. [2011] Open-source MATLAB implementation of consistent discretisations on complex grids. *Computational Geosciences*, 1–26.
- Nogueira, P.d.B. and Schiozer, D. [2009] An efficient methodology of production strategy optimization based on genetic algorithms. *Paper SPE 122031 presented at the SPE Latin American and Caribbean Petroleum Engineering Conference, Cartagena, Colombia, 31 May - 3 June.*
- NTNU IO Centre [2012] Norne benchmark case. <http://www.ipt.ntnu.no/~norne/>.
- Onwunalu, J. and Durlofsky, L. [2010] Application of a particle swarm optimization algorithm for determining optimum well location and type. *Comput. Geosci.*, **14**, 183–198.
- Onwunalu, J. and Durlofsky, L. [2011] A new well-pattern-optimization procedure for large-scale field development. *SPE Journal*, **16**(3), 594–607.
- Ozdogan, U., Sahni, A., Yeten, B., Guyaguler, B. and Chen, W. [2005] Efficient assessment and optimization of

- a deepwater asset development using fixed pattern approach. *Paper SPE 95792 presented at the SPE Annual Technical Conference and Exhibition, Dallas, Texas, 9-12 October.*
- Peaceman, D. [1978] Interpretation of well-block pressures in numerical reservoir simulation. *SPE Journal*, **18**(3), 183–194.
- Peters, E., Arts, R., Brouwer, G. and Geel, C. [2009] Results of the Brugge benchmark study for flooding optimization and history matching. *Paper SPE 119094 presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 2-4 February.*
- Sarma, P., Chen, W., Durlofsky, L. and Aziz, K. [2006] Production optimization with adjoint models under non-linear control-state path inequality constraints. *Paper SPE 99959 presented at the SPE Intelligent Energy Conference and Exhibition, Amsterdam, The Netherlands, 11-13 April.*
- SINTEF Applied Mathematics [2011] Matlab reservoir simulator toolbox v. 2011a. <http://www.sintef.no/Projectweb/MRST/>.
- Sudaryanto, B. and Yortsos, Y. [2001] Optimization of displacements in porous media using rate control. *Paper SPE 71509 presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, 30 September-3 October.*
- van Essen, G., Van den Hof, P. and Jansen, J. [2011] Hierarchical long-term and short-term production optimization. *SPE Journal*, **16**(1), 191–199.
- Vaz, A. and Vicente, L.N. [2007] A particle swarm pattern search method for bound constrained global optimization. *J. Glob. Optim.*, **39**(2), 197–219.
- Vaz, A. and Vicente, L. [2009] Pswarm: a hybrid solver for linearly constrained global derivative-free optimization. *Optimization Methods and Software*, **24**(4-5), 669–685.
- Wang, C., Li, G. and Reynolds, A. [2009] Production optimization in closed-loop reservoir management. *SPE Journal*, **14**(3), 506–523.
- Yang, D., Zhang, Q. and Gu, Y. [2003] Integrated optimization and control of the production-injection operation systems for hydrocarbon reservoirs. *Journal of Petroleum Science and Engineering*, **37**, 69–81.
- Yeten, B., Durlofsky, L. and Aziz, K. [2003] Optimization of nonconventional well type, location and trajectory. *SPE Journal*, **8**(3), 200–210.
- Zandvliet, M., Bosgra, O., Jansen, J., Van den Hof, P. and Kraaijevanger, J. [2007] Bang-bang control and singular arcs in reservoir flooding. *Journal of Petroleum Science and Engineering*, **58**, 186–200.
- Zandvliet, M., Handels, M., van Essen, G., Brouwer, D. and Jansen, J. [2008] Adjoint-based well-placement optimization under production constraints. *SPE Journal*, **13**(4), 392–399.