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:

While developing a field, prediction of reservoir response to change in the variable is an am important factor to have an efficient reservoir management. The field of reservoir engineering is rich in the development and application of full-physics numerical simulations for forward modeling. However, the computational power needed to run such numerical simulators most of the time is huge. Especially, the framework of the Robust Optimization where uncertainty are considered through multiple of geological realization (thousand or multi-thousand), the practical applicability of such forward modeling is considerably limited. To address this challenge, the proxy-modeling for reservoir managemnet has emerged to reduce the cost of forward simulation. The root of this field goes back to the work of the (Bruce, 1943) where the analogy between flow of electricty in tthorugh te electrical device and the response of the reservoir was constructed. (Albertoni and Lake, 2003) Proposed the a Multivariate Linear Regression (MLR) to estimate the production from the well, where it claimed that total production at each is linear combination of injection rates with diffusitivity filter. Building on the work of (Albertoni and Lake, 2003), (Yousef, 2006) proposed a new procedure to quantify communication between wells, where for each injector/producer pair, two coefficients, capacitance to quantify connectivity and time constant to quantify the degree of fluid storage were considered. (Sayarpour, 2008) used superposition in time to analytically solve the numerical integration in CRM. (Zhao et al., 2015)(Zhao et al. 2015) articulated that CRM models are not applicable when production rates change significantly, mainly due to those models neglect to include intearction of production-production and injector-injector pair well. Formulating the model in the framework titled INSIM (interwell numerical simulation model), it was used as efficient replacement to reservoir simulator when history-matching phase rate data or water-cut data obtained during water flooding operations.

Seperatley, in sake of utilization of recent advancemnet in the world of Information technology, couple of reaserach has been don eon the development of “Surrogate Reservoir Models” - (Mohaghegh and Guruswamy, 2006) proposed the workflow for SRM model where Fuzzy Paatern Recognition (FPR) technology was dimensionality reduction, in both static and dynamic parameters of the reservir . Key Performance Indicator (KPI) was used to select the most important variable.- (Sampaio, 2009) tried on use of feed-forward neural networks as nonlinear proxies of reservoir response. A few set of rock and fluid properties were considere a input (porosity, permeability in “x” direction, permeability in “y” direction, and rock compressibility) where in developing the neural network model, only one hidden layer was used. flow proxy modeling to replace full simulation in history matching, and built the proxy with a single hidden layer artificial neural network (ANN). To predict the oil production from the SAGD rocess, (Fedutenko et al. 2014) (Fedutenko et al., 2014)employed the RBF Neural Network to predict the cumulative oil production over the time horizon of production (10 years) .

1.1. Survey of suurgate -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, $\chi \subseteq \mathbb{R}^D$:

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

In Figure 1 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:

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

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

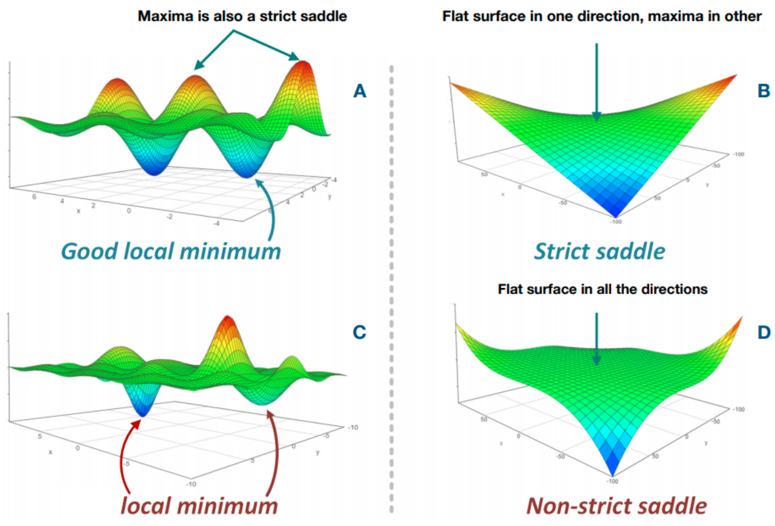

Maxima is also a strict saddle
A
Good local minimum
Flat surface in one direction, maxima in other
B
Strict saddle
Flat surface in all the directions
C
local minimum
D
Non-strict saddle

Figure 1: 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 $\mathbf{u} = [u_1, u_2, \cdots, u_N]^D$, 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)

$$\overline{J}(\mathbf{u}) = \frac{\sum_{re=1}^{n_e} J(\mathbf{u}, \mathbf{G_{re}})}{n_e} \tag{3}$$

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., 2018). 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 $\overline{J}(u)$. 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 ~1hr. Then, simply having 100 realizations means that each forward computation of $\overline{J}(u)$ 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$\frac{6\times30\times24}{100} = 43.2 \approx 50$ 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. Overal 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. Evaluet 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. Probalistic 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, $\mathbf{f} = [\mathbf{f}(\mathbf{x_1}), ..., \mathbf{f}(\mathbf{x_M})]$, is jointly Gaussian, with mean and Covariance

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

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 $\mathcal{D}$.

$$\mathbf{f}_* = [\mathbf{f}(\mathbf{x_1}), .., \mathbf{f}(\mathbf{x_{N_*}})] \tag{5}$$

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

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

$$\begin{aligned} \mu_X &= [m(x_1), ..., m(x_N)] \\ \mu^* &= [m(x_1^*, ...m(x_N^*))] \end{aligned} \tag{7}$$

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

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	$\kappa(x, x') = \sigma_f^2 exp(-(\frac{\|h\|}{\ell})^p)$

