

Bayesian Optimization: A New Sample Efficient Workflow for Reservoir Optimization under Uncertainty

Peyman Kor^{*,a}, Aojie Hong^a, Reidar Brumer Bratvold^a

^a*Energy Resources Department, University of Stavanger, Stavanger, Norway*

Abstract

An underlying challenge in well-control optimization during field development is that flow simulation of a 3D, full physics grid-based model is computationally prohibitive. In a robust optimization setting, where flow simulation has to be repeated over a hundred(s) of geological realizations, performing a proper optimization workflow becomes impractical in many real-world cases. In this work, to alleviate this computational burden, a new sample-efficient optimization method is presented. In this context, sample efficiency means that the workflow needs a minimum number of forward model evaluations (flow-simulation in the case of reservoir optimization) while still being able to capture the global optimum of the objective function. Moreover, the workflow is appropriate for cases where precise analytical expression of the objective function is nonexistent. Such situations typically arise when the objective function is computed as the result of solving a large number of PDE(s), such as in reservoir-flow simulation. In this workflow, referred to as “Bayesian Optimization” the objective function for samples of decision variables is first computed using a proper design experiment. Then, a Gaussian Process (GP) is trained to mimic the surface of the objective function as a surrogate model. While balancing the exploration-exploitation dilemma, a new decision variable is queried from the surrogate model and a flow simulation is run for this query point. Later, the output of the flow-simulation is assimilated back to the surrogate model which is updated given the new data point. This process continues sequentially until termination criteria are reached. To validate the workflow and get better insight into the details of optimization steps, we first optimize a 1D problem. Then, the workflow is implemented for a 3D synthetic reservoir model in order to perform robust optimization in a realistic field scenario. Finally, a comparison of the workflow with two other commonly used algorithms in the literature, namely Particle Swarm Intelligence (PSO) and Genetic Algorithm (GA) is performed. The comparison shows that the workflow presented here will reach the same near-optimal solution achieved with GA and PSO, yet reduce computational time of the optimization 5X (times). We conclude that the method presented here significantly speeds up the optimization process leveraging a faster workflow for real-world 3D optimization tasks, potentially reducing CPU times by days or months, yet gives robust results that lead to a near-optimal solution.

Key words: Optimization, Gaussian Process, Probabilistic Modeling, Bayesian

*Corresponding Author, peyman.kor@uis.no

1. Introduction:

Well control optimization can be defined as given a pre-defined objective function, what are the best set of control decisions to make. The objective function is usually net present value or total oil recovery. These functions, generally rely on a reservoir simulator for the evaluation of proposed well control decision for the period of reservoir life cycle. On the other hand, the control decisions to make are usually well rates or bottom-hole pressure (BHPs). Given the objective function and decision to make to maximize the objective, the uncertainty in geological model are considered through optimizing the objective function over hundred(s) of geological realization. Known, as Robust optimization, Within RO, the geological uncertainties are represented by a set of realizations (i.e., an ensemble). The objective of RO is to find a control vector (e.g., a water injection scheme) the goal is to maximize the expected value of the objective function in contrast to deterministic case, where sources of uncertainty in geological model is ignored. Well control optimization typically poses challenges as the objective function is non-linear and non-convex, expensive to evaluate. Also, in RO setting, the optimization problem becomes computationally demanding as not one, rather many of geological models must be considered. This renders the optimization computationally expensive, if not prohibitive, in large-scale systems where (potentially) hundreds of wells are involved.

Literature of well control optimization can be viewed from two angles. In first the focus is what type of optimization algorithm have been used for this type of problem. Broadly, speaking the type of optimization algorithm could be divided two categories, gradient based and gradient-free.

(Sarma et al., 2005) applied adjoint-gradient based optimization to waterflooding problems. (van Essen et al., 2009) optimized hydrocarbon production under geological uncertainty (in RO setting), where an adjoint-based method is used for obtaining the gradient information. The adjoint-based procedure is more efficient because it uses gradients that are constructed efficiently from the underlying simulator. However, the adjoint-based method requires access to the source code of the reservoir simulator, which is seldom available for commercial simulators, and it is computationally intensive. Chen et al. (2009) introduced the ensemble-based optimization method (EnOpt), in which the gradient is approximated by the covariance between the objective function values and the control variables. Do and Reynolds (2013) analyzed the theoretical connections between EnOpt and other approximate gradient-based optimization methods. Having realized that it is unnecessary to approximate the ensemble mean to the sample mean as was done by Chen et al. (2009), Do and Reynolds (2013) used the ensemble mean in their EnOpt formulation. Stordal et al. (2016) had a theoretical look at EnOpt formulation and showed that EnOpt is as special case of ell-defined natural evolution strategy known as Gaussian Mutation. It is special case from this perspective that that EnOpt of Gaussian Mutation without evolution of covariance matrix \sum of multivariate Gaussian density.

On the other hand, gradient free methods represent a useful alternative when gradients are not available or are too expensive to compute. Can be divided into two major classes, stochastic and pattern search, these methods are noninvasive with respect to the simulator, though they are usually less efficient and require a larger number of function evaluations than adjoint-gradient methods.

Probably as first use of gradient free method in oil and gas industry (Harding et al., 1998) applied genetic algorithm (GA) to production scheduling of linked oil and gas field. Comparison study was performed and they showed the GA outperform simulated annealing (SA) and sequential quadratic programming (SQP) techniques, as well as a hybrid of the two. GA was utilized later by (Almeida et al., 2007) for the optimization of control valves in the intelligent wells. They found that significant profit was achieved in comparison to using conventional methods (no valve control). More recently, (Lushpeev and Margarit, 2018) applied Particle Swarm Optimization (PSO) to real case field to find optimum injection mode for brown field. Control variables were the change in injection rates of 3 injectors and results of field experiments shows the improve in relative recovery after applying the modified solution's of optimization algorithm.

Generalized Pattern-search (GPS) methods (Dennis, n.d.; Torczon, 1997) are another types of gradient free techniques has been has been applied in well control optimization problem. Pattern-search method relies on polling, at any particular iteration a stencil is centered at the current solution. The stencil comprises a set of directions such that at least one is a descent direction. If some of the points in the stencil represent an improvement in the cost function, the stencil is moved to one of these new solutions. GPS and including variante of it (Mesh Adaptive direct search) usually considered as local search methods but due to ability to parallization, it has been a one of the favorite method in well control optimization. (Asadollahi et al., 2014; Echeverría Ciaurri et al., 2010; Foroud and Seifi, 2016; Nwachukwu et al., 2018a). To overcome the locality of the GPS methods, hybrid of the PSO as global method and then using pattern search for local search in iterative (PSO-MADS) way as well proposed in the work of (de Brito and Durlofsky, 2020; Isebor et al., 2013).

1.1. Survey of surrogate-based papers in Onepetro database

In the previous section we reviewed commonly used optimization algorithm in well control optimization. However, it should be noted that even with low cost optimization method like Enopt, still performing full RO optimization is computationally prohibitive (we will discuss about detail of this computational burden more in Section 2). To make RO feasible, two general approaches have been introduced. The first set of techniques targets the flow problem. This generally entails approximating the high-fidelity flow simulation model with a lower-fidelity or reduced order model, or with a model based on simplified flow physics. To get insight how this line of research is evolving , we did a small text mining task. Mining the more than 100000 peer-reviewed

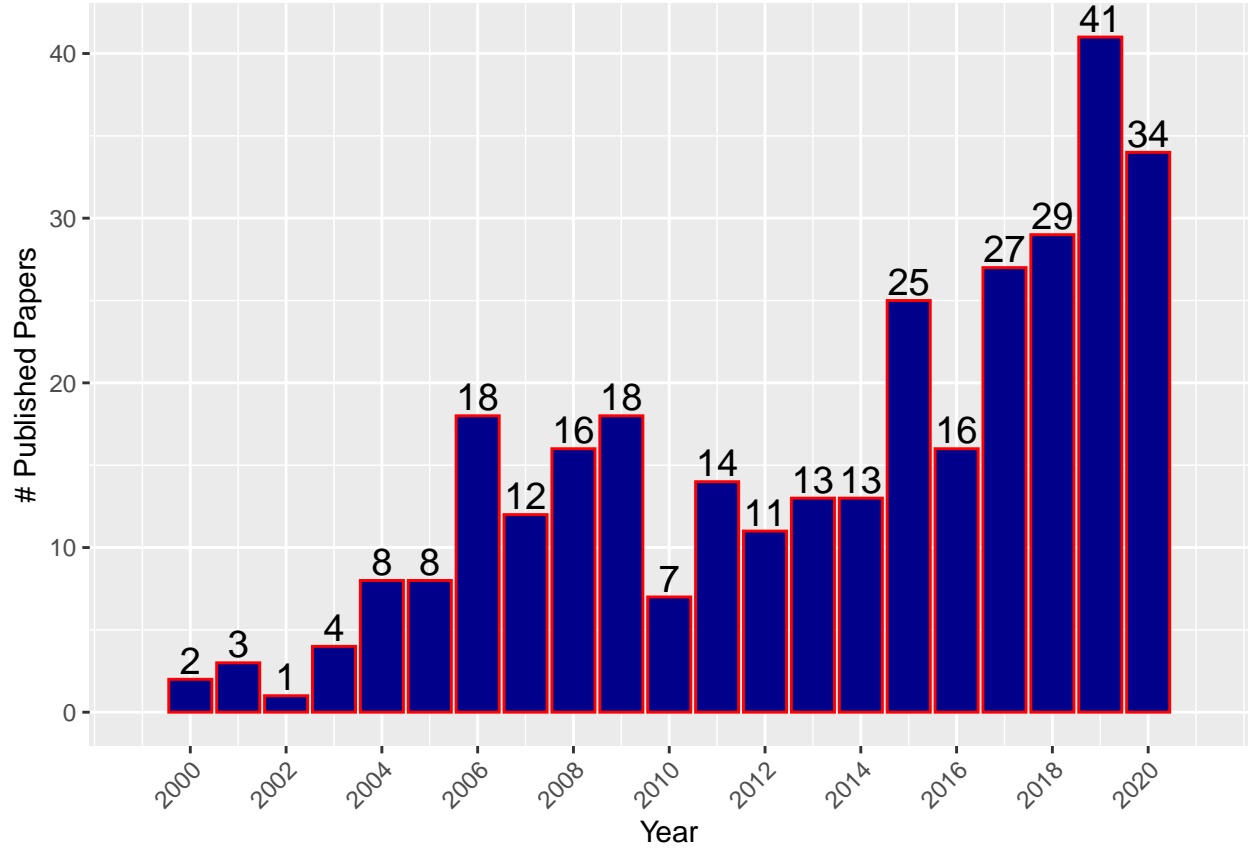


Figure 1: Counting the number of papers with the keyword in their abstract vs year

papers published in (www.onepetro.org), we count the number of the papers with “one” of following keyword in their abstract:

1. “Proxy” + ” Optimization” + “Reservoir”
2. “Surrogate” + “Optimization” + ” Reservoir”
3. “Reduced-physics model” + “Optimization” + ” Reservoir”

The time period of 1995-2020 were considered. As the 1

- small function value $f(x)$ with least search cost there are two conflicting objectives
- Global Optimiation [¹]

—>

1.2. Survey of surrogate -based papers in onepetro database

2. Problem Statement

In general, an optimization task can be defined as a search process for the maximum output value of a “well behaved”¹ objective function f . Can be defined as $f : \chi \rightarrow \mathbb{R}$ where acceptable solutions χ , has a dimension of D , $\chi \subseteq \mathbb{R}^D$:

$$\begin{aligned} & \underset{x}{\text{maximize}} && f(x) \\ & \text{subject to} && x \subseteq \chi \end{aligned} \tag{1}$$

In Figure 2 we can see some examples where the surface of f could be challenging to be optimized. The surfaces on the left side need careful attention to avoid getting stuck in local optima. Figures on the right side show presence of saddle area, where the gradient of function f is zero, in some cases in only one direction, possibly all directions. In this work, the focus is on the type of objective function f , which is challenging to optimize because of the following three difficulties:

- f is explicitly unknown. This is a typical case in reservoir optimization problems where the Net Present Value (NPV) or Recovery Factor (RF) is computed through solving a vast number of partial differential equations through flow simulation. Thus, a precise analytical expression for the objective function is not available, avoiding the applicability of techniques that exploit the analytical expression of the objective function.
- The surface of f is multi-modal. Meaning that f is non-convex in the domain of χ , and the optimization algorithm must visit all local optima to find the “global” one.
- Most importantly, forward evaluation of f is computationally expensive. This point will be discussed more in detail below.

