Proceedings of the Twenty-Ninth AAAI Conference on Innovative Applications (IAAI-17))

Optimal Sequential Drilling for Hydrocarbon Field Development Planning

Ruben Rodriguez Torrado
Repsol S.A.
Mostoles, Spain

Jesus Rios and **Gerald Tesauro**
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA

Abstract

We present a novel approach for planning the development of hydrocarbon fields, taking into account the sequential nature of well drilling decisions and the possibility to react to future information. In a dynamic fashion, we want to optimally decide where to drill each well conditional on every possible piece of information that could be obtained from previous wells. We formulate this sequential drilling optimization problem as a POMDP, and propose an algorithm to search for an optimal drilling policy. We show that our new approach leads to better results compared to the current standard in the oil and gas (O&G) industry.

1 Introduction

The life cycle of a reservoir consists mainly of five phases: exploration, appraisal, development, production and abandonment. It is the development phase that is the most important one from a decision making viewpoint since the largest investments are associated with this phase. In addition, the great amounts of uncertainty about the subsurface of the reservoir make these investment decisions very challenging. In this context, the evaluation and selection of the best development plan is critical to guarantee the profitability of a reservoir.

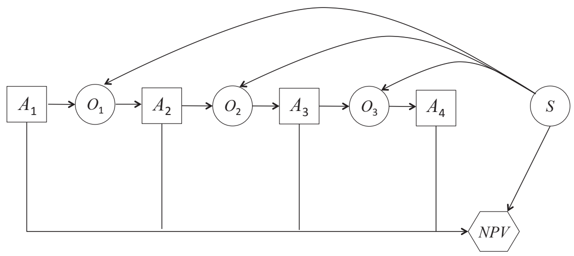
The reservoir is typically modeled as a grid with petrophysical property values associated to each cell. The exact values for these properties are obviously unknown. Uncertainty around the reservoir's petrophysical properties is modeled probabilistically by estimating a geostatistical model from well and seismic data obtained during the appraisal phase. A field development plan (FDP) determines the production strategy, specifying, among other things, the location and drilling schedule of wells. Reservoir flow simulators (Batycky et al., 2007) are used to forecast field production rates for any production strategy and petrophysical properties. This allows the estimation of the economic value of an FDP candidate.

Field development planning (FDP) can be formulated as an optimization problem under uncertainty. The current state-of-the-art for optimizing an FDP formulates the decision making problem as a *static* stochastic mixed-integer nonlinear program (MINLP), where the objective function is typically estimated running a computationally expensive black-box reservoir flow simulator over a set reservoir's property realizations sampled from the geostatistical model. Several methods based on local optimization and heuristic global search approaches have been proposed for solving this MINLP formulation (Davidson and Beckner, 2003; Litvak et al., 2002; Wang, 2003), all requiring an excessive number of estimations of the objective function and therefore impractical in many real applications. In particular, several authors, such as Artus et al. (2006) and Bangerth et al. (2006) have favored genetic algorithms among these methods when comparing their performance in several examples.

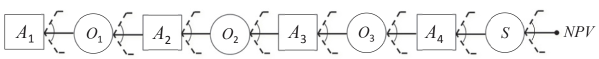
However, these optimization approaches for FDP do not take into consideration the sequential nature of the problem. In practice, drilling location decisions alternate with the observation of information acquired between drillings. For example, petrophysical properties at the location of each well become known after the well is drilled and this localized information is used to infer properties in the nearby area through spatial correlations of the geostatistical model. We propose a dynamic programing approach to model the opportunity of adapting the drilling plan to the information that is revealed as wells are drilled.

2 Sequential field development planning

We consider, without loss of generality, the following field development optimization problem. Suppose we need to decide the location and drilling schedule of $n = 4$ wells so that the economic value obtained from the reservoir is maximized. We assume access to only one drilling rig and therefore that wells are drilled sequentially. Once a well is completed, information about the type of rock (facies) at the well location is revealed before the next well is drilled. The time taken to drill and analyze the core samples and well logs needed to uncover facies is assumed to be 3 months. Thus, the first well is drilled at time $t = 0$, and the next wells are drilled in intervals of 3 months. In this setting, we must optimize the well locations as well as the order in which wells are drilled. The drilling order implicitly determines the new information that is observed at each decision step, which is used to improve the knowledge of the reservoir before the next well decision. In particular, facies are observed at the



(a) Influence diagram



(b) Decision tree

Figure 1: The sequential FDP decision problem (for $n = 4$)

location of a well right after it is drilled, and used to update the geostatistical model. A policy will tell us where to drill the first well, as well as all the next ones conditional on the facies observed at all the previously drilled sites.