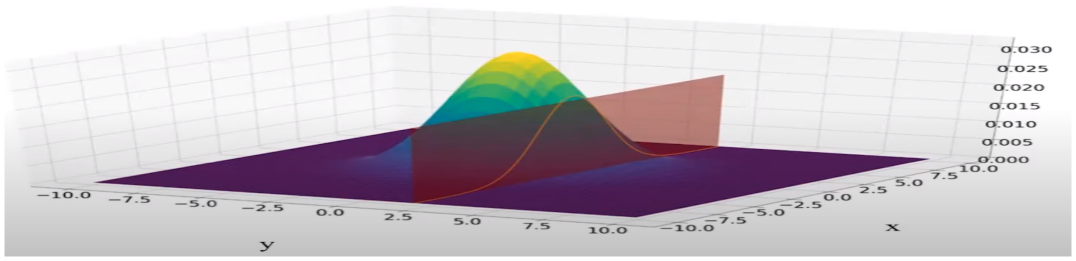


Figure 2: proof of

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

3.1.1.2. Covariance Kernel, Parameter estimation.

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

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

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

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

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

3.1.2. Step 2. Posterior of Probabilistic Model

3.1.2.1. Posterior of Gaussain Process, (conditioning on initial data). Here in 2

$$p(f_*|X_*,\mathcal{D})=\mathcal{N}(f_*|\mu_*,\sum\nolimits_*) \tag{13}$$

$$\begin{gathered}\mu_*=m(\mathbf{X}_*)+\mathbf{K^\intercal}_{\mathbf{X},*}\mathbf{K^{-1}}_{\mathbf{X},\mathbf{X}}(\mathbf{f_X}-\mathbf{m}(\mathbf{X}))\\ \sum\nolimits_*=\mathbf{K}_{*,*}-\mathbf{K}_{X,*}^\intercal\mathbf{K}_{X,X}^{-1}\mathbf{K}_{X,*}\end{gathered} \tag{14}$$