In the examples of this paper, the goal is to maximize the subsurface-outcomes-based NPV (in USD). Thus, the primary objective function is also referred to as simply NPV in the rest of this paper. This objective function has been widely used in both well control and field development optimization studies. In a deterministic setting, the uncertainty in the geological parameters is disregarded and the optimization is performed based on a single geological model. Therefore, in the case of deterministic optimization, the objective function can be defined as:

$$J(\mathbf{u}, \mathbf{G}) = \sum_{k=1}^K \left[\sum_{j=1}^{N_p} p_{o,j,k}(\mathbf{u}, \mathbf{G}) - \sum_{j=1}^{N_p} p_{wp,j,k}(\mathbf{u}, \mathbf{G}) - \sum_{j=1}^{N_{wi}} p_{wi,j,k}(\mathbf{u}, \mathbf{G}) \right] \frac{\Delta t_k}{(1+b)^{\frac{t_k}{D}}} \tag{2}$$

¹In this context, it means the function is defined everywhere inside the input domain, it is single-valued and continuous.

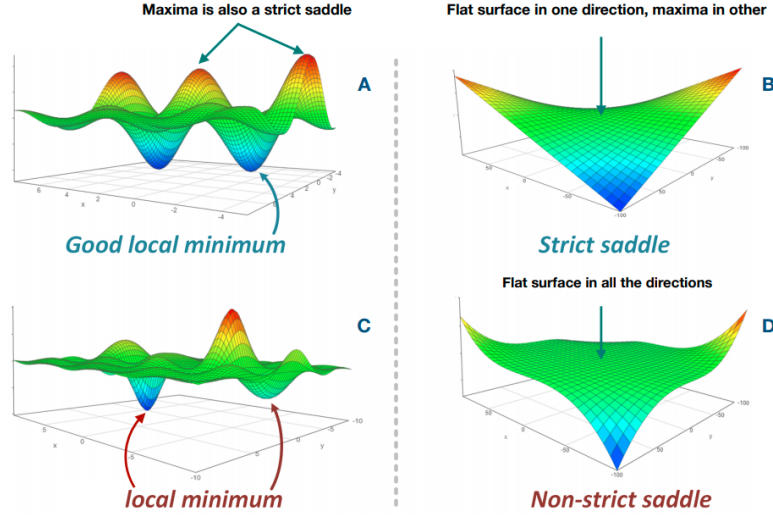


Figure 2: This plot may change, it does not show what exactly I want to say...

Where the first term in the double summation corresponds to the oil revenue; the second term is water-production cost and third term corresponds to the water-injection cost. Equation (2) is considered as objective function in the deterministic setting since only a single geological model is considered. The G in the Equation (2) is “the geological model.” The additional parameters in the Equation are as follows: K is the total number of timesteps; N_p is the total number of production wells subject to optimization; N_{wi} is the total number of water-injection wells subject to optimization; k is the timestep index; j is the well-number index; p_o is the revenue from oil production per unit volume (in USD/bbl); p_{wp} is the water-production cost per unit volume (in USD/bbl); p_{wi} is the water-injection cost per unit volume (in USD/bbl); q_o is the oil-production rate (in B/D); q_{wp} is the water-production rate (in B/D); q_{wi} is the water-injection rate (in B/D); Δt_k is the time interval for timestep k (in days); b is the discount rate (dimensionless); t_k is the cumulative time for discounting; and D is the reference time for discounting ($D = 365$ days if b is expressed as a fraction per year and the cash flow is discounted daily). \mathbf{u} in Equation (2) is the control vector (i.e., a vector of control variables) defined as $\mathbf{u} = [u_1, u_2, \dots, u_N]^D$, where D is the number of control variables (dimension of optimization problem).

As mentioned above, Equation (2) lacks to capture the uncertainty in the geological model. In contrast, in a Robust Optimization (RO) setting, the objective is to optimize the expected value over all geological realizations (assumption here is decision maker is risk-neutral). The objective function for the RO setting then can be defined as: (in the case of equiprobable geological realization)

$$\bar{J}(\mathbf{u}) = \frac{\sum_{re=1}^{n_e} J(\mathbf{u}, \mathbf{G}_{re})}{n_e} \quad (3)$$

Where in Equation (3) contrary to Equation (2), there is not one, rather n_e geological realizations, each of them written as G_{re} . In this work, the objective is to optimize the Equation (3), where it is simply EV value of NPV defined in (2) over all realizations.

It is well defined in the literature that optimizing Equation (3) is computationally prohibitive (de Brito and Durlofsky, 2021a; Hong et al., 2017a; Nwachukwu et al., 2018b). Not only because thousand(s) of PDE have to be solved in the flow-simulation in order to compute the q_o, q_{wp}, q_{wi} ; the flow simulation must be enumerated over all realizations n_e to compute $\bar{J}(u)$. Let's assume a simple case to illustrate the computational burden of this optimization problem. Assume that an E&P enterprise is in the process of finding the bottom hole pressure of five injection wells and shut-in time of other five production wells, $D = 10$. The geology team of the enterprise comes up with 100 geological realizations of the model. ($n_e = 100$). Now, if we suppose that the reservoir model is 3D with a moderate number of grid cells, it is not hard to imagine that flow-simulation of a fine grid model will take ~ 1 hr. Then, simply having 100 realizations means that each forward computation of $\bar{J}(u)$ takes around ~ 100 hr. Considering that the enterprise has to decide in 6 month period (in the best case, it can be interpreted as 6 months CPU running time), which means that a total number of the available budget for running the forward model is $\frac{6 \times 30 \times 24}{100} = 43.2 \approx 50$ is around 50. The budget of only 50 forward model in ten-dimensional, non-linear, and non-convex optimization problem is relatively low. To put this in simple terms, if we say that each dimension of the control variable \mathbf{u} , could be discretized into ten possible cases, then total available solutions for this optimization problem will be Number of all possible solutions = 10^{10} . As it is clear, finding the best solution from a pool of ten billion possible solutions with only 50 shots is a pretty much hard undertaking.

The rest of this paper will be arguing that the Bayesian Optimization workflow has especial strength to deal with the three difficulties described above. Where the workflow needs to capture the optimum global point (area) while having a small forward evaluation budget.

3. Bayesian Optimization Workflow

3.1. Overall View

Workflow to perform global optimization of multimodal black-box functions:

- Step 1. Choose some initial design points and build a probabilistic model over the space of possible objective f , this probabilistic model serves as prior.
- Step 2. Combine prior and the likelihood to get a posterior of probabilistic model over the objective given some observations.
- Step 3. Use the posterior to decide where to take the next evaluation \mathbf{x}^* according to some policy for decision making.
- Step 4. Evaluate the f at \mathbf{x}^* and augment it to the initial data, in step 1.

Iterate between 2 and 4 until the evaluation budget is over. ## 3.1 Gaussian Process

3.1.1. Step 1. Probabilistic Model as Prior

3.1.1.1. Gaussian Process (GP). reference to the book (Murphy, 2022)

Key Assumption in (GP) is that: the function values at a set of $M > 0$ inputs, $\mathbf{f} = [\mathbf{f}(\mathbf{x}_1), \dots, \mathbf{f}(\mathbf{x}_M)]$, is jointly Gaussian, with mean and Covariance

$$(\mu = m(x_1), \dots, m(x_M)) \sum_{i,j} = \kappa(x_i, x_j) \quad (4)$$

and κ is a positive definite (Mercer) kernel. Suppose we have a initial design points, set $\mathcal{D} = (x_n, y_n) : n = 1 : N$, where $y_n = f(x_n)$ is the noise-free observation of the function evaluated at x .

Now we consider the case of predicting the outputs for new inputs that may not be in \mathcal{D} .

$$\mathbf{f}_* = [\mathbf{f}(\mathbf{x}_1), \dots, \mathbf{f}(\mathbf{x}_{N_*})] \quad (5)$$

By definition of the GP, the joint distribution $p(\mathbf{f}_X, \mathbf{f} | \mathbf{X}, \mathbf{X}_*)$ has the following form:

$$\begin{bmatrix} \mathbf{f}_X \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mu_X \\ \mu_* \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{X,X} & \mathbf{K}_{X,*} \\ \mathbf{K}_{X,*}^\top & \mathbf{K}_{*,*} \end{bmatrix} \right) \quad (6)$$

$$\begin{aligned} \mu_X &= [m(x_1), \dots, m(x_N)] \\ \mu_* &= [m(x_1^*), \dots, m(x_N^*)] \end{aligned} \quad (7)$$

$$\begin{aligned} K_{X,X} &= \kappa(X, X; \theta), & size(N \times N) \\ K_{X,*} &= \kappa(X, X_*; \theta), & size(N \times N_*) \\ K_{*,*} &= \kappa(X_*, X_*; \theta), & size(N_* \times N_*) \end{aligned} \quad (8)$$

Covariance Kernels	assumeing $h = x - x' $
Gaussain	$\kappa(x, x') = \sigma_f^2 \exp(-\frac{h^2}{2\ell^2})$
Matern $\mu = \frac{5}{2}$	$\kappa(x, x') = \sigma_f^2 (1 + \frac{\sqrt{5} h }{\ell} \frac{5h^2}{3\ell^2}) \exp(-\frac{\sqrt{5} h }{\ell})$
Matern $\mu = \frac{3}{2}$	$\kappa(x, x') = \sigma_f^2 (1 + \frac{\sqrt{3} h }{\ell}) \exp(-\frac{\sqrt{3} h }{\ell})$
Exponetial	$\kappa(x, x') = \sigma_f^2 \exp(-\frac{ h }{\ell})$
Power-Exponetial	$\kappa(x, x') = \sigma_f^2 \exp(-(\frac{ h }{\ell})^p)$

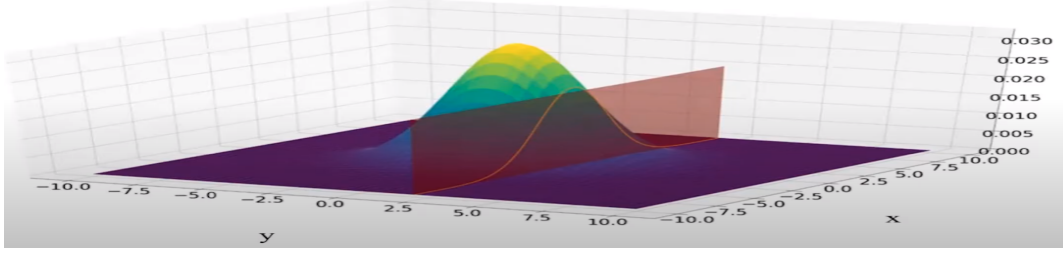


Figure 3: proof of

$$\kappa(x, x'; ; \theta) = (1 + \frac{\sqrt{5}|h|}{\theta} + \frac{5h^2}{3\theta^2}) \exp(-\frac{\sqrt{5}|h|}{\theta}) \quad (9)$$

3.1.1.2. Covariance Kernel, Parameter estimation.

$$p(y|\mathbf{X}, \theta) = \int \mathbf{p}(\mathbf{y}|\mathbf{f}, \mathbf{X}) \mathbf{p}(\mathbf{f}|\mathbf{X}, \theta) \quad (10)$$

$$\log p(y|\mathbf{X}, \theta) = \mathcal{L}(\zeta, \sigma_f^2) = -\frac{1}{2}(\mathbf{y} - \mu_{\mathbf{X}})^T \mathbf{K}_{\mathbf{X}, \mathbf{X}}^{-1} (\mathbf{y} - \mu_{\mathbf{X}}) - \frac{1}{2} \log |\mathbf{K}_{\mathbf{X}, \mathbf{X}}| - \frac{n}{2} \log(2\pi) \quad (11)$$

168 Where the dependence of the $\mathbf{K}_{\mathbf{X}, \mathbf{X}}$ on θ is implicit. The gradient-based optimizer is performed in order to:

$$[\zeta^*, \sigma_f^{2*}] = \operatorname{argmax} \mathcal{L}(\zeta, \sigma_f^2) \quad (12)$$

169 However, since the objective \mathcal{L} is not convex, local minima can be a problem, so we may need to use
170 multiple restarts.