Figure 1 shows the sequential FDP decision problem represented both as an influence diagram and as decision tree. The value node NPV(a, s) in the influence diagram, shown as a hexagon, represents the economic valuation of a production strategy consisting of a sequence of wells drilled at locations $a = (a_1, a_2, a_3, a_4)$ for a realization $S = s$ of the reservoir's petrophysical properties. This evaluation typically requires running a computationally expensive 'black-box' reservoir flow simulator to obtain simulated production rates that are converted to cash flows when passed through an economic model before computing their Net Present Value (NPV). The company's risk attitude (Pratt, 1964) is modeled by a utility function $u(\mathit{NPV})$. Thus, the company is assumed to prefer $A = a$ with higher expected utility and therefore the objective is to maximize expected utility.

A history $h_t = (a_1, o_1, \ldots a_t, o_t)$ is a sequence of well location decisions $A_i = a_i$ and observations of facies $O_i = o_i$ at wells $i = 1, \ldots, t$. Decisions at node A_{t+1} are taken knowing history h_t. The chance node S represents the state of the reservoir's petrophysical properties and its probability distribution $p(S)$ is given by the geostatistical model. Next observation probabilities $p(O_{t+1}|h_t, a_{t+1})$ correspond to the marginals at location a_{t+1} of the updated geostatistical model $p(S|h_t)$. Updating the geostatistical model requires reversing the arcs from chance node S to $O_1, \ldots O_t$ using Bayes rule.

In theory, an optimal FDP can be found applying the standard decision analytic approach (French and Rios Insua, 2000) based on dynamic programming. For example, the optimization problem to solve for $n = 4$ is

$$\max_{a_1} \sum_{o_1} \max_{a_2} \sum_{o_2} \max_{a_3} \sum_{o_3} \left[\max_{a_4} \sum_{s} u(a, s)\, p(s|h_3) \right] p(o_3|h_2, a_3)\, p(o_2|h_1, a_2)\, p(o_1|a_1) \quad (1)$$

Specifically, to solve (1) we would proceed as follows

1. Compute $p(S = s|h_3) \propto p(s)\, p(h_3|s)$, for all h_3 by inverting arcs from S to O_1, O_2, and O_3.
2. For every history (h_3, a_4), compute the expected utility $\psi(h_3, a_4) = \sum_s u(a, s)\, p(s|h_3)$ by eliminating chance node S.
3. At decision node A_4, for every history h_3 find $a_4^*(h_3) = \arg\max_{a_4 \in \mathcal{A}_4(h_3)} \psi(h_3, a_4)$ to obtain the optimal policy for the location of the fourth well, where $\mathcal{A}_4(h_3)$ represents the set of feasible locations at decision A_4 given h_3. Node A_4 is then eliminated by computing $\psi(h_3) = \psi(h_3, a_4^*)$, the expected utility of a_4^* for every history h_3.
4. Eliminate chance node O_3 by computing $\psi(h_2, a_3) = \sum_{o_3} \psi(h_2, a_3, o_3)\, p(o_3|h_2, a_3)$, the expected utility of every history (h_2, a_3).
5. Eliminate decision node A_3 by expected utility maximization to obtain the third well location policy $a_3^*(h_2) = \arg\max_{a_3 \in \mathcal{A}_3(h_2)} \psi(h_2, a_3)$ and its expected utility $\psi(h_2) = \psi(h_2, a_3^*)$ for every history h_2.
6. Eliminate nodes O_2 and A_2 by computing $\psi(h_1, a_2) = \sum_{o_2} \psi(h_1, a_2, o_2)\, p(o_2|h_1, a_2)$, for all $(h_1\, a_2)$; and finding $a_2^*(h_1) = \arg\max_{a_2 \in \mathcal{A}_2(h_1)} \psi(h_1, a_2)$, for all h_1, whose expected utility is $\psi(h_1) = \psi(h_1, a_2^*)$.
7. Eliminate nodes O_1 and A_1 in order to obtain $a_1^* = \arg\max_{a_1 \in \mathcal{A}_1} \sum_{o_1} \psi(a_1, o_1)\, p(o_1|a_1)$, the first well's optimal location, and its expected utility $\psi^* = \sum_{o_1} \psi(a_1^*, o_1)\, p(o_1|a_1^*)$

Thus by using backward induction, we have found that a_1^*, $a_2^*(a_1^*, o_1)$, $a_3^*(a_1^*, o_1, a_2^*, o_2)$, and $a_4^*(a_1^*, o_1, a_2^*, o_2, a_3^*, o_3)$ constitutes an optimal well location policy a^*. It specifies the best location for the first well, and every consecutive well as a function of any future observations from previously drilled optimal wells.