3.1.3. Example of Step.1 and Step.2

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

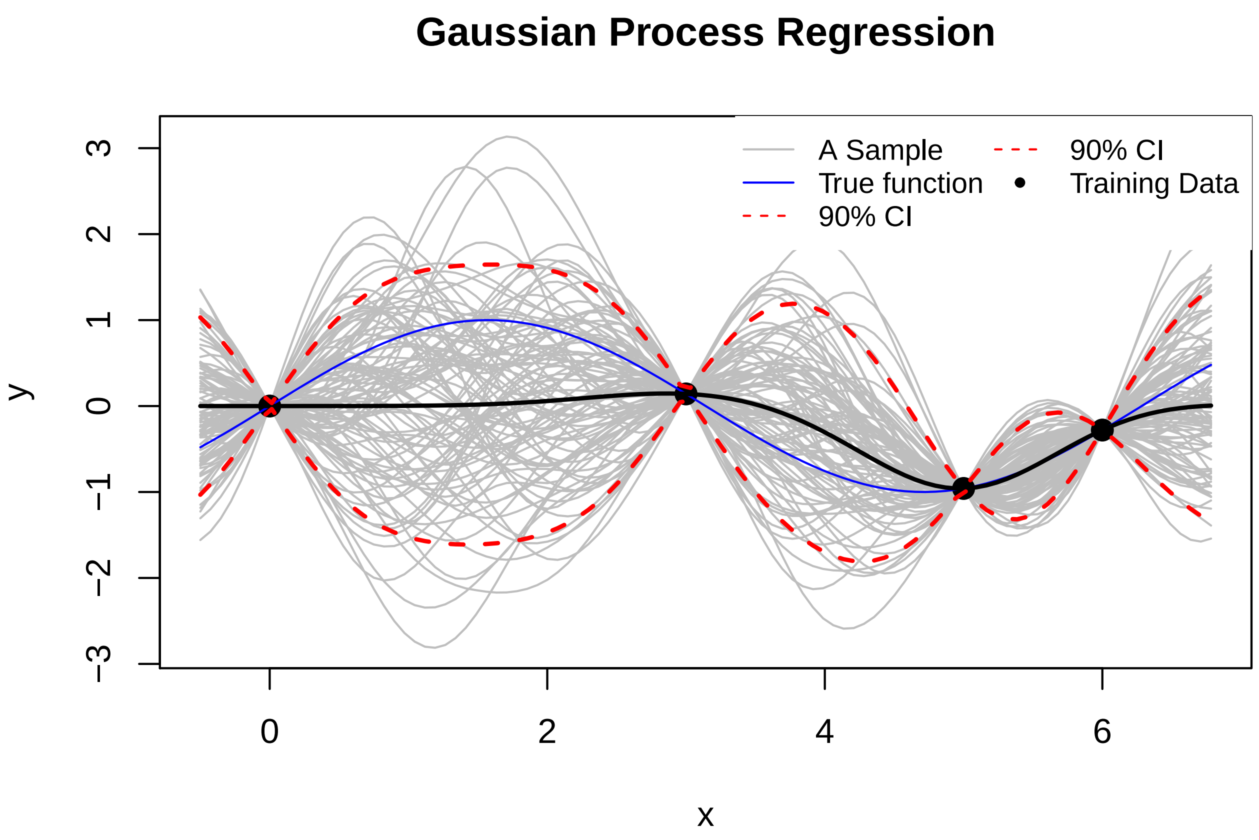


Figure 3: Gaussin Process Regression conditioned on 4 points

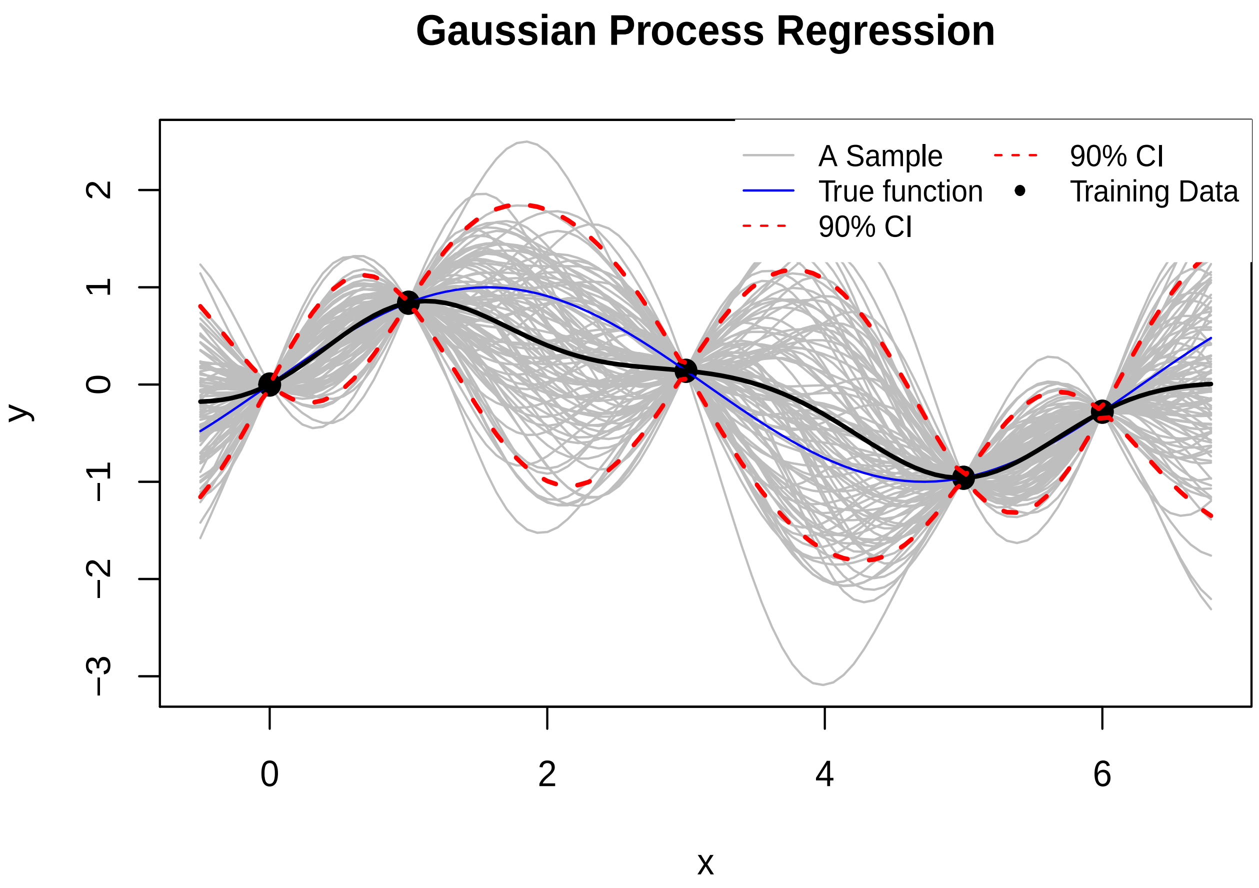


Figure 4: Gaussin Process Regression conditioned on 5 points

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*

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

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

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

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

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

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

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

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

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

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

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

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

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

$$u_M = \underset{u\in\text{constraints}}{\text{argmax}}\ \overline{J}(u) \tag{21}$$

unsolvabale!

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

solvabale!

- $\alpha_{EI}(u)$ is inexpensive to evaluate.
- 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}.

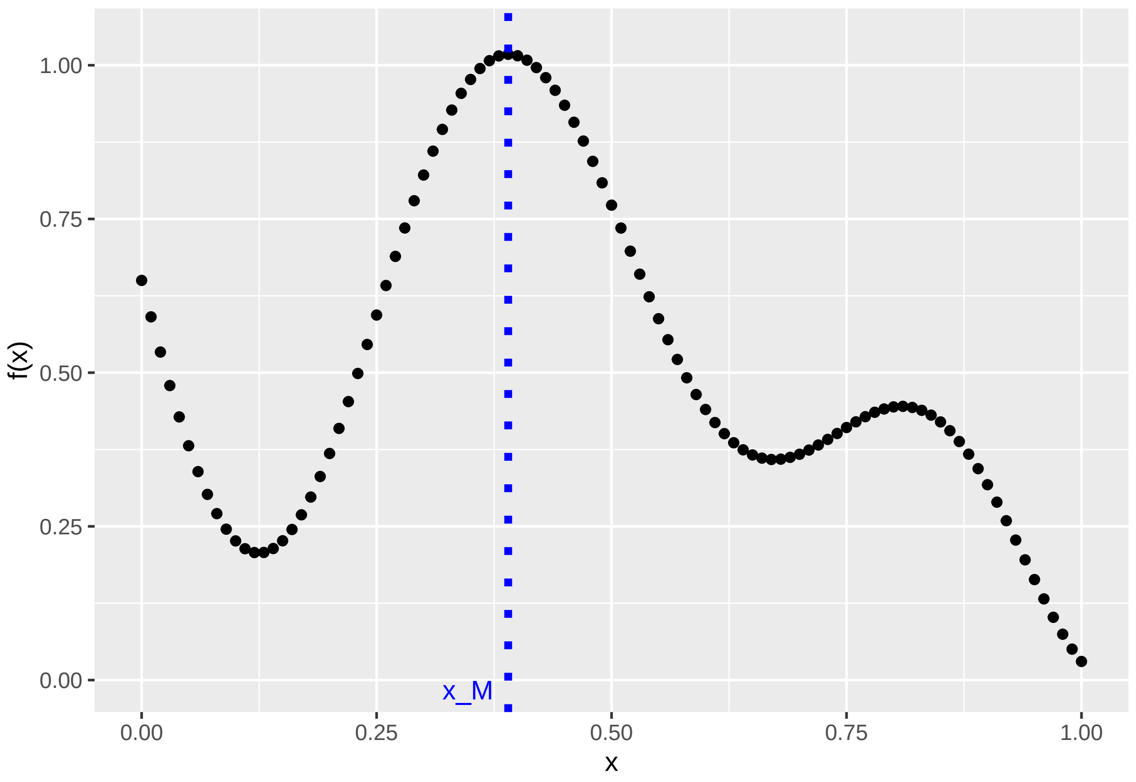


Figure 5: Plot of 1-D equation with blue dash line representing the global optimum