171 3.1.2. Step 2. Posterior of Probabilistic Model

172 3.1.2.1. Posterior of Gaussain Process, (conditioning on initial data). Here in 3

$$p(f_*|X_*, \mathcal{D}) = \mathcal{N}(f_*|\mu_*, \sum_*) \quad (13)$$

$$\begin{aligned} \mu_* &= m(\mathbf{X}_*) + \mathbf{K}_{\mathbf{X},*}^T \mathbf{K}_{\mathbf{X},\mathbf{X}}^{-1} (\mathbf{f}_{\mathbf{X}} - \mathbf{m}(\mathbf{X})) \\ \sum_* &= \mathbf{K}_{*,*} - \mathbf{K}_{\mathbf{X},*}^T \mathbf{K}_{\mathbf{X},\mathbf{X}}^{-1} \mathbf{K}_{\mathbf{X},*} \end{aligned} \quad (14)$$

173 3.1.3. Example of Step.1 and Step.2

174 Assume $X = [0, 3, 5, 6]$ and $f_X = \sin(X)$, giving $\mathcal{D} = (X, f_X)$. What is $p(f_*|X_*, \mathcal{D})$

Gaussian Process Regression

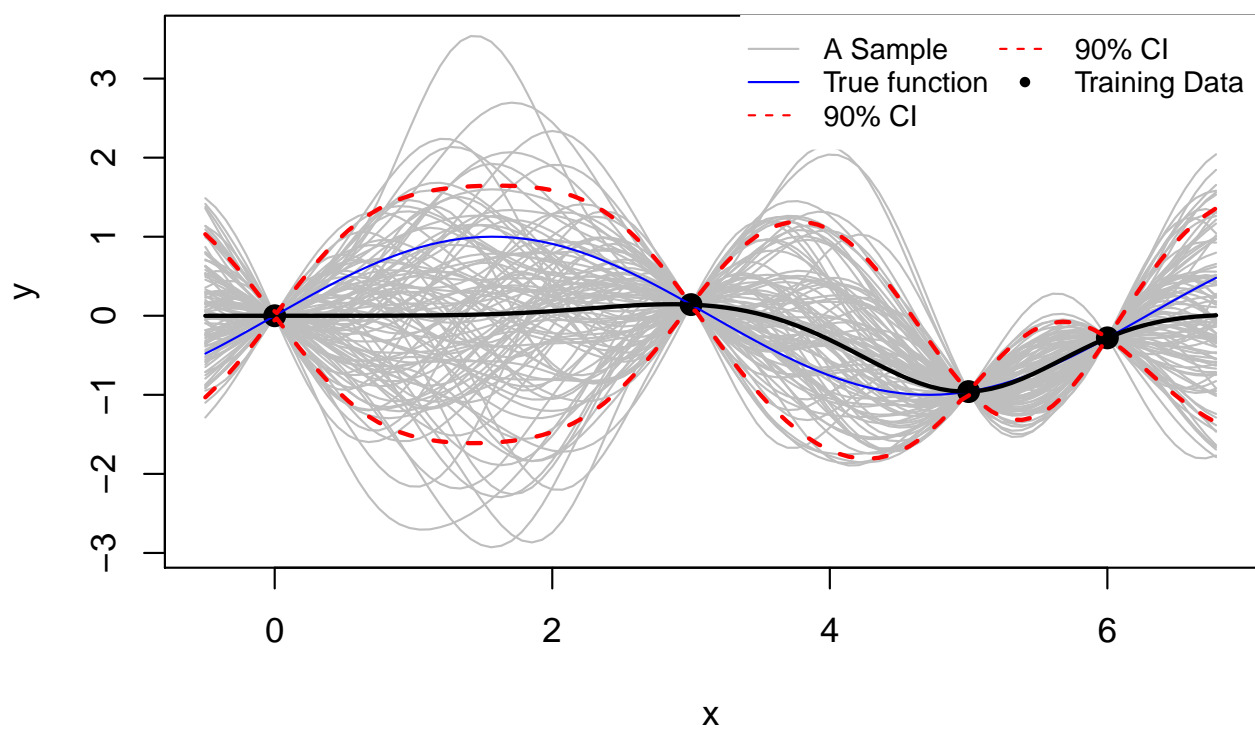


Figure 4: Gaussian Process Regression conditioned on 4 points

Gaussian Process Regression

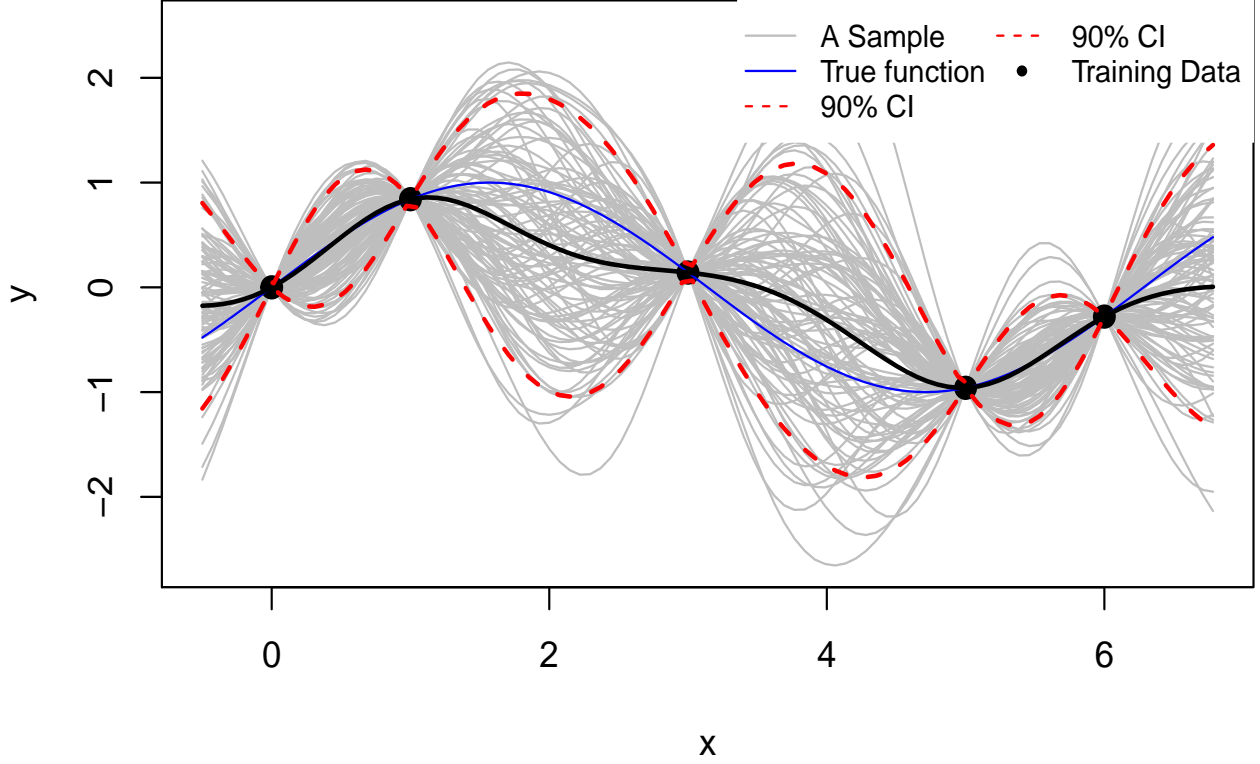


Figure 5: Gaussin Process Regression conditioned on 5 points

Now we sample the point $X = 1$, and add to \mathcal{D}

3.1.4. Step.3 Deciding on next \mathbf{x}^* based on Posterior

Posterior of the probalistic model quantify the uncertainty over the space of the f . The question is what is the next \mathbf{x}^* to be sampled from the *expensive function*?

Define an utility function to collect new data points satisfying some optimality criterion: optimization as decision making.

There are a few of policies in the literature of Bayesopt, here the *Expected Improvement (EI)* policy will be used.

3.1.4.1. Expected Improvement as Policy for Decision Making. In Expected Improvement (EI) policy choose the next query point as the one which has the highest expected improvement over the space of the *expensive function*

$$utility(x; \theta, \mathcal{D}) = \alpha_{EI} = \int_y \max(0, y - f) p(y|x; \theta, \mathcal{D}) \quad (15)$$

$$utility(x; \theta, \mathcal{D}) = \alpha_{EI} = \int_y \max(0, y - f) p(y|x; \theta, \mathcal{D}) dy$$

186 However, we do not have access to the *expensive function*, f , therefore we replace the f with the best
 187 available solution found so far, y^+

$$utility(x; \theta, \mathcal{D}) = \alpha_{EI} = \int_y \max(0, y - y^\dagger) p(y|x; \theta, \mathcal{D}) dy \quad (16)$$

188 y^+ : The best solution found in the training dataset \mathcal{D}

189 The good news: The analytical form of the utility function is available for the gaussian process

$$\gamma(\mathbf{x}) = \frac{\mu(\mathbf{x}; \theta, \mathcal{D}) - y^\dagger}{\sigma(\mathbf{x}; \theta, \mathcal{D})} \quad (17)$$

$$utility(\mathbf{x}; \theta, \mathcal{D}) = \alpha_{EI}(x; \theta, \mathcal{D}) = (\mu(x; \theta, \mathcal{D}) - y^\dagger) \Phi(\gamma(x)) + \sigma(x; \theta, \mathcal{D}) \phi(\gamma(x)) \quad (18)$$

190 Where $\Phi(\cdot)$ and $\phi(\cdot)$ are CDF and PDF of standard Gaussian distribution.

191 It is too greedy in the context of the sequential decision making. Therefore, an explorative term is added
 192 as explorative” parameter ϵ .

$$\gamma(\mathbf{x}) = \frac{\mu(\mathbf{x}; \theta, \mathcal{D}) - y^\dagger - \epsilon}{\sigma(\mathbf{x}; \theta, \mathcal{D})} \quad (19)$$

$$\alpha_{EI}(x; \theta, \mathcal{D}) = (\mu(x; \theta, \mathcal{D}) - y^\dagger - \epsilon) \Phi(\gamma(x)) + \sigma(x; \theta, \mathcal{D}) \phi(\gamma(x)) \quad (20)$$

193 3.1.4.2. BO As a “mapping” between two problems. BO is an strategy to transform the problem

$$u_M = \underset{u \in \text{constraints}}{\operatorname{argmax}} \bar{J}(u) \quad (21)$$

194 unsolvabile!

$$u^{next} = \underset{u \in \text{constraints}}{\operatorname{argmax}} \alpha_{EI}(u; \mathcal{D}_n, \theta^*) \quad (22)$$

195 solvabile!

- 196 • $\alpha_{EI}(u)$ is inexpensive to evaluate.
- 197 • The analytical expression for gradient of $\alpha_{EI}(u)$ is available.

- Still need to find u^{next} , the multi-start BFGS is used for finding u^{next} .

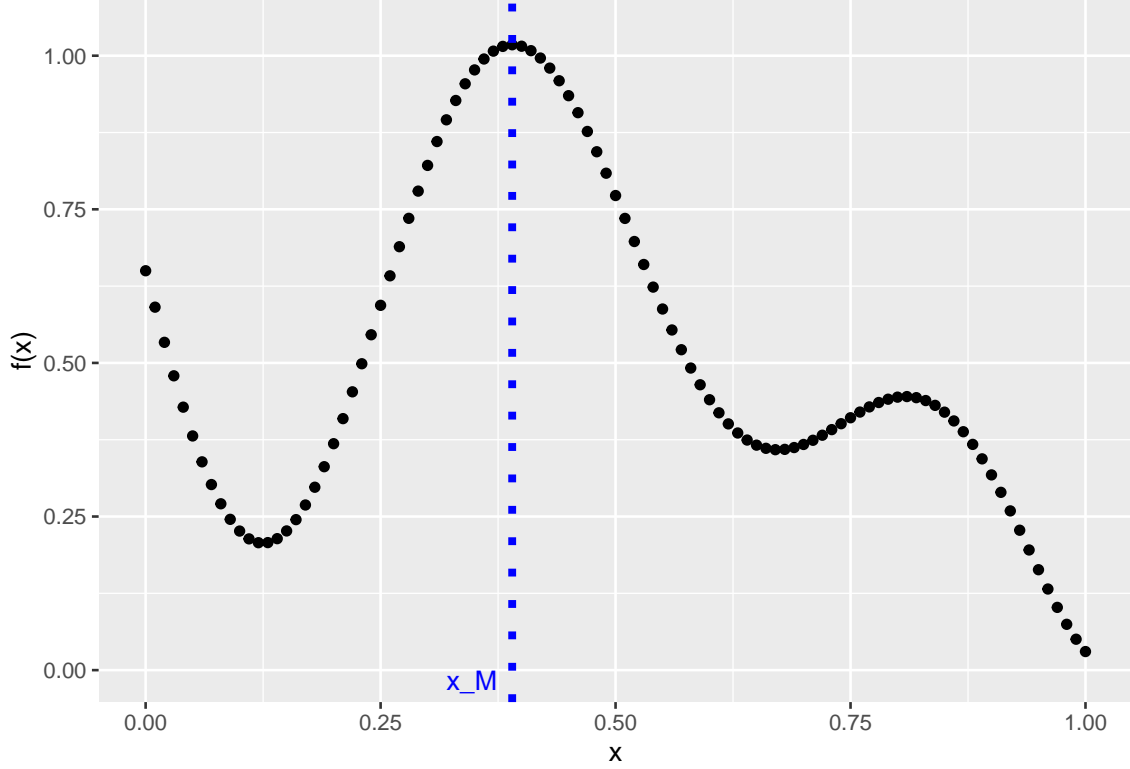


Figure 6: Plot of 1-D equation with blue dash line representing the global optimum

4. Example Cases

4.1. 1-D toy Problem

In this section, a 1-D toy problem is considered to illustrate the Bayes Optimization workflow discussed in the previous section. The 1-D problem was selected since it will help to visualize all the steps of the workflow making easier explanation of the concepts. Though, it can be seen from the next section, the workflow can easily be extended to higher dimensional problems. The *True function* to be optimized in this section has an analytical expression as, given the box constraints:

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & f(x) = 1 - \frac{1}{2} \left(\frac{\sin(12x)}{1+x} + 2 \cos(7x)x^5 + 0.7 \right) \\ \text{subject to} \quad & 0 \leq x \leq 1 \end{aligned} \tag{23}$$

Since the analytical expression of the function is available and being a 1-D problem, the global optimum of the function has been found at $x_M = 0.39$. The plot of the function and the optimum point has been shown in Figure 6.