To measure the computational complexity of this procedure, we can use the total number of branches $(a_1, o_1, \ldots, a_{n-1}, o_{n-1}, a_n, s)$ that need to be evaluated in a decision tree like the one in Figure 1. For a reservoir represented by a 2-dimensional $n_x \times n_y$ grid, a sequential drilling problem with n well decisions, n_{obs} possible observations and n_s realizations representing the reservoir's state space of petrophysical properties, would require approximately a total of $(n_x \times n_y)^n\, n_{obs}^{n-1}\, n_s$ NPV evaluations at the leaf nodes of such a tree. Thus, for example, let us suppose we are to decide the location and drilling order of $n = 4$ wells in a 20×20 reservoir whose uncertainty is characterized by $n_s = 1,000$ possible reservoir realizations, and where we can observe at each drilling site one of $n_{obs} = 2$ types of facies. Then, we would need approximately $400^4 \times 2^{4-1} \times 1,000 = 2.048 \times 10^{14}$ NPV evaluations to start. In particular, there are approx. 2×10^{11} combination of possible histories that have to be evaluated against the set of n_s possible reservoir realizations. If each evaluation takes approximately 2 seconds, for such a small case, this would already require over 12 million years to compute.

2.1 POMDP formulation

Now we formulate the sequential FDP optimization problem as a Partially Observable Markov Decision Problem (POMDP). The world environment under consideration is the reservoir with set of possible states consisting of all the possible realizations of petrophysical properties for that reservoir. We have n decision epochs at periods $t = 1, \ldots, n$. The set of possible actions $\mathcal{A}_t$ to choose from at each decision epoch t corresponds with all the possible locations in the reservoir grid at which a well can be drilled. Since wells cannot be drilled very close to each other, we need to consider location constraints defining $\mathcal{A}_t(h_{t-1})$ as a function of all previously chosen wells $a_1, \ldots, a_{t-1}$ specified by history h_{t-1}. We have assumed so far that we will always choose to drill a well at every decision epoch t and, therefore, that we do not have the option of not to drill. The set of possible observations at epoch t corresponds with all the possible petrophysical properties that can be revealed at location a_t after drilling.

Since drilling actions do not affect the state of the reservoir's petrophysical properties, the conditional *transition probabilities* would thus be $p(S_{t+1} = s'|S_t = s, A_t = a) = 1$ iff $s' = s$, and zero otherwise. The petrophysical properties are static, and hence do not change over time. Our initial beliefs over these properties are given by the *initial state probabilities* $p(S_0 = s)$, which corresponds to the spatial geostatistical model of the reservoir given by $p(S)$. While the exact state of the reservoir is not observable, partial observations of this state are available at every epoch in the location where the well is drilled. For a reservoir's state $S = s$ and a well drilling location $A_t = a_t$, the *observation* is deterministic and given by $o_t = s[a_t]$, the petrophysical property at location a_t. The conditional *observation probabilities* at a given location $p(O_t|A_t = a_t)$ correspond with the marginal of the geostatistical model $p(S_t = s)$ at the location $A_t = a_t$. The state $S = s$ of the reservoir and the observations $O_t = o_t$ can be simulated from the probabilities of the geostatistical model. Observations from drilled wells update our beliefs over the state of the reservoir's petrophysical properties. Given the observations in history h_t, Bayes rule can be used to update the beliefs over the state of the reservoir $p(S_t = s|h_t)$.

The *return* corresponds with the economic performance of the reservoir, measured as the NPV of the generated cash flows. The NPV is equivalent to the total discounted *reward* accumulated through all the decision periods. The return NPV(a, s) is then calculated after all decisions have been made and is a function of the actions $a = (a_1, \ldots, a_n)$ in a complete history and the realization $S_n = s$ of the final state of the reservoir. This allows for the comparison of different development strategies over the productive life of the reservoir.