4. Example Cases

4.1. 1-D toy Problem

In this section, a 1-D toy problem is considered to illkuystrate the Bayes Optimization workflow dicussed in the previous section. The 1-D problem was selected sinnce it will help to visuluze all the steps of the workslow making easier explanation of the concepts. Though, it it can bee senn from the next section, the workflow can easily extened to to higer dimetional problem. The *True function* to be optimized in this section has an analytical expression as, given the box constrains:

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

Since the analytical expression of function available and being 1-D problem, the global optimum of the function had been found at the $x_M = 3.90$. The plot of the function and the optimum point has been shown in the Figure 5

However, it is worth to mention that the analytical expression of objective function in many of real world problems are not avilable, what is avilable is a *samples* from the objective function. Thefore, in the coming example a few samples are sequentiall 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 copare the performace of Bayesian optimisation algorithem.

Thefore, as Figure 6, 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 represnets the samples from 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 6 is there no uncertainty on the sample point. As shown, there is no grey zone on sample point since as was dicussed in the previous section, here we consider the "noise-free" observation. Also, worth to mention that we have wide more uncertainity (wider grey band) in the reas that are more distant from the observation, simply meaning we are less uncertain close to observation points. on the "extrpolation," meaning in the ares outseide of the observation points, the probalistic model shows inetrsting behaviour. on those "exterme" area, the mean curve tend to move toward the mean of all observation points , here around 0, showing the model refelctes the mean-revervion behaviour when it comes extrpolation.

The lower part of Figure 6, shows the plot of utility function at each x values. Worth to note that as the plot suggest, the utility(α_{EI}) function will have the muti-modal structure, meaning in the optimization process multi-start gradient method will be helpful, in order to avoid stock in the local optima. In this work, as was explained in the preious section, the multi-start gradient method was used. The blue dotted line line shows the the x_{next} which is the point where the utility function, is maximum. Then this x_{next} is qured from the real function, and the the pair of $(x_{next}, f(x_{next}))$ is added to the intial data set, $\mathcal{D}$. Going back to the lower figure at Figure 6, the utility has two mode 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 rfereed to the upper plot and it is clear that there is more uncertainity around point $x^-_{0.5}$ than $x^+_{0.5}$ which given the form of utility function, that is understandable. The utility function always looking for the point that not only maximize the mean value, but also intereded in the points that has higher variannce, which is the case between two points $x^+_{0.5}$ and $x^-_{0.5}$.

Calling the Figure 6 as the iteration # 1, now we can start sampling sequentially. In the Figure 7 another two iteration has been provided. Wher in each row, the plot on the left represents the posterior on the gaussian condistioning, the right show the utility function. Note that in the Figure 7 all axis labels and legned were not included, to have better visibity. (more info about each plot can ben found in 6) . Interesting to see tat in this example case, at the irteration #2, the workflow query the point $x = 0.385$ which presents the best point so far found through BayesOpt workflow. Thefiore, after just two iteration we are around $\frac{x_{best}}{x_M} = \frac{0.385}{0.390} = 98.7$ of the global optima. Although this is case for 1-D problem, it is clearly showing the stength of the workflow to approach the lglobal optima, in as few as possible iteration. In this case after iteration#2, tyhe total