However, it is worth to mention that the analytical expression of the objective function in many of real-world problems is not available, what is available is a *samples* from the objective function. Therefore, in the coming

example a few samples are sequentially drawn from the objective function to resemble the real case scenario. However, we know the global optimum of the objective function in hindsight, just in the case we want to compare the performance of Bayesian optimisation algorithm.

Therefore, as Figure 7, the 5 sample points, $x = [0.05, 0.2, 0.5, 0.6, 0.95]$ were selected as the initialization of the workflow. In the upper plot, blue lines represent the samples from the posterior of the gaussian model conditioned on the five sample points. The grey area represents the 95% confidence interval while the red curve represents the mean value of the samples (blue lines). The first point to infer from the Figure 7 is there is no uncertainty on the sample point. As shown, there is no grey zone on sample point since as was discussed in the previous section, here we consider the “noise-free” observation. Also, worth to mention that we have wide more uncertainty (wider grey band) in the areas that are more distant from the observation, simply meaning we are less uncertain close to observation points. On the “extrapolation,” meaning in the areas outside of the observation points, the probabilistic model shows interesting behaviour. On those “extreme” area, the mean curve tends to move toward the mean of all observation points, here around 0, showing the model reflects the mean-reversion behaviour when it comes to extrapolation.

The lower part of Figure 7, shows the plot of utility function at each x values. Worth to note that as the plot suggests, the utility(α_{EI}) function will have the multi-modal structure, meaning in the optimization process multi-start gradient method will be helpful, in order to avoid stuck in the local optima. In this work, as was explained in the previous section, the multi-start gradient method was used. The blue dotted line shows the x_{next} which is the point where the utility function is maximum. Then this x_{next} is queried from the real function, and the pair of $(x_{next}, f(x_{next}))$ is added to the initial data set, \mathcal{D} . Going back to the lower figure at Figure 7, the utility has two modes around point $x = 0.5$, say $x_{0.5}^+$ and $x_{0.5}^-$, however the point $x_{0.5}^-$ is selected as the next query point. Readers can be referred to the upper plot and it is clear that there is more uncertainty around point $x_{0.5}^-$ than $x_{0.5}^+$ which given the form of utility function, that is understandable. The utility function always looks for the point that not only maximizes the mean value, but also is interested in the points that have higher variance, which is the case between two points $x_{0.5}^+$ and $x_{0.5}^-$.

Calling the Figure 7 as the iteration # 1, now we can start sampling sequentially. In the Figure 8 another two iterations have been provided. Where in each row, the plot on the left represents the posterior on the gaussian conditioning, the right shows the utility function. Note that in the Figure 8 all axis labels and legends were not included, to have better visibility. (more info about each plot can be found in 7). Interesting to see that in this example case, at the iteration #2, the workflow queries the point $x = 0.385$ which presents the best point so far found through BayesOpt workflow. Therefore, after just two iterations we are around $\frac{x_{best}}{x_M} = \frac{0.385}{0.390} = 98.7$ of the global optima. Although this is the case for 1-D problem, it is clearly showing the strength of the workflow to approach the global optima, in as few as possible iterations. In this case after iteration #2, the total

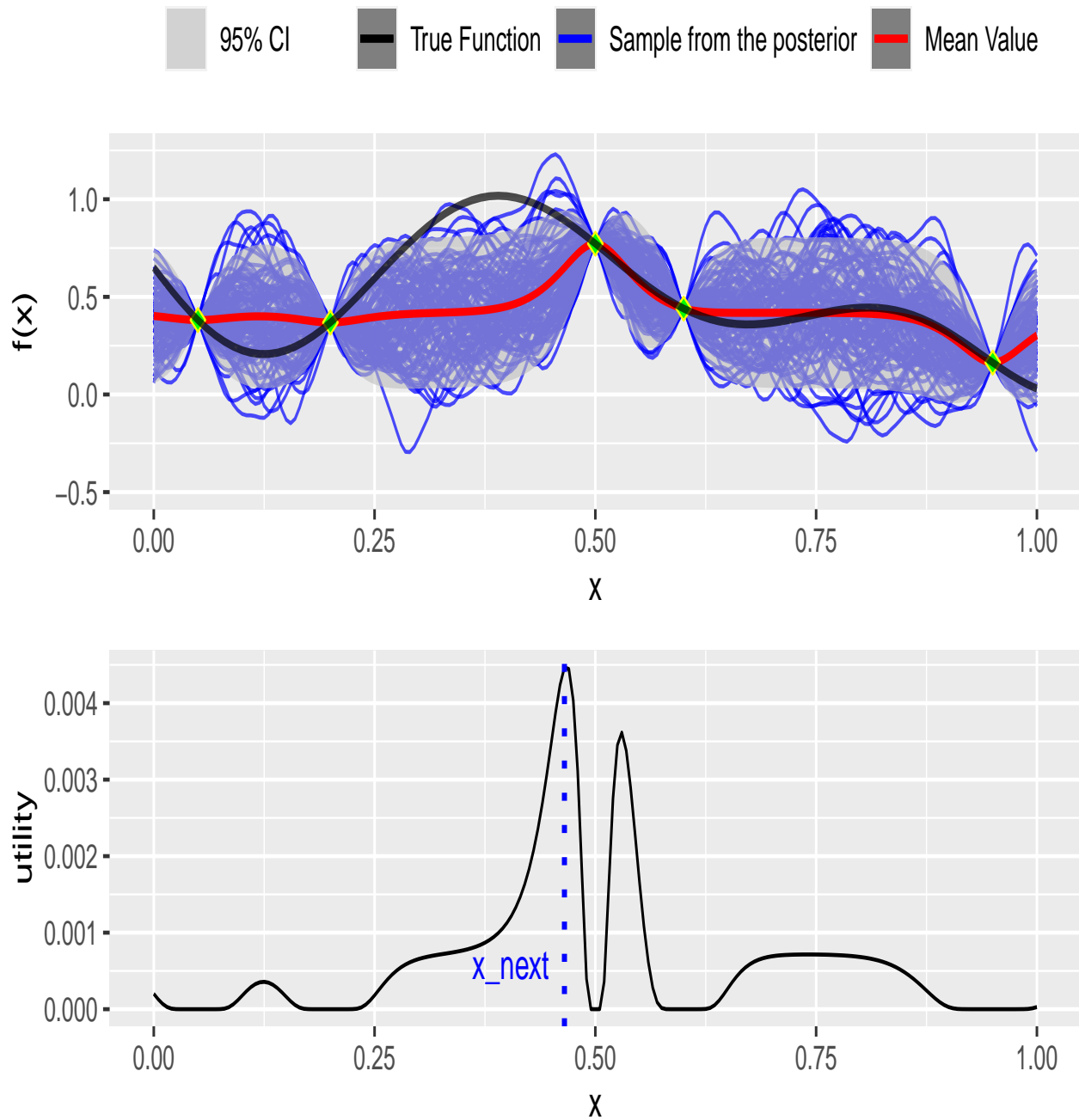


Figure 7: Itel - Top: Gaussian posterior over the initial sample points; Lower: Utility function over the x values

244 number of time that the real function has been queried is $\text{size}(\mathcal{D}) + \text{size}(\text{totaliteration}) = 5 + 2 = 7$.

245 Before going to apply the same workflow at the field scale, the 1-D example presented here offer another
246 useful feature of the Bayesian Optimisation. Looking at 8, we can see that the maximum of utility function is
247 at the iteration # 3 in order of 10^{-6} . That show that after optimization, eve best point to be queried in the
248 next section has a very little utility. So can safely stop the process, since querying points to be sampled from
249 the expensive function has a negligible potential to improve our search in optimization.

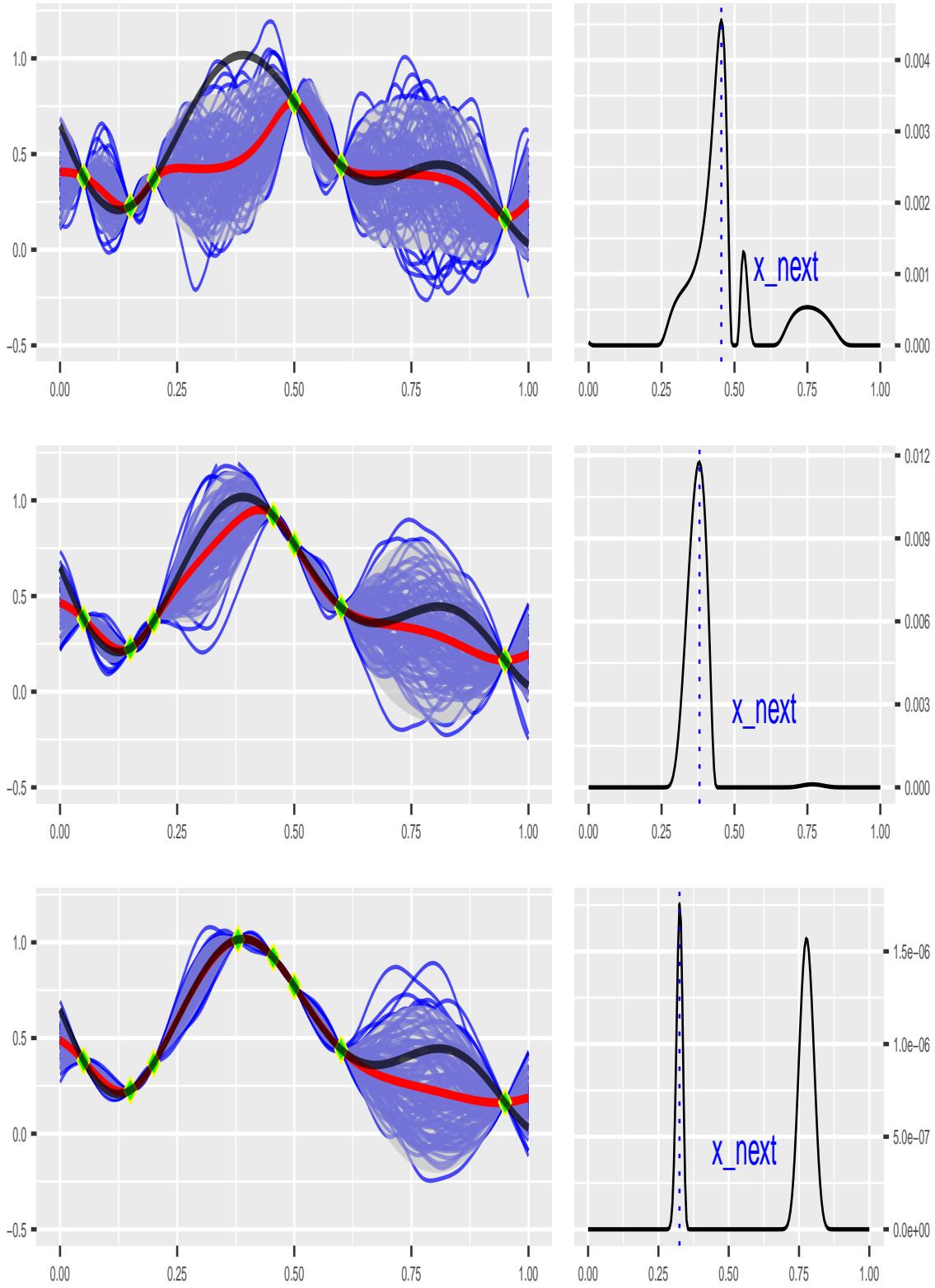


Figure 8: Gaussian posterior of over the initial sample points

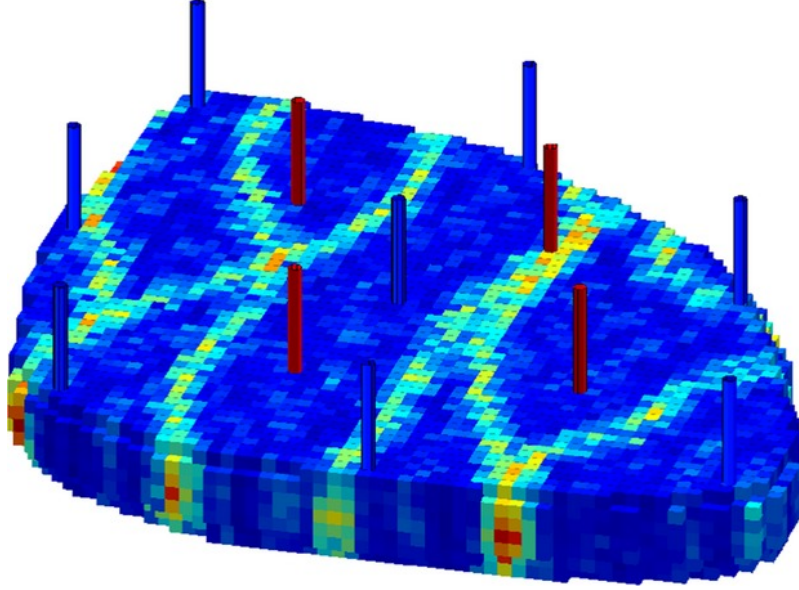


Figure 9: Well locations in Egg model, blue ones are injection, the red producers

4.2. Field Scale

4.2.1. Synthetic 3D Reservoir Model

In this section, the BayesOpt workflow is applied to the synthetic 3D reservoir model. The introduction of the model and geological description can be found in (Jansen et al., 2014). Known as “Egg Model” it has a geology of channelized depositional system.