A development *policy* specifies which action to take for every possible history. We allow for random distributions over the set of possible actions. An optimal policy is a solution of (1) and then the optimal value function $V^*(h_t) = \psi(h_t, a^*_{t+1})$ corresponds with the maximum achievable expected return given history h_t (by following the optimal policy a^*) .

3 Partially observable Monte-Carlo development planning

We present here a novel simulation-based search algorithm to find optimal field development policies. The basic idea is to start with a random rollout policy and iteratively modify it as we simulate and evaluate histories generated from these policies, until convergence is detected.

3.1 A PO-UCT based algorithm

We adapted the Partially Observable Monte-Carlo Planing (POMCP) algorithm introduced by Silver and Veness (2010) to solve the sequential field development planning problem. See Algorithm 1. We aim to find a policy that optimizes the well locations and drilling schedule for the POMDP formulation described in Section 2.1. To initialize the algorithm we need to specify (i) a characterization of the reservoir, including a probabilistic geostatistical model quantifying the uncertainty around the petrophysical properties; (ii) an initial set of possible well locations along any location constraints; and (iii) an exploration/exploitation trade-off parameter $c > 0$.

As with the PO-UCT algorithm, we make use of a *search tree* of histories. The nodes in this tree represent partial histories. As complete Monte Carlo Tree Search (MCTS) histories are simulated, each node of the search tree keeps track of $N(h)$, the total number of times a partial history h has been visited, and $V(h)$ the average return of all simulations starting with h. Each MCTS simulation starts at the root node ($h = \emptyset$) of the search tree. Actions a_t and observations o_t are then simulated sequentially at $t = 1, \ldots n$. The MCTS simulation ends with the last action a_n and a simulation s from the updated geostatiscial model $p(S|h_{n-1})$. The reward of a MCTS simulation is then computed as the NPV(a, s), where the production strategy $a = (a_1, \ldots, a_n)$ is obtained from the simulated actions.

The selection of action a_{t+1} in a MCTS simulation depends on whether h_t is a node in the search tree. If this is the case and all the feasible locations in the action set $\mathcal{A}_{t+1}(h_t)$ have been visited at least once in previous simulations, then the UCB1 policy (Auer et al., 2002) is used to select the action

$$a_{t+1} = \mathit{argmax}_{a \in \mathcal{A}_{t+1}(h_t)} V(h_t a) + c\sqrt{\frac{\mathit{log} N(h_t)}{N(h_t a)}}$$

This is our tree policy, which is used as far as nodes $h_t a$ exist in the search tree as children of node h_t, for all $a \in \mathcal{A}_{t+1}(h_t)$. Otherwise, actions are selected by a Monte Carlo rollout policy. In our implementation, the rollout policy selects actions randomly following a uniform probability distribution over the set of feasible actions: $a_{t+1} \sim \mathcal{U}\{\mathcal{A}_{t+1}(h_t)\}$.

The search tree is expanded at every MCTS simulation that generates a complete history that is not already in the search tree. When a MCTS simulation trial reaches a leaf node, we add to the search tree new nodes corresponding to the next action and/or observation encountered by that simulation.

Algorithm 1 Partially Observable Monte-Carlo Development Planning (POMCDP) algorithm

specify
Geostatistical model: $p(S)$
Action space $\mathcal{A}$ (initial feasible locations) and well location constrains
Exploration/exploitation parameter: $c > 0$
repeat
1. Simulate reservoir realization $s \sim p(S)$
2. MCTS simulation
$\triangleright$ $t = 0$, $h_0 = \emptyset$, $\mathcal{A}_0 = \mathcal{A}$
Start at the root node of the Tree Search
$\triangleright$ Tree policy stage while in Tree Search
while all children of h_t visited **do**
$a_{t+1} = \textit{argmax}_{a \in \mathcal{A}_{t+1}} V(h_t a) + c\sqrt{\frac{logN(h_t)}{N(h_t a)}}$
update Action Space: $\mathcal{A}_{t+2}$
Simulate $o_{t+1} = s[a_{t+1}] \sim p(S_{t+1}|h_t, a_{t+1})$
$t = t + 1$
end while
$\triangleright$ Expand Tree Search
if h_t first time visited **then** add h_t
add $h_t a_{t+1}$ and h_{t+1} with first rollout iteration
$N(h^{new}) = V(h^{new}) = 0$
$\triangleright$ Rollout policy
while $\mathcal{A}_{t+1} \neq \emptyset$ AND $t \leq n$ **do**
Choose well positions: $a_{t+1} \sim \mathcal{U}\{\mathcal{A}_{t+1}\}$
update Action Space: $\mathcal{A}_{t+2}$
$t = t + 1$
end while
3. Evaluate NPV(a, s)
where a = (tree policy, rollout policy)
4. Backpropagation
update Tree Search for all $h \subset h^{new}$
$N(h) = N(h) + 1$
$V(h) = V(h)\frac{N(h)-1}{N(h)} + \text{NPV}\frac{1}{N(h)}$
until detecting convergence or timeout