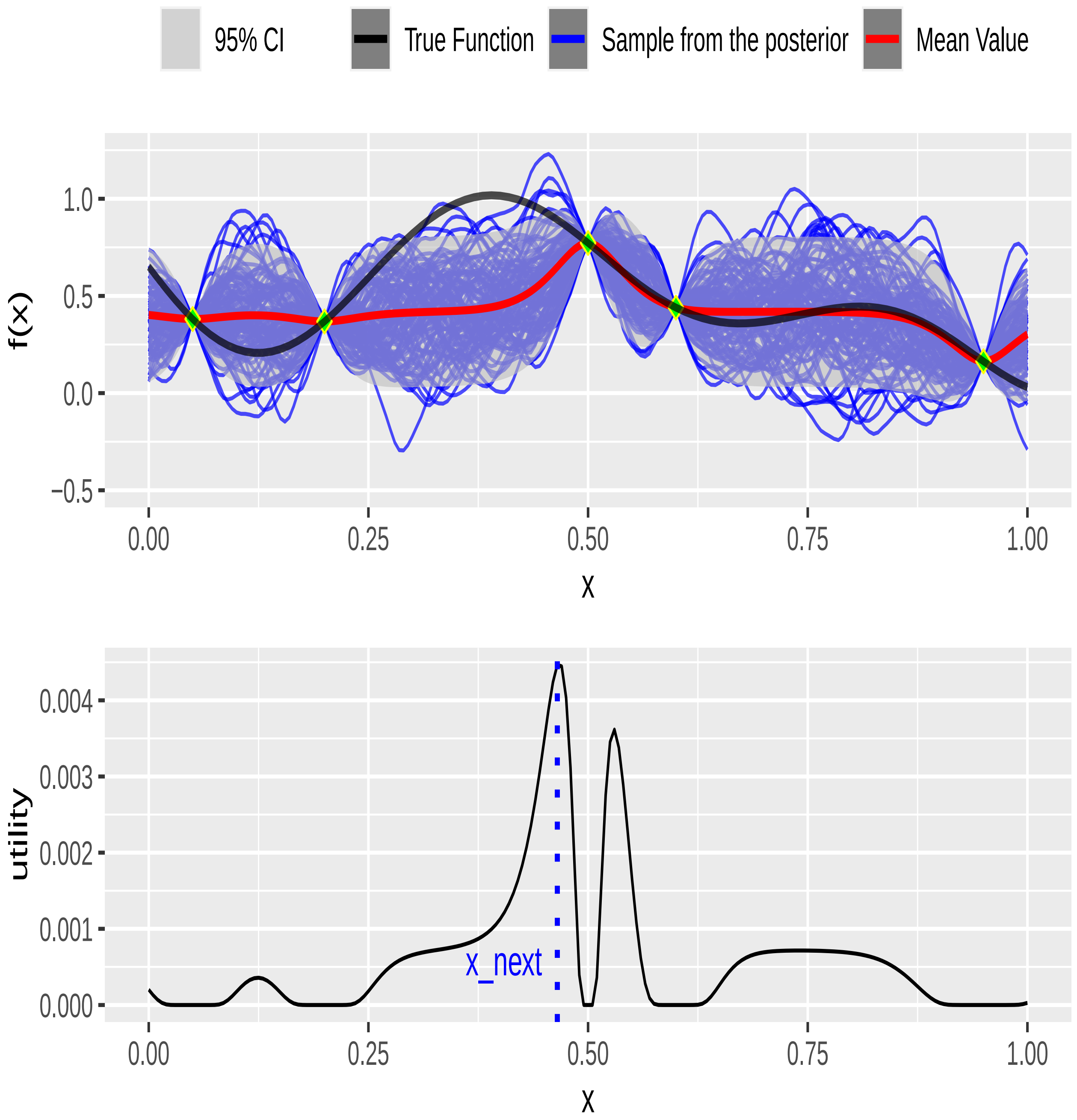


Figure 6: Ite1 - Top: Gaussian posterior over the initial sample points; Lower: Utility function over the x values

number of tim,e that the real function has been quered is size$(\mathcal{D})$ + size$(totaliteration) = 5 + 2 = 7$.

Before going to apply the same workflow at the field scale, the 1-D example presented here offer another useful feature of the Bayesian Optimisation. Looking at 7, we can see that the maximum of utility function is at the iteration # 3 in order of 10^{-6} . That show that after optimization, eve best point to be queried in the next section has a very little utility. So can safely stop the process, since querying points to be sampled from the expensive function has a negligible potential to improve our search in optimization.