The 3D model has eight water injectors and four producers wells shown in Figure 9. The model has geological realizations of patterns of highly permeable channels which are described by 100 equi-probable geological realizations, three of which are illustrated in left side of Figure 10. (Hong et al., 2017b).

Relative permeabilities and the associated fractional flow curve of the model have shown in right side of Figure 10. All the wells are vertical and completed in all seven layers. Capillary pressure is ignored. The reservoir rock is assumed to be incompressible. The model has a life-cycle of 10 years. Here, the injection rate to be maintained over life-cycle of reservoir is going to be optimized. Thus, given eight injection wells, the optimization workflow has the eight dimensions. However, the optimization is not unbounded, the water can be adjusted from 0 to 100 m³/day, making the box-constrained optimization. The injectors are operated with no pressure constraint, and the producers are under a minimal BHP of 395 bars without rate constraint.

4.2.2. Well Control Optimization

Reviewing the equation raised in the section 3, here the goal is robust optimization of the field, given geological realizations as follow:

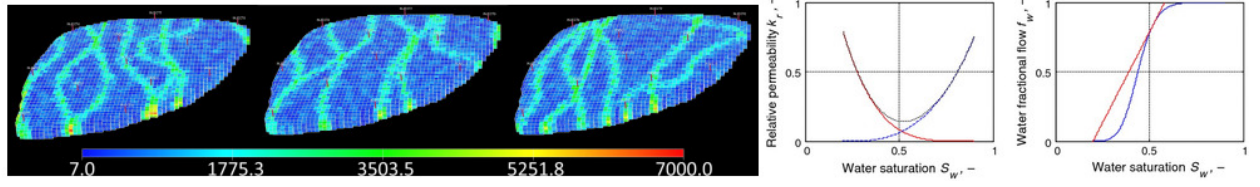


Figure 10: Left: Three geological realizations of the 3D model; Right: Rel perm and fractional flow curve

$$\text{Objective Func}(u) = \bar{J}(u) = \frac{\sum_{i=1}^{n_e} J_r(u, G_i)}{n_e} \quad (24)$$

Equation (24)

u is Injection rate for the each injection well, therefore the control vector, to be optimizaed in this case is defined as:

$$u = [u_{inj1}, u_{inj2}, u_{inj3}, u_{inj4}, u_{inj5}, u_{inj6}, u_{inj7}, u_{inj8}]^T \quad (25)$$

As the (24) suggest, the $\bar{J}(u)$ need some parameters to be defined. The oil price (P_o), water production cost (p_{wp}) and water injection cost (P_{wi}) in *dollar/m³* has been provided in the Table 1. Also, in this work the cash flow is disconted daily and the discount factor is available in the 1. We would like to note that in this work due to avoid further computational burden in optimization process, 10 realizations of the egg model has been considered, therefore $n_e = 10$ in Equation (24).

Table 1: Required Parameters needed for calculation of Expected NPV

Item	Pric	Items	Value
P_o	315	b	8%
P_wp	47.5	D	365
P_wi	12.5	n_e	10

4.2.3. BayesOpt Workflow

As it was discussed, the starting point of the BAYesOpt workflow is to randomly sample the initial data pairs \mathcal{D} which is used to build the Gaussian model of the response surface to the input variables. In this work, forty samples fom the Latin hyper cube sampling (LHS) method were drawn. The LHS is prefred in this work to Monte Carlo since it provides the stratification of the CDF of each variable, leading to better coverage of the input variable space. The Figure 11 show the results of the $\bar{J}(u)$ for each sample from LHS. Also, The maximum $\bar{J}(u)$ found from sampling has been shown with blue line. Setting the specific seed number (since LHS is in itself is random process), we get the max *NPV* aciehved here was 35.65\$MM. Looking at Figure

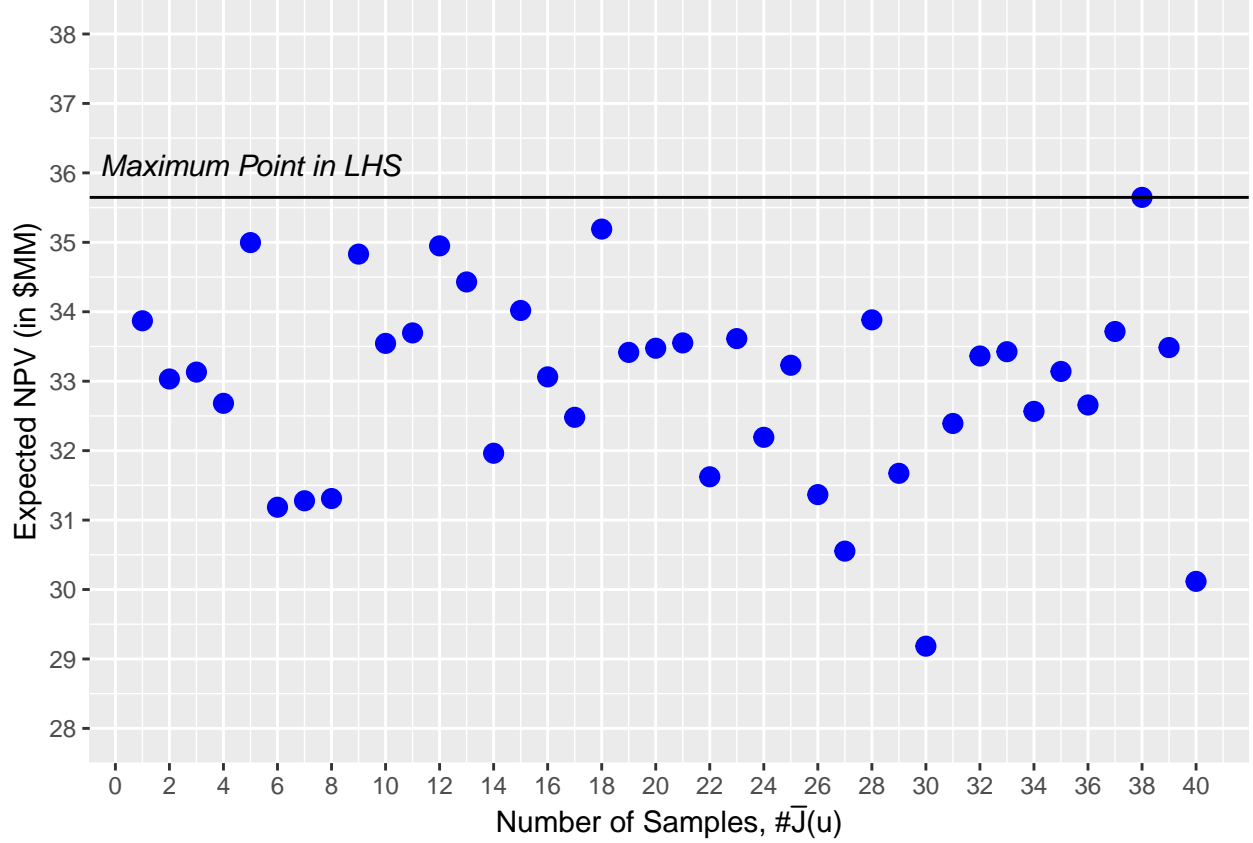


Figure 11: Expected NPV as result of forty sampling from LHS

11 it is worth to mention that random sampling like the LHS is not helpful to consistently approach the
 global optimum point, and there is a need for efficient workflow to find the optimum point while using the a
 few as possible sampling from real function.

Having the initial data found through LHS, we can build the probalistic model of the reposnse surface
 and sequentially sample from the *expensive-to-evaluate* function. Unfortunately, win this section we can not
 plot the posterior of the probalistic model, condition on the above forty LHS samples, due being the space is
 eight-dimentional, and hard to visulize. The Figure 12 shows the expected NPV found after ten sequential
 sampling resulted from the BayesOpt workflow. Readers are refreed to this point that in the figure, not all
 red points are increasing and some points are lower than previous points. The reason for this behaviour is
 the nature of BayesOpt algorithm. We can suggest that in the points that has lower expected NPV from the
 previous, we may reached the lower optimum point, but those points helped us to decrease the uncertainty,
 which is helpful for the further sampling. We can see that after just ten evaluation of the expenside function
 (here it means finding the expected NPv from running 10 geological realization using flow simulation) we
 reach the new record Expeted NPV of $\max \bar{J}(u) = 36.85\$MM$.

Now, as we explained in the 1-D section, the plot of the utility at each iteration could provide some useful

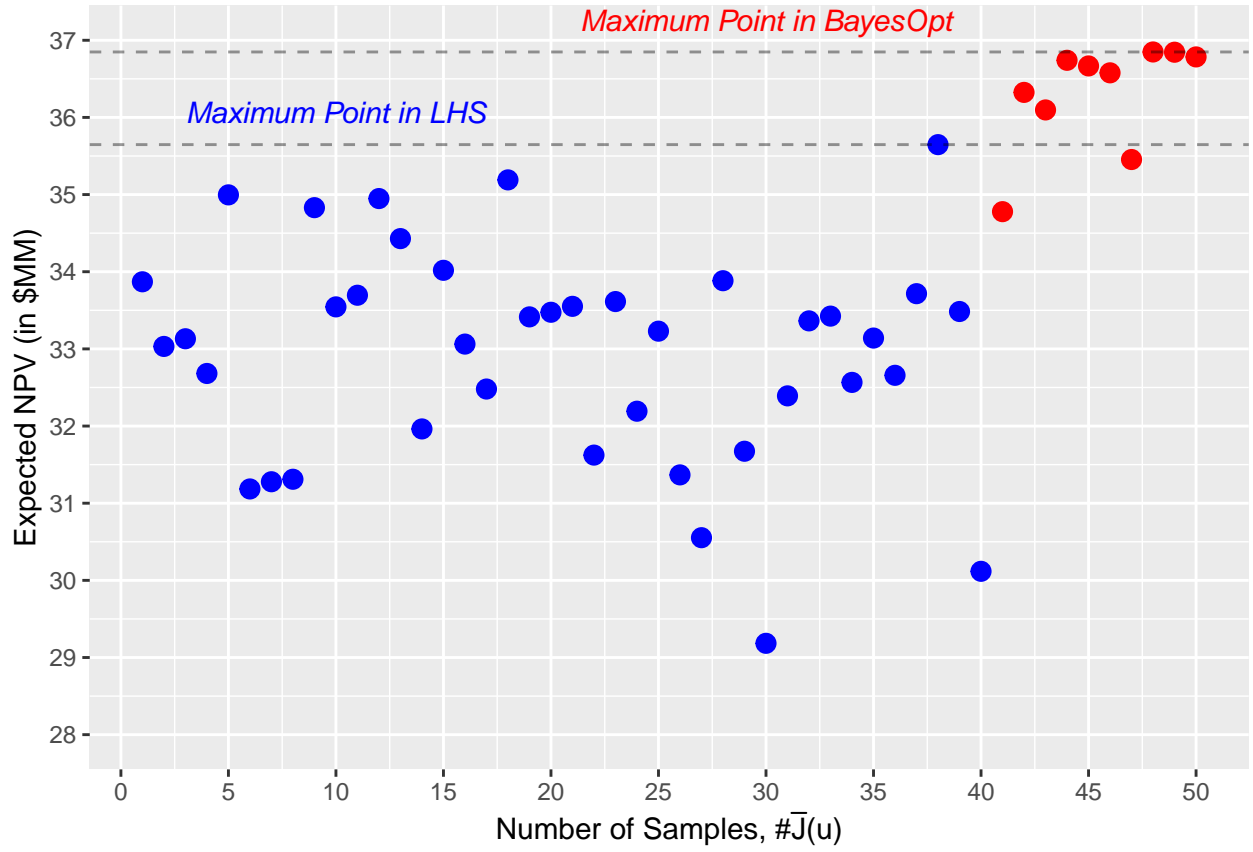


Figure 12: Blue points represents the sample from LHS, red points represents the samples from the BayesOpt Workflow