Our algorithm iteratively learns how to select best drilling strategies by modifying the values of the N and V tree search functions for each MCTS iteration using backpropagation as shown in Algorithm 1. At first the simulated actions are random. As more MCTS simulations are performed, the action selection policy progressively improves. In particular, as for the POMCP algorithm, it is guaranteed that for a suitable parameter c this policy asymptotically converges (in probability) to an optimal FDP policy, and $V(h)$ to the optimal value function.

3.2 Parallelization

One of the main empirical limitations of our POMCDP algorithm is the computational time necessary to converge to the optimal value function. Specifically, the NPV calculation at each MCTS simulation is very time consuming and this limits the number of MCTS simulations that can be performed per time unit and, therefore, the time to convergence. In addition, the breadth of the tree at its action nodes given by the cardinality of the action space, e.g. $|\mathcal{A}_{t+1}(h_t)| \approx n_x \times n_y$ for a 2D case, increases dramatically the time needed for the tree policy to kick in for every new node added to the tree search, since this will not happen until all its children have been visited by a MCTS simulation at least once. Several authors e.g. Kocsis and Szepesvari (2006) have proposed the use of domain knowledge to initialize (N, V) in new nodes and narrow the search to the most promising actions. However, this strategy is difficult to implement in our case because there is no clear intuition or enough data to fit an evaluation function that will be able to substitute the reservoir flow simulator (Zubarev, 2009; Jurecka, 2007).

Our Parallel-POMCDP (P-POMCDP) algorithm modifies Algorithm 1 by applying an *at-the-leaves* parallelization approach (Cazenave and Jouandeau, 2007). Every time the search tree must be expanded at one of its leaf nodes $h_t = (a_1, o_1, \ldots, a_t, o_t)$ we, in parallel, add to the tree as many new child action nodes $h_t a_{t+1}$ as alternatives in the set $\mathcal{A}_{t+1}(h_t)$. In particular, for all $a_{t+1} \in \mathcal{A}_{t+1}(h_t)$ we simulate from $h_t\, a_{t+1}$ in parallel (and independently of each other) the remaining sequence of actions and observations. Access to a multicore processor or computer cluster allows processing these simulation jobs in parallel by running simultaneously threads in different cores and reducing the total computational time of this step. This parallel rollout policy is only applied at the expansion stage when a new action needs to be added to the tree search. After that, the standard serial rollout policy is applied within each of the parallel simulations. At the beginning of each simulation branch the geostatistical simulator is called to obtain each branch's own reservoir's petrophysical realization conditional on h_t. The observations at each sampled well location are then obtained from these simulated realizations. Then, the NPV associated with each parallel simulation branch is calculated using the reservoir flow simulator. Once all of these parallel simulations are finished, their results are propagated backwards through the search tree and the algorithm goes back to serial mode. Specifically, for all nodes $h \subset h_t$ we update $V(h) = (N(h)\, V(h) + m\, \overline{NPV})/(N(h) + m)$ and $N(h) = N(h) + m$, where m is the cardinal of the set $\mathcal{A}_{t+1}(h_t)$ and $\overline{NPV}$ represents the average NPV across all parallel m simulations started from h_t.

We note that in our case each parallel simulation will take very similar time to run. So if we have k available processors to run simulations simultaneously, the improvement in the estimation of the value function within a history h_t is done with $\lfloor m/k \rfloor$ extra computational cost. In particular, if $k > m$ the improvement will be achieved at basically no extra computational time. On the other hand, this parallelization only increases the exploration of actions and not observations, speeding-up the initialization of tree nodes created by the addition an action succeeding leaf nodes that represent a history h_t ending in an observation. Thus, this does not increase the exploration or initialization of unvisited observations after leaf nodes $h_t a_{t+1}$ representing histories ending in an action. This ensures we do not spend too much time exploring unpromising actions.