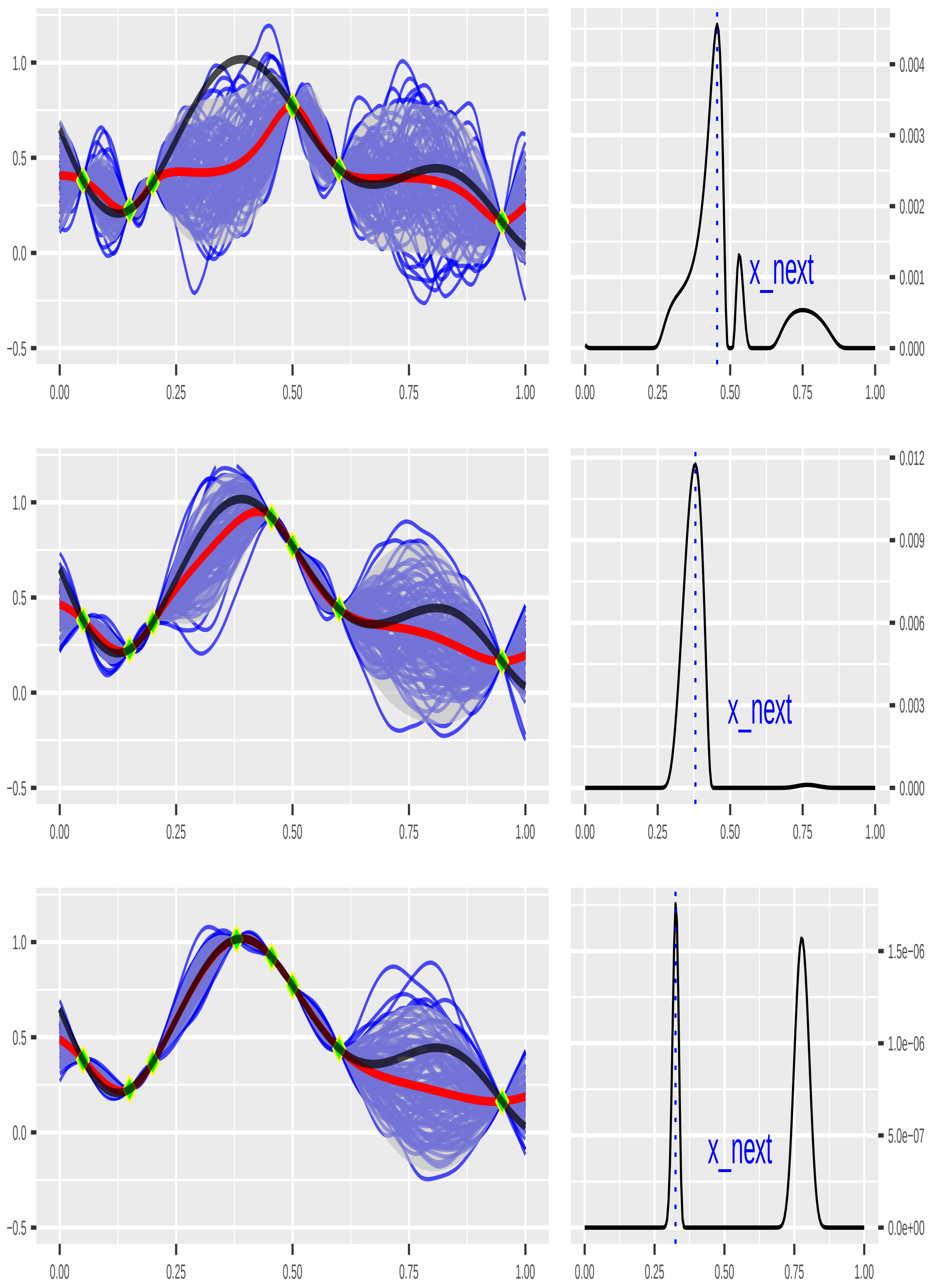


Figure 7: Gaussian posterior of over the initial sample points

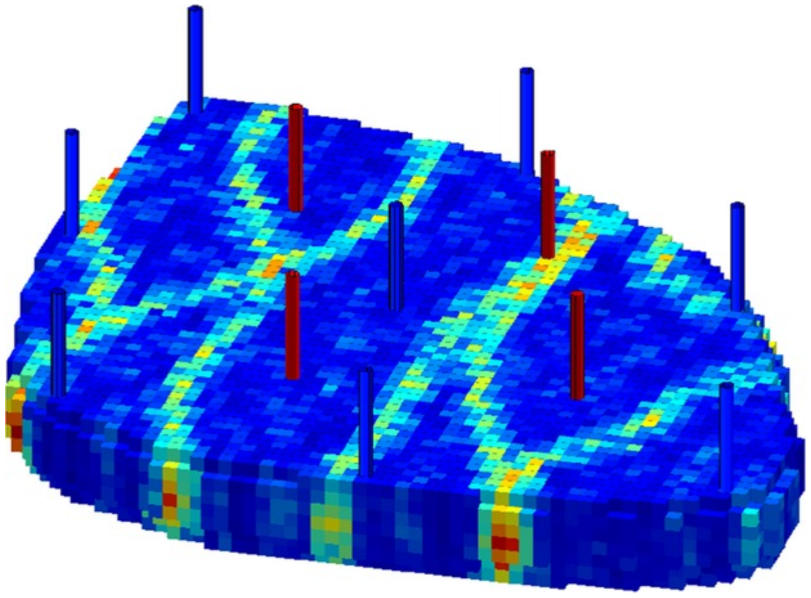


Figure 8: Well locations in Egg model, blue ones are injection, the red producers

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 trough introduction of the model and gelogical describtion 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 8. The has a 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 9.(Hong et al., 2017b).

Relative permeabilities and the associated fractional flow curve of the model have shown in right side of Figure 9 .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 maiintaned over life-cycle of reservoir is going to be optimized. Thus, given eight injection wellls, the optimizatijon workflow has the eight dimnetions.However, the optimization in not unbounded, the water can be adjusted from 0 to 100 m3/day, making the box-constrain 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:

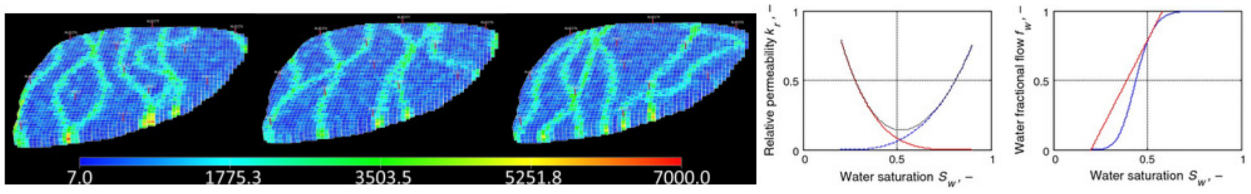


Figure 9: Left: Three geological realizations of the 3D model; Right: Rel perm and fractional flow curve

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

Equation (24)

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

$$u = [u_{inj1}, u_{inj2}, u_{inj3}, u_{inj4}, u_{inj5}, u_{inj6}, u_{inj7}, u_{inj8}]^{\intercal} \tag{25}$$

As the (24) suggest, the $\overline{J}(u)$ need some parameters to be defined. The oil price (P_o), water production cost (p_{wp}) and water injection cost (P_{wi}) in $dollar/m^3$ has been provided in the Table 1. Also, in this work the cash flow is disconted daily and the discount factor is avilable in the 1. We would like to note that in this work due to avoid further computional 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 stratifcation of the CDF of each variable, leading to better coverage of the input variable space. The Figure 10 show the results of the $\overline{J}(u)$ for each sample from LHS. Also, The maximum $\overline{J}(u)$ 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