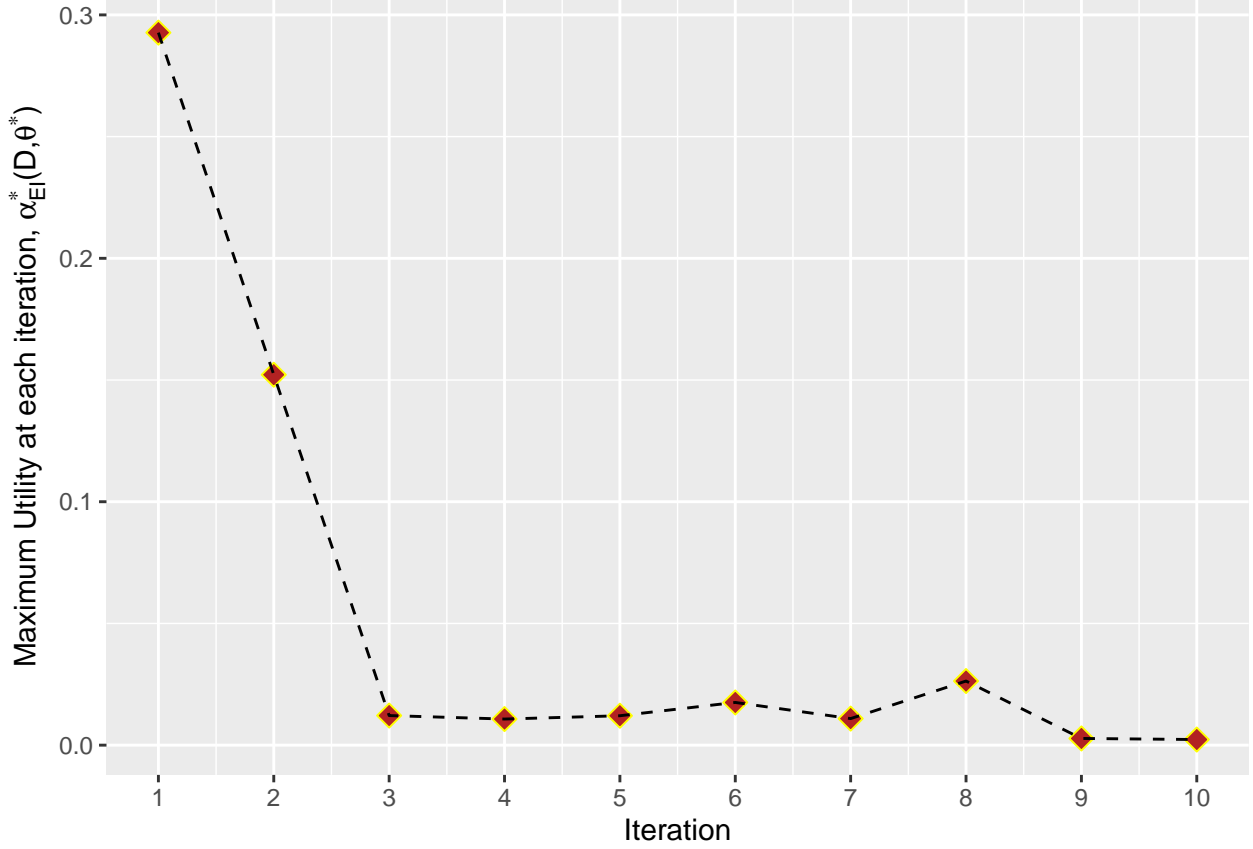


Figure 13: Maximum utility at each iteration, after running L-BFGS-B to find the u with max utility, α_{EI}^*

information about the optimization process. The Figure 13 plots the $\alpha_{EI}^*(\mathcal{D}, \theta^*)$ (Equation (22)) versus the ten iteration in this work. In fact the notation α_{EI}^* means the optimum of the $\alpha_{EI}(u; \mathcal{D}, \theta^*)$ after running multi-start (1000)- L-BFGS-B on all u values. Now, we can see that in the figure the α_{EI}^* is decreasing going toward the zero. It can be inferred from this trend that, we are going out of the *good* u values to be sampled from the expensive function, can be interpreted that we are in the vicinity of global optima, if we see after several iteration still α_{EI}^* is less than 10^{-6} .

Given that the BayesOpt inherently has stochastic nature (from this perspective that having different initialization in LHS sampling will affect the final solution), in this section BayesOpt is repeated with different initialization. Ideally, this repetition should be conducted 100 or 1000 times, to get better overview of the convergence of the algorithm given different initialization. Though, because of the computational burden, in this work only three repetitions were performed. Repeat the Optimization, three times, in different initial design points. Figure 14 shows results of three repetitions. At each repetition (top, middle, bottom), the blue dots come from different seed numbers and they are different. Then, given that initialization \mathcal{D} , sequential sampling from the expensive function is performed, shown in the red points. Like previous case,

313 in these repetitions, 40 samples drawn from LHS algorithm, the 10 were taken thorough BayesOpt algorithm,
 314 totaling 50 samples. At each row of the Figure 14, two horizontal lines show the maximum point \overline{NPV}
 315 in both random sampling phase (LHS) and BayesOpt phase. As it can be noted from the Figure 14, at
 316 each repetition, the BayesOpt will improve the solution with small sample evaluation of the $\overline{J}(u)$. Therefore,
 317 improvement following the BayesOpt phase independent of the initial design, yet the bigger question is whether
 318 given different initial design, the algorithm converge the vicinity of global optima. What is referred here is that
 319 if having different initialization will lead completely different final solution, that hints that the algorithm has
 320 a “local” search, in contrast, if the solution leads to one specific close u^* , that represents that algorithm have a
 321 “global” view on the surface of the objective function. In the case of “global” optimization having different
 322 initialization should lead to similar final solution, since the algorithm will not get stuck in local optimum points,
 323 close to initialized data. This is common practice in the gradient-based optimization where the algorithm is
 324 powerful in local optimization and in order to avoid stuck in local extreme points, “multi-start” runs are
 325 performed in order to search the global point in the objective function.

326 To further continue this discussion on the effect of initialization on the final solution, the u^* value for each
 327 repetition has been shown on the left side of Figure 15. Where the u^* is the vector of 8 dimension, each value
 328 shows the optimum injection rate for the 10 years life cycle of the field, in m^3/D . We would like to note that
 329 the y axis was plotted from the range of 5 to 100. The reason for this is to show that in this optimization
 330 problem, injection of each wells can take any number between $5m^3/D$ to $100 m^3/D$, and the y axis shows
 331 the full extent of the value optimum zation workflow can reach. Visually, looking at the left plot at Figure
 332 15, we can see that the final solution of three repetitions at each wells, does not differ significantly from each
 333 other. With small exception of (injection #2), it seems all the final solutions converge to the same solution.
 334 This feature that can be loosely said as “robustness” of optimization workflow to initial design is very helpful,
 335 from this sense that we do not need to restart the optimization with different initialization, since they all
 336 will converge to the similar solution. From this perspective, authors can infer that BayesOpt workflow
 337 can be considered as “global” optimization method, as it shows the workflow avoids stuck in local extreme
 338 points or saddle regions. The plot on the left side of Figure 15 shows that mean injection rate (mean of three
 339 repetitions) and error bar at each injection wells. The bottom of error bar in this plot shows the $mean - sd$
 340 and top of bar is $mean + sd$. As we can see that we do not see significant variation in the final solution in
 341 each repetitions, also the plots recommends that in the case of repeating the optimization with more than three
 342 times (like 10 or 100), it can lead to lower variation in final solution.

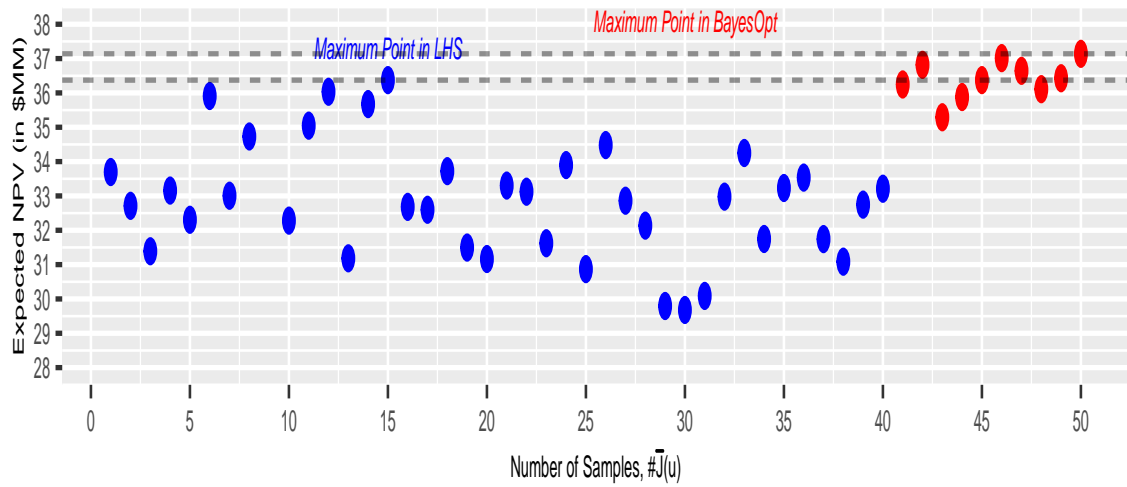
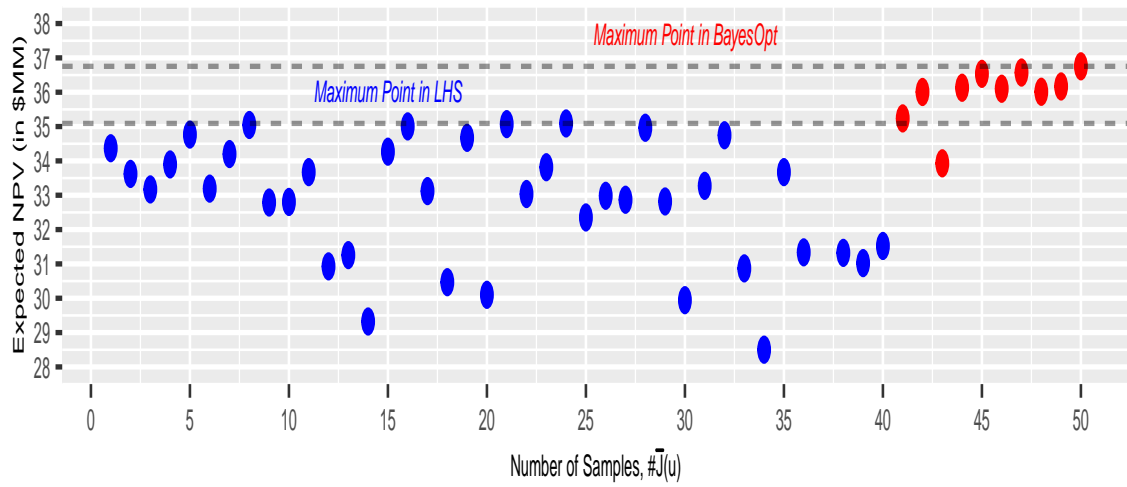
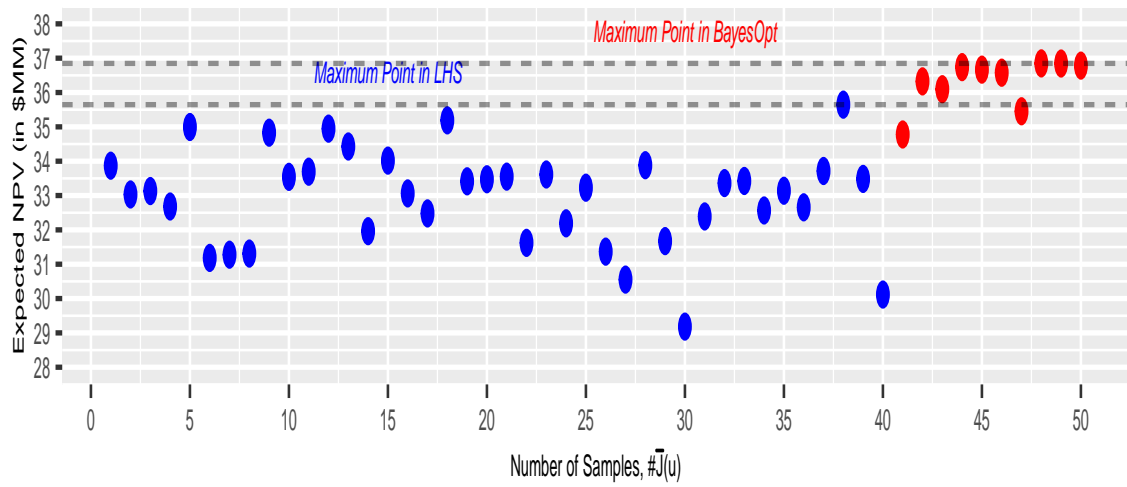


Figure 14: BayesOpt workflow applied to Syntetic 3D model, in three different initialization