3.3 Implementation

We have implemented Algorithm 1 in Matlab. We used the AD-GPRS software (Cao, 2002) as reservoir flow simulator and the SGeMS software (Remy et al., 2009) to simulate realizations from the geostatistial model. The parallelization is done in a cluster with 6 nodes and a total of 216 processors. The cluster's nodes run in CentOS 6 with the IBM *Platform Load Sharing Facility* (LSF) software as workload manager in charge of controlling jobs dispatch and scheduling. Parallel simulations are prepared in Matlab by generating the data and batch scripts to be sent to the cluster for execution.

A computation of NPV(a, s) requires an AD-GPRS simulation which takes approximately an average of 2 secs in our implementation. This is the most time-consuming computation in each MCTS iteration and the bottleneck for any optimization approach. The output of this simulation is a deterministic prediction of the production profile for a given (a, s), which is later evaluated by an economic model to finally obtain the NPV(a, s).

4 Results

We tested our FDP sequential optimization approach with a numerical example. The selected case is based on a realistic reservoir represented by a 20×20 grid with two types of facies: sandstone and shale. These facies have high and low porosity respectively and, therefore, yield different levels of productivity. The reservoir's depositional environment consists of meandering channels where oil is trapped after generation and migration. Thus, we have a geostatistical model of a reservoir with non-obvious features that would need to be learnt by the optimization algorithm in order to come up with a good solution.

We want to decide on the location and drilling order of $n = 4$ wells. To reduce the search space, we imposed that wells cannot be less than 6 cells distance from each other, since it is known that it is not economically advantageous to do so in practice due to interferences between wells. After drilling a well, we observe the type of rock (facie) at that well's location. We have only two possible types of observations: sandstone or shale. The return of a well location policy is measured by its expected NPV. This problem has approximately 2×10^{11} possible histories.

Figure 2 compares the results obtained with (i) POMCDP, described in Algorithm 1, (ii) P-POMCDP, our parallel version of it, and (iii) a standard static optimization approach used in the O&G industry, specifically a parallel genetic optimization algorithm (P-GA), see e.g. de Andrade Filho (2010) for details. We have run each algorithm for 24 hours (86,400 sec), our hypothetical time before having to make a decision. During this time, POMCDP ran a total of approximately 27,000 serial MCTS simulation iterations, and P-POMCDP ran 400 MCTS iterations, in which an average of 320 simulations at each iteration were performed in parallel, for a total of approx. 126,000 NPV evaluations. The exploration parameter c for our UCT-based algorithms, POMCDP and P-POMCDP, were set to different value schedules. These parameter schedules were the ones that give us the best results after a few trials and errors. For the genetic algorithm we also fixed its meta-parameters in a similar way.

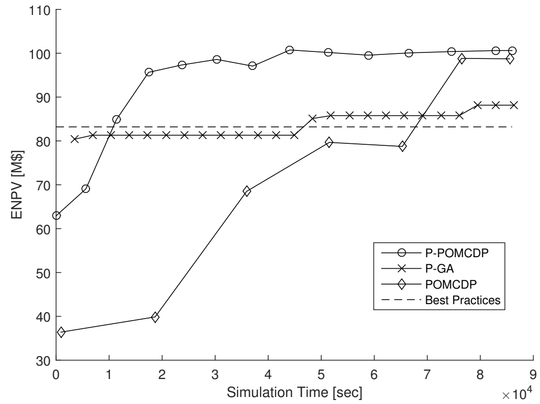


Figure 2: Convergence of P-POMCDP, POMCDP, and P-GA. For P-POMCDP and POMCDP each point in the graph shows the expected NPV of the policy at the corresponding simulation time. For P-GA each point shows the expected NPV of the best solution found so far. Best practices represent the best FDP suggested by subject matter experts.

We can see in Figure 2 that in our implementation P-POMCDP achieved the best results, followed by POMCDP, both with approximately 13% higher expected NPV than P-GA. Our P-POMCDP not only reached a slightly better optimal value than POMCDP, but it also converged faster. As a baseline to benchmark these results, we used the best FDP identified by our O&G industry experts. The expected NPV of the expert's solution is represented by a dashed horizontal line. All the optimization approaches were able to find better solutions than the one identified by the human experts.