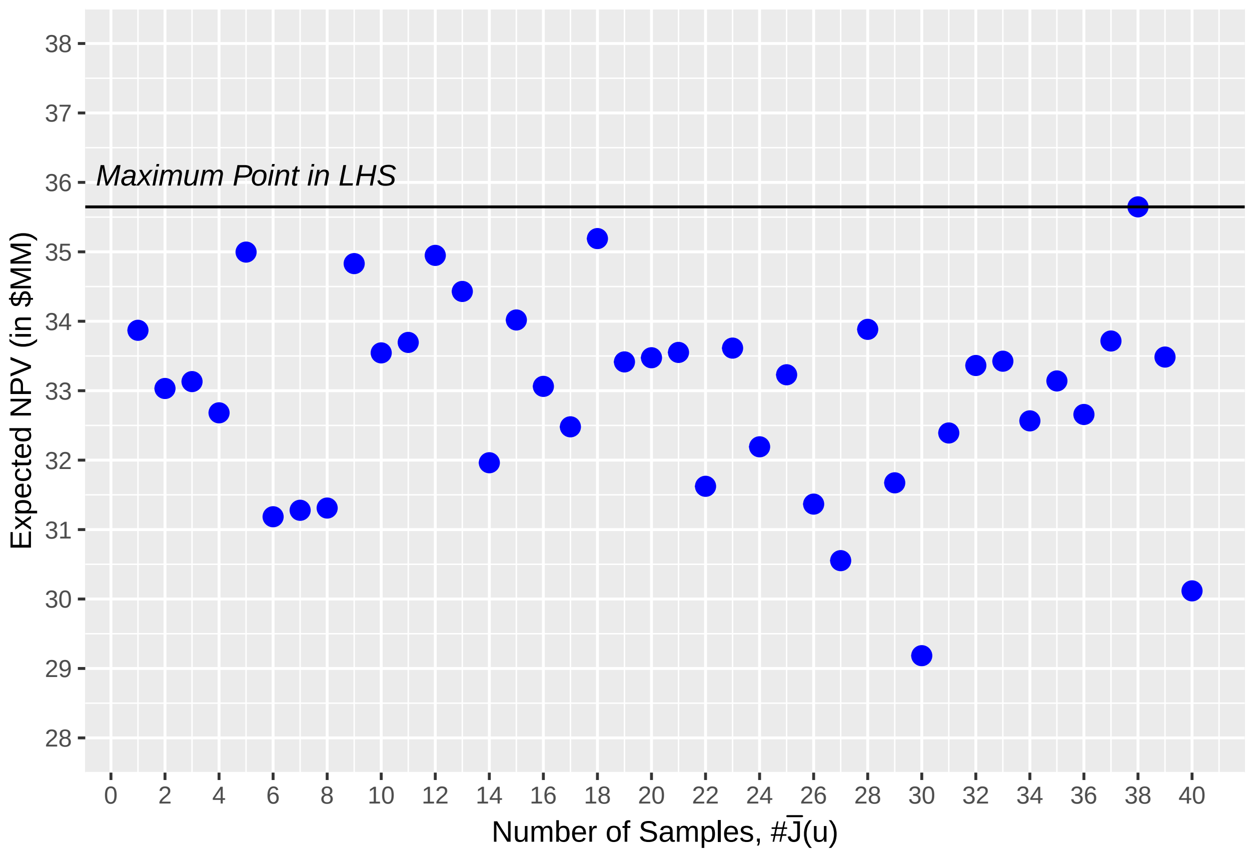


Figure 10: Expected NPV as result of forty sampling from LHS

10 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-dimetional, and hard to visulize. The Figure 11 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 algorith. 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 uncertainity, 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 $max\overline{J}(u) = 36.85\$MM$.

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

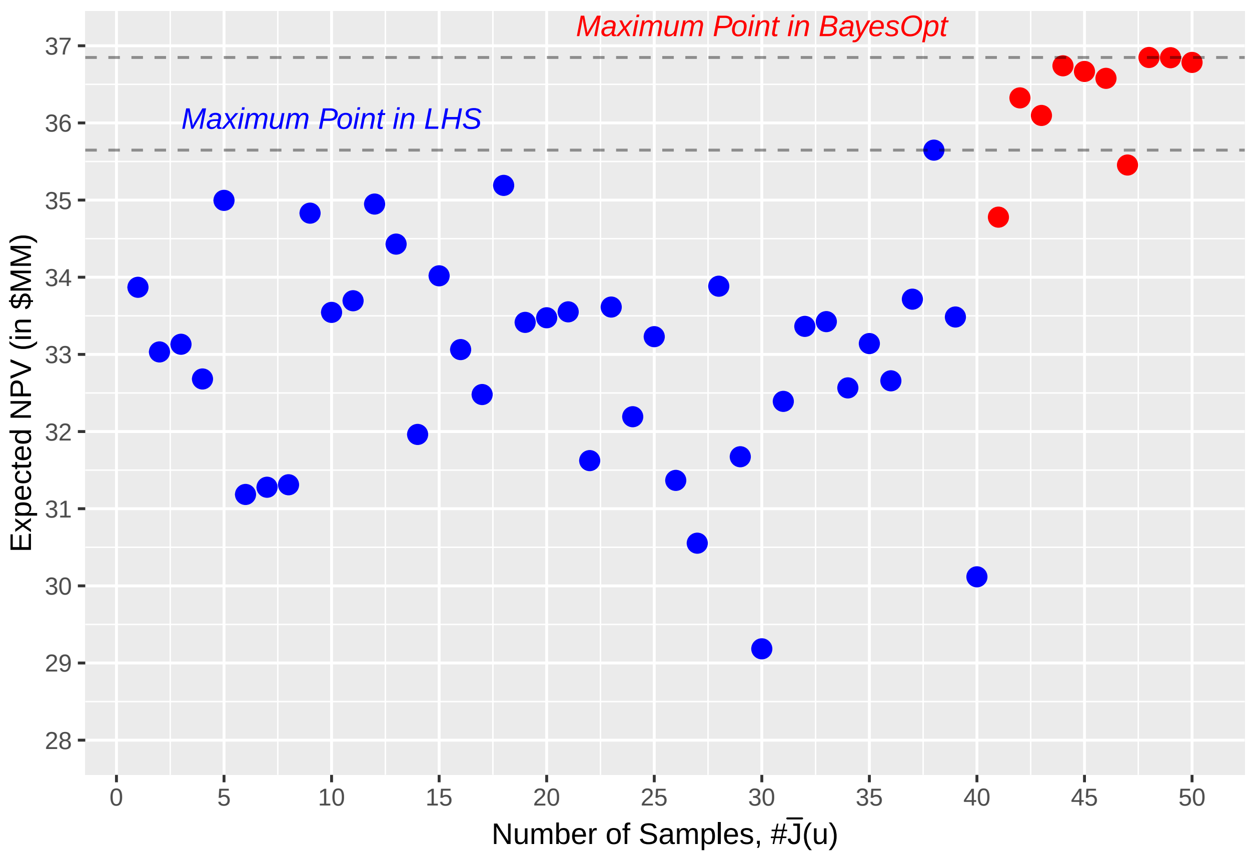


Figure 11: Blue points represnts the sample from LHS, red points represents the samples from the BayesOpt Workflow