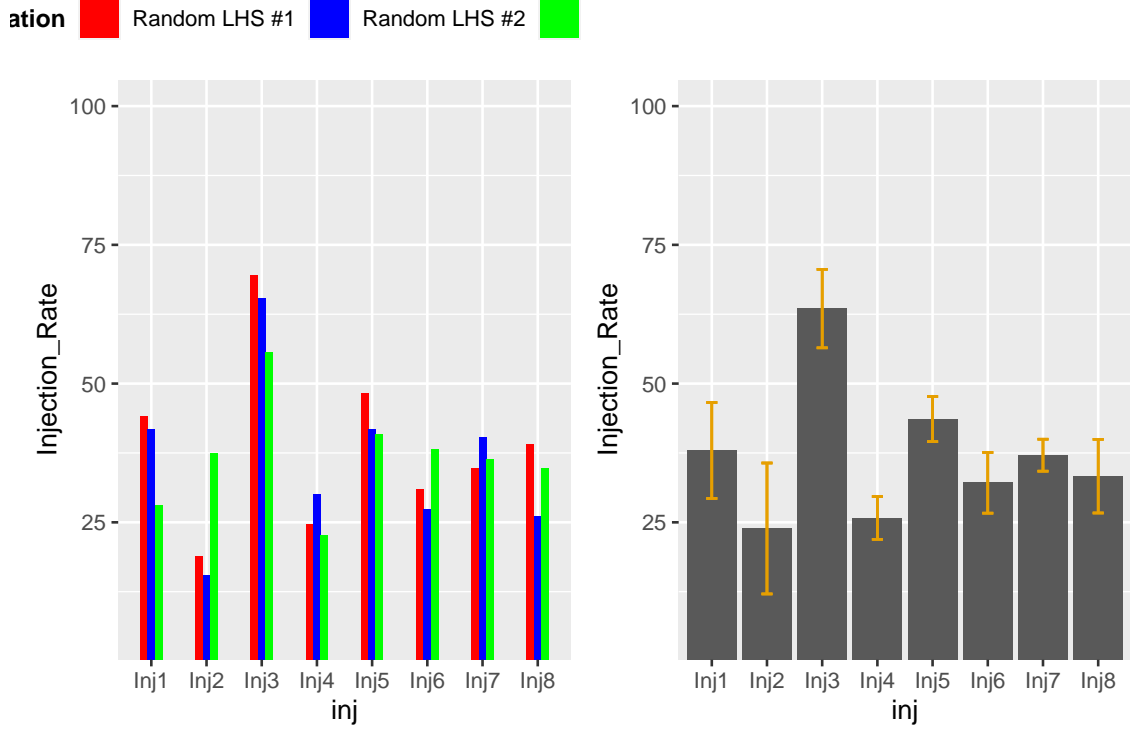


Figure 15: Left: final solution of optimization algorithm in three different initialization, Right: Mean and error bar of each injection rate at each injection wells

5. BayesOpt performance versus other alternatives

In this section the aim is to compare the performance of the Bayesopt workflow with other available optimization algorithm commonly used for reservoir optimization under uncertainty. The literature of field development optimization enjoys wide varieties of the workflow and algorithm applied to field development. Broadly speaking those can be divided into two categories adjoint-gradient and derivative-free. Adjoint methods, such as those described in (Forouzanfar and Reynolds, 2014; Li and Jafarpour, 2012; Volkov and Bellout, 2018) can provide computational advantage in terms of efficiency. They are, however, local methods, and it is known that broad (global) searches can be advantageous in field development optimization methods.(de Brito and Durlofsky, 2021b) - Therefore, in this work two well-know Derivative-free optimization (DFO) methods, extensively used reservoir optimization, named Genetic Algorithm (GA) (Chai et al., 2021; Holland, 1992a) and Particle Swarm Optimization (PSO) (Eberhart and Kennedy, 1995a; Jesmani et al., 2016) have been considered. In this section we provide a brief overview of each methods, but interested readers are refereed to the original papers.(Eberhart and Kennedy, 1995b; Holland, 1992b)

5.1. Particle Swarm Optimization (PSO)

PSO is a global stochastic search technique that operates based on analogies to the behaviors of swarms/flocks of living organisms. Originally developed by (Eberhart and Kennedy, 1995a), Considering a swarm with P particles, there is a position vector $X_i^t = (x_{i1}, x_{i2}, x_{i3}, x_{in})^T$ and a velocity vector $V_i^t = (v_{i1}, v_{i2}, v_{i3}, v_{in})^T$ at a t iteration for each one of the i particle that composes it. These vectors are updated through the dimension j according to the following equations:

$$V_{ij}^{t+1} = \omega V_{ij}^t + c_1 r_1^t (pbest_{ij} - X_{ij}^t) + c_2 r_2^t (gbest_j - X_{ij}^t) \quad (26)$$

where $i = 1, 2, \dots, P$ and $j = 1, 2, \dots, n$. Equation (26) explains that there are three different contributions to a particle's movement in an iteration. In the first term, the parameter ω is the inertia weight constant. In the second term, The parameter c_1 is a positive constant and it is an individual-cognition parameter, and it weighs the importance of particle's own previous experiences. The other parameter second term is r_1^t , and this is a random value parameter with $[0,1]$ range. The third term is the social learning one. Because of it, all particles in the swarm are able to share the information of the best point achieved regardless of which particle had found it, for example, $gbest_j$. Its format is just like the second term, the one regarding the individual learning. Thus, the difference $(gbest_j - X_{ij}^t)$ acts as an attraction for the particles to the best point until found at some t iteration. Similarly, c_2 is a social learning parameter, and it weighs the importance of the global learning of the swarm. And r_2 plays exactly the same role as r_1 . Where Equation (27) updates the particle's positions. (Almeida and Leite, 2019)

$$X_{ij}^{t+1} = X_{ij}^t + V_{ij}^{t+1} \quad (27)$$

5.2. Genetic Algorithm (GA)

Genetic algorithm is a stochastic search algorithms that use evolutionary strategies inspired by the basic principles of biological evolution. First developed by John Holland (Holland, 1975) and his collaborators in the 1960s and 1970s, later has been applied for optimization and search problems Goldberg and Holland (1988); Mitchell (1998). The evolution process is as follow: GA starts with the generation of an initial random population of size P , so for step $k = 0$ we may write $\theta_1^{(0)}; \theta_2^{(0)}, \dots, \theta_p^{(0)}$, (step 1). The fitness of each member of the population at any step k , $f(\theta_i^{(k)})$, is computed and probabilities $p_i^{(k)}$ are assigned to each individual in the population, usually proportional to their fitness, (step 2). The reproducing population is formed **selection** by drawing with replacement a sample where each individual has probability of surviving equal to $p_i^{(k)}$, (step 3). A new population $\theta_1^{(k+1)}; \theta_2^{(k+1)}, \dots, \theta_p^{(k+1)}$ is formed from the reproducing population using

Table 2: Parameters of GA and PSO Methods

parameters	value
PSO	
Size of the swarm	25
Local exploration constant	$5 + \log(2)$
Global exploration constant	$5 + \log(2)$
GA	
Population Size	25
Probability of crossover	80%
Probability of mutation	20%
Number of best fitness individuals to survive	5%

crossover and mutation operators, step (4). Then, set $k = k + 1$ and the algorithm returns to the fitness evaluation step, (back to step 2). When convergence criteria are met the evolution stops, and the algorithm deliver as the optimum (Scrucca, 2013).

5.3. Comparison in Fixed Reservoir Simulation Budget ($N=50$)

In first part of the comparison, we compare the Bayesopt with PSO and GA in fixed $\bar{J}(u)$. It means that optimization process could continue, until they use $\bar{J}(u) = 50$ function evaluations. It is worth to mention that in fact $\bar{J}(u) = 50$ is equal to 500 reservoir simulations, due to number of realization, $n_e = 10$ and ten computation per each $\bar{J}(u)$. Another point is parameters of PSO and GA. These two methods needs parameters to be defined by the user. In GA, these parameters are: Population Size, probability of crossover between pairs of chromosomes, probability of mutation in a parent chromosome, The number of best fitness individuals to survive at each generation. For PSO, the algorithm parameters are as below: size of the swarm, the local exploration constant, the global exploration constant.

In 16 results of comparison has shown. As all of three algorithm is stochastic (meaning they depends on initial random samples), comparison has been repeated three times. We would like to note that in 16 the y axis is “Max NPV Reached,” meaning that in each generation of GA and PSO algorithm, the “Max” of the each generation has been shown. Moreover, the Figure shows that in BayesOpt method, number of $\bar{J}(u)$ grows as $n_{initial} + n_{iteration}$, which in this case in $n_{initial} = 40$ and $n_{iteration} = 10$, summing up to 50. Whereas, in PSO and GA, number of $\bar{J}(u)$ grows as $n_{popsize} \times iteration$. As 16 shows, in all repetition, the BayesOpt outperform the other two algorithms with reaching higher NPV at fixed simulation budget. Part of performance could be attributed how algorithms use forward model. In BayesOpt, after initial sampling, the algorithm sequentially query a “one” from the expensive function, while GA and PSO needs another sample size n_p per each iteration.

In this work we did not suffice the comparison to only 17. In Figure 17 we further allowed the number of $\bar{J}(u)$ evaluations to 250, while keep the results of BayesOpt to the 50. Meaning that PSO and GA algorithm

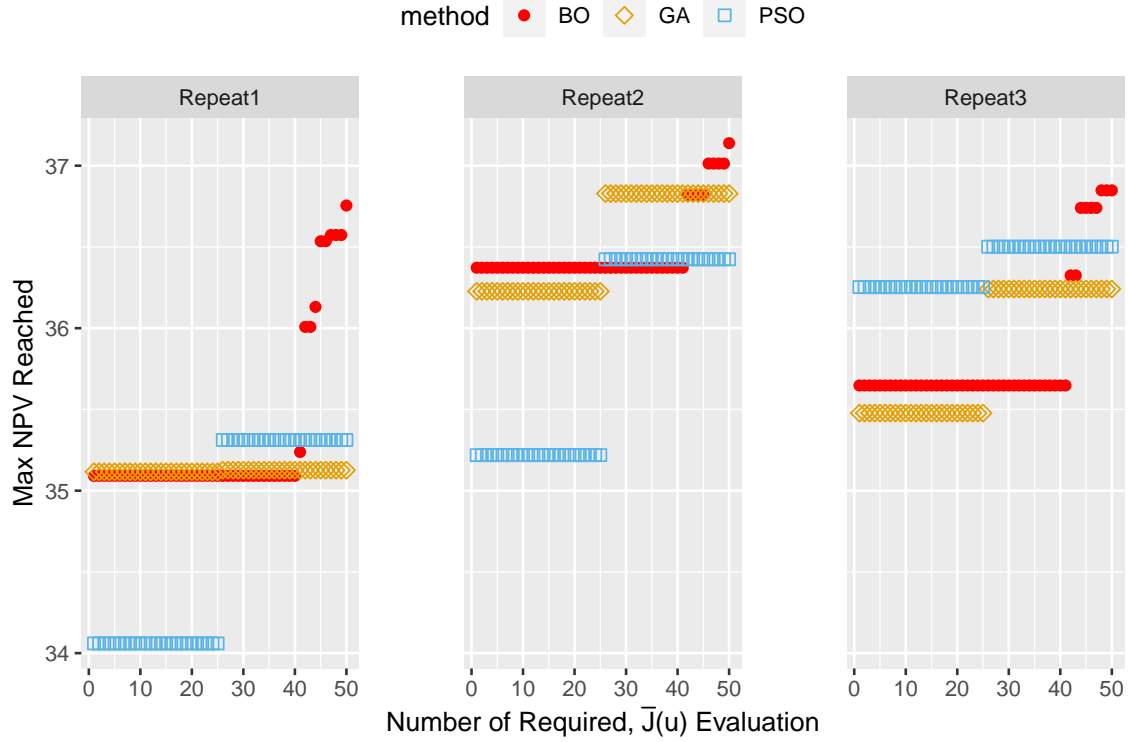


Figure 16: Comparison of GA, PSO and BayesOpt performance at fixed function evaluation budget

Table 3: Summary table for comparison of GA/PSO and Bayesopt

Optimization Method	Maximum Achieved NPV (median)	$\bar{J}(u)$ Evaluations
Bayesian Optimization	36.848	50
Particle Swarm Optimization	36.894	250
Genetic Alghorithm Optimization	36.429	250

will enjoy another 8 iterations ($25 \times 8 = 200$) and then their results, will be compared with BayesOpt from previous section. Figure 17 does not convey a single message about performance of these methods. In 3 median value of three algorithm was compared. The value in second column of 3 is mean value of each optimization method. (In three repetitions, the maximum achieved NPV is $a < b < c$, b was selected). As the 3 shows, the difference between the NPV value of BayesOpt is almost negligible compared to PSO and GA, while the max NPV in BayesOpt was achieved in 50 $\bar{J}(u)$ while other two in 250. In this work and optimization setting of the 3D, synthetic reservoir model, BayesOpt reaches same optimal solution, while having computaional complexity of 5X (times) less.