Finally, the increment in expected NPV from the optimal solution in P-GA to the one in P-POMCDP corresponds to the value of the information (VoI) that can be obtained when drilling wells in a sequential fashion. By comparing this VoI with the cost of obtaining this information, we can decide whether it is worth to invest in information acquisition procedures such as core sample analysis, well logging, etc.

5 Conclusion

Planning the development of an oil field is a very challenging task. Typical approaches do not take into account that in actuality the decisions made during the development of a field are sequential in nature. Their models fail to consider the value of obtaining information that can be used for future drilling decisions, leading to suboptimal planning. To address these issues, we have formulated the optimization of the location and drilling schedule of wells as a POMDP and adapted the POMCP algorithm to search for optimal development policies. Since the evaluation of simulated histories in the Monte Carlo Tree Search is quite computationally expensive, we have proposed a new parallelization approach to improve the computational efficiency of this algorithm.

We have illustrated our new approach with a numerical example for which POMCDP was able to find sequential FDP policies significantly better than the solutions obtained with industry-standard optimization algorithms. The example also showed how our parallelization (P-POMCDP) can find even better solutions in considerably less time.

References

Artus V., Durlofsky L.J., Onwumunalu,J and Aziz K. (2006) Optimization of nonconventional wells under uncertainty using statistic approaches. *Computational geosciences*, 10, 389–404

Auer, P., N. Cesa-Bianchi, and P. Fischer (2002) Finite-time analysis of the multiarmed bandit problem, *Machine learning*, 47, 2-3, 235-256.

Bangerth W. Kylie H. Wheeler M.D. Stoffa P.L. and Sen M.K. (2006) On optimization algorithms for the reservoir oil well placement problem. *Computational geosciences*, 10, 303–319.

Batycky R. P., Thiele, M. R., Coats, K.H. , Grindheim, A., Ponting, D. , Killough, J. E., Settari, T., Thomas, L. K., Wallis, J., Watts, J.W. and Whitson C. H. (2007) Reservoir Simulation, *Petroleum Engineering Handbook, Volume V: Reservoir Engineering and Petrophysics*, Ch. 17, 1399 – 1478. Holstein, E. D. (Ed). Society of Petroleum Engineers: Richardson, TX.

Cao, H. (2002) *Development of techniques for general purpose simulators*, Ph.D. thesis, Stanford University.

Cazenave T. and N. Jouandeau (2007) On the parallelization of UCT. In H.J. van den Herik, J.W.H.M. Uiterwijk, M.H.M. Winands, and M.P.D. Schadd, (Eds), *Proceedings of the Computer Games Workshop 2007 (CGW 2007)*, 93–101. Universiteit Maastricht, Maastricht, The Netherlands.

Davidson, J.E. and Beckner, B.L. (2003) Integrated optimization for rate allocation in reservoir simulation, *SPE Reservoir Simulation Symposium*, 79701, Houston, TX, USA.

de Andrade Filho, A. C. B. (1997) Optimizing hydrocarbon field development using a genetic algorithm based approach, *Doctoral dissertation*, Stanford University.

French, S., Rios Insua, D. (2000) *Statistical Decision Theory*, Arnold.

Jurecka, F. (2007) *Robust design optimization based on metamodeling techniques*, Aachen: Shaker.

Kocsis L., Szepesvari, C. (2006) Bandit based Monte-Carlo planning. In *15th European Conference on Machine Learning*, 282–293.

Litvak, M. L., Hutchins, L. A., Skinner, R. C., Darlow, B. L., Wood, R. C., Kuest, L. J., Prudhoe Bay (2002) E-Field Production Optimization System Based on Integrated Reservoir and Facility Simulation, *SPE Annual Technical Conference and Exhibition*, 77643, San Antonio, USA.

Pratt, J. W. (1964) Risk aversion in the small and in the large, *Econometrica*, 32, 1/2, 122–136.

Remy, N., A. Boucher, and J. Wu (2009) *Applied geostatistics with SGeMS: A user's guide*, Cambridge University Press.

Silver, D., Veness, J. (2010) Monte-Carlo planning in large POMDPs, *Advances in neural information processing systems*

Wang, P. (2003) Development and applications of production optimization techniques for petroleum fields, *PhD Thesis*, Palo Alto, CS, USA: Stanford University.

Zubarev, D. I. (2009) Pros and cons of applying proxy-models as a substitute for full reservoir simulations, *SPE Annual Technical Conference and Exhibition*, Society of Petroleum Engineers.