Comparing the Final Solution u of the Opt algorithms... (the Median Replication was used)

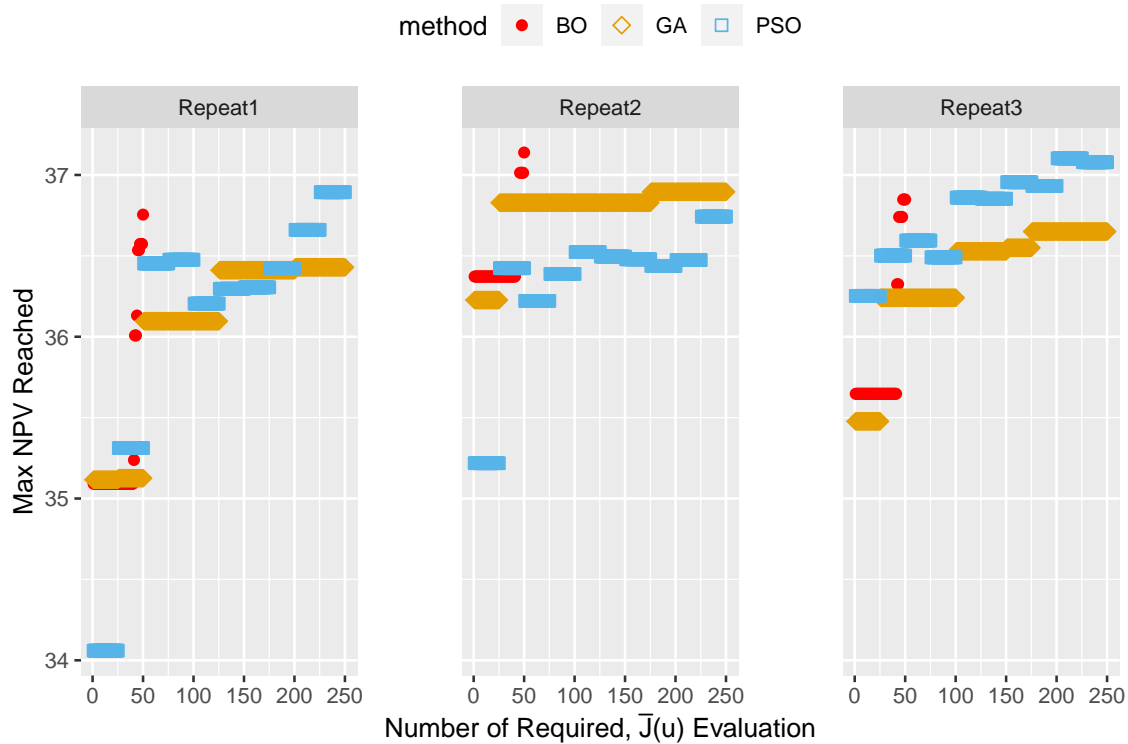
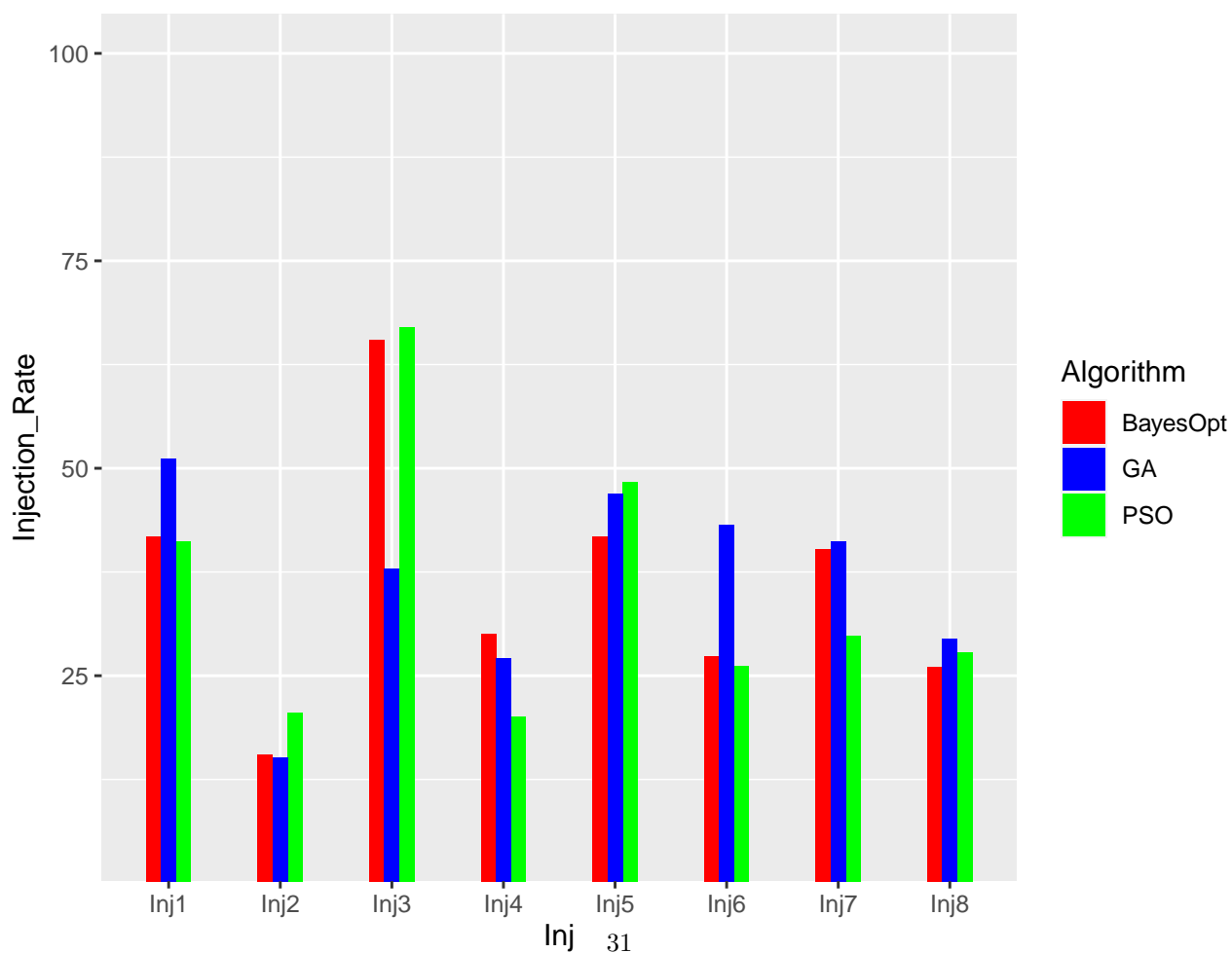


Figure 17: Comparison of GA PSO and BayesOpt performance at fixed function evaluation budget



7. Acknowledgements

This work received support from the Research Council of Norway and the companies AkerBP, Wintershall-DEA, Vår Energy, Petrobras, Equinor, Lundin, and Neptune Energy, through the Petromaks-2 DIGIRES project (280473) (<http://digires.no>). We acknowledge the access to Eclipse licenses granted by Schlumberger.

References

- Almeida, B.S.G. de, Leite, V.C., 2019. Particle Swarm Optimization: A Powerful Technique for Solving Engineering Problems. IntechOpen.
- Almeida, L.F., Tupac, Y.J., Pacheco, M.A.C., Vellasco, M.M.B.R., Lazo, J.G.L., 2007. Latin American & Caribbean Petroleum Engineering Conference. OnePetro. doi:10.2118/107872-MS
- Asadollahi, M., Nævdal, G., Dadashpour, M., Kleppe, J., 2014. Production optimization using derivative free methods applied to Brugge field case. *Journal of Petroleum Science and Engineering* 114, 22–37. doi:10.1016/j.petrol.2013.12.004
- Chai, Z., Nwachukwu, A., Zagayevskiy, Y., Amini, S., Madasu, S., 2021. An integrated closed-loop solution to assisted history matching and field optimization with machine learning techniques. *Journal of Petroleum Science and Engineering* 198, 108204. doi:10.1016/j.petrol.2020.108204
- Chen, Y., Oliver, D.S., Zhang, D., 2009. Efficient ensemble-based closed-loop production optimization. *SPE Journal* 14, 634–645. doi:10.2118/112873-PA
- de Brito, D.U., Durlofsky, L.J., 2020. Well control optimization using a two-step surrogate treatment. *Journal of Petroleum Science and Engineering* 187, 106565. doi:10.1016/j.petrol.2019.106565
- de Brito, D.U., Durlofsky, L.J., 2021b. Field development optimization using a sequence of surrogate treatments. *Computational Geosciences* 25, 35–65. doi:10.1007/s10596-020-09985-y
- de Brito, D.U., Durlofsky, L.J., 2021a. Field development optimization using a sequence of surrogate treatments. *Computational Geosciences* 25, 35–65. doi:10.1007/s10596-020-09985-y
- Dennis, J.E., n.d. A rigorous framework for optimization of expensive functions by surrogates 13.
- Do, S.T., Reynolds, A.C., 2013. Theoretical connections between optimization algorithms based on an approximate gradient. *Computational Geosciences* 17, 959–973. doi:10.1007/s10596-013-9368-9
- Eberhart, R., Kennedy, J., 1995a. A new optimizer using particle swarm theory. *Ieee*, pp. 39–43.
- Eberhart, R., Kennedy, J., 1995b. A new optimizer using particle swarm theory. *Ieee*, pp. 39–43.
- Echeverría Ciaurri, D., Isebor, O.J., Durlofsky, L.J., 2010. Application of derivative-free methodologies to generally constrained oil production optimization problems. *Procedia Computer Science, ICCS 2010* 1, 1301–1310. doi:10.1016/j.procs.2010.04.145
- Foroud, T., Seifi, A., 2016. A Guided Pattern Search with a non-intrusive reduced order modeling for oil production optimization: Brugge field case study. *Journal of Petroleum Science and Engineering* 147, 570–584. doi:10.1016/j.petrol.2016.09.026
- Forouzanfar, F., Reynolds, A.C., 2014. Joint optimization of number of wells, well locations and controls using a gradient-based algorithm. *Chemical Engineering Research and Design* 92, 1315–1328.

- Goldberg, D.E., Holland, J.H., 1988. Genetic algorithms and machine learning.
- Harding, T.J., Radcliffe, N.J., King, P.R., 1998. Hydrocarbon production scheduling with genetic algorithms. SPE Journal 3, 99–107. doi:10.2118/36379-PA
- Holland, J., 1975. Adaptation in natural and artificial systems, university of michigan press, ann arbor,”. Cité page 100.
- Holland, J.H., 1992a. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. MIT press.
- Holland, J.H., 1992b. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. MIT press.
- Hong, A.J., Bratvold, R.B., Nævdal, G., 2017b. Robust production optimization with capacitance-resistance model as precursor. Computational Geosciences 21, 1423–1442. doi:10.1007/s10596-017-9666-8
- Hong, A.J., Bratvold, R.B., Nævdal, G., 2017a. Robust production optimization with capacitance-resistance model as precursor. Computational Geosciences 21, 1423–1442. doi:10.1007/s10596-017-9666-8
- Isebor, O.J., Ciaurri, D.E., Durlofsky, L.J., 2013. SPE Reservoir Simulation Symposium. OnePetro. doi:10.2118/163631-MS
- Jansen, J.D., Fonseca, R.M., Kahrobaei, S., Siraj, M.M., Essen, G.M.V., Hof, P.M.J.V. den, 2014. The egg model – a geological ensemble for reservoir simulation. Geoscience Data Journal 1, 192–195. doi:10.1002/gdj3.21
- Jesmani, M., Bellout, M.C., Hanea, R., Foss, B., 2016. Well placement optimization subject to realistic field development constraints. Computational Geosciences 20, 1185–1209. doi:10.1007/s10596-016-9584-1
- Li, L., Jafarpour, B., 2012. A variable-control well placement optimization for improved reservoir development. Computational Geosciences 16, 871–889.
- Lushpeev, V., Margarit, A., 2018. OPTIMIZATION OF OIL FIELD DEVELOPMENT PROCESS BASED ON EXISTING FORECAST MODEL. Journal of Applied Engineering Science 16. doi:10.5937/jaes16-17218
- Mitchell, M., 1998. An introduction to genetic algorithms. MIT press.
- Murphy, K.P., 2022. Probabilistic machine learning: An introduction. MIT Press.
- Nwachukwu, A., Jeong, H., Sun, A., Pyrcz, M., Lake, L.W., 2018b. SPE Improved Oil Recovery Conference. OnePetro. doi:10.2118/190239-MS
- Nwachukwu, A., Jeong, H., Sun, A., Pyrcz, M., Lake, L.W., 2018a. SPE Improved Oil Recovery Conference. OnePetro. doi:10.2118/190239-MS
- Sarma, P., Aziz, K., Durlofsky, L.J., 2005. SPE Reservoir Simulation Symposium. OnePetro. doi:10.2118/92864-MS

488 Scrucca, L., 2013. GA: A Package for Genetic Algorithms in R. *Journal of Statistical Software* 53, 1–37.
489 doi:10.18637/jss.v053.i04

490 Stordal, A.S., Szklarz, S.P., Leeuwenburgh, O., 2016. A Theoretical look at Ensemble-Based Optimization in
491 Reservoir Management. *Mathematical Geosciences* 48, 399–417. doi:10.1007/s11004-015-9598-6

492 Torczon, V., 1997. On the convergence of pattern search algorithms. *SIAM Journal on Optimization* 7, 1–25.
493 doi:10.1137/S1052623493250780

494 van Essen, G.M., Zandvliet, M.J., Jansen, J.D., 2009. Robust Waterflooding Optimization of Multiple
495 Geological Scenarios. *SPE Journal* 9.

496 Volkov, O., Bellout, M.C., 2018. Gradient-based constrained well placement optimization. *Journal of*
497 *Petroleum Science and Engineering* 171, 1052–1066.