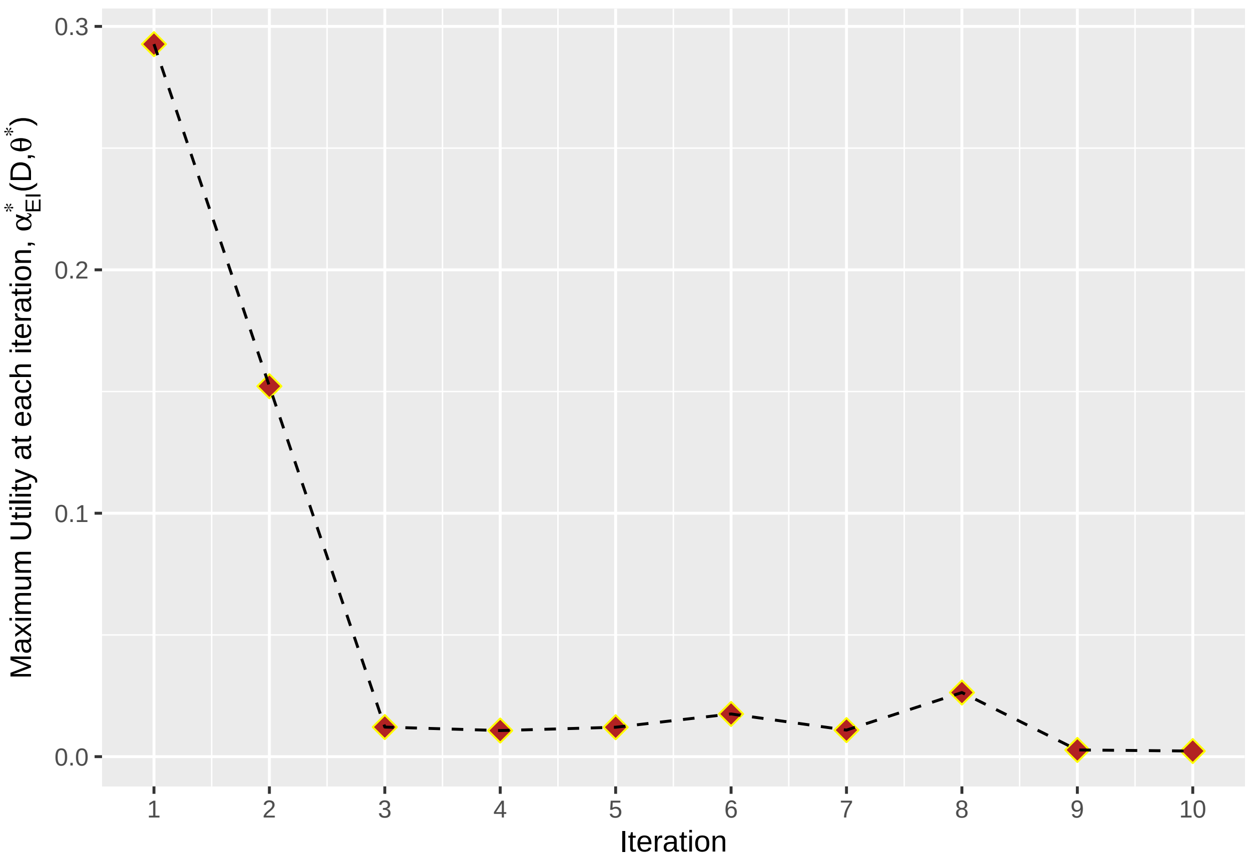


Figure 12: Maximum utility at each iteration, after running L-BFGS-B to find the u with max utility, α^*_{EI}

information about the optimization process. The Figure 12 plots the $\alpha^*_{EI}(\mathcal{D}, \theta^*)$ (Equation (22))versus the ten iteration in this work. In fact the notaion 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 intepreted 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 inherintely has stochasric natrae (from this perspective that having thje diffrenet initialization in LHS sampling will affect the final solution), in this section BayesOpt is repeated with diffret initilization. Ideally, this repeation shouwl be conducted 100 or 1000 times, to get better overview of the convergence of the algorithm given diffrent initilization. Though, because of the computional burden, in this work only three repeations were performed. optimization Repeat the Optimization, three times, in different initial design points. Figure 13 shows results of three repeations. At each repeation (top, middle, bottom), the blue dots come from diffrente seed numbers and they are diffrente. Then, gicen that initialization $\mathcal{D}$, sequential sampling from the expenive function is perfomred, shown in the red points. Like previous case, in these repeations, 40 samples drawn from LHS algortihem, the 10 were taken thorigh BAyesOpt lagorith, totaling 50 samples. At each row of the Figure 13, two horizontal lines show the maximum point $\overline{NPV}$ in both random sampling phase (LHS) and BayesOpt phase. As it can be noted from the Figure 13, at each repeation, the BayesOpt will improve the solution with small sample evaluation of the $\overline{J}(u)$. Thefore, improvemnet following the BayesOpt phase indepned of the initial design, yet the bigger question is whether given different initial design, the algorithm converge the vicinity of global optima. What is refered here is that if having different initilaization will lead completely different final solution, that hints that the algorithm has a "local" search, in conrast, if the solutions leads to one specif close u^*, that reprsents that algorithm have a "global" view on the surface of the objective function. In the case of "global" optimization having diferent initilizatin should lead to simular final solution, since the algorithm will not get stuck in local optimum points, close to initilalized data. This is common practice in the gradient-based optimization where the algorithm is powerfull in local optimization and in order to avoid stuck in local exterme points, "multi-start" runs are performed in order to search the global point in the objective function.

To further continue thiss dicussion on the effect of initialization on the final solution, the u^* value for each repeatation has been show on the left side of Figure 14. Where the u^* is the vector of 8 dimention, each value shows the optimum injection rate for the 10 years life cycle of the field, in m^3/D. We woul like to note that the y axis was plotted from the range of 5 to 100. The reason for this is to show that in this optimization problem, injection of each wells can take any number between $5m^3/D$ to 100 m^3/D, and the y axis shows the full extend of the value optimum zation worlkflow can reach. Visually, looking at the left plot at Figure 14, we can see that the final solution of three repeations at each weels, does not differ significantly from each other . With small exception of (injection #2), it seems all the final solutions converges to the same solution. This feature that can be loosly said as "robustness" of optimization workflow to initial design is very helpfu, from this sense that we do not neeed to resetart the optimization with different initilaization, since they all will converges to the similar solution. From this perspective, authours can infere that BayesOpt workflow can be considered as "global" optimization method, as it shows the workflow avoids stuck in local exterme pointsor saddle regions. The plot on the left side of Figure 14 shows that mean injection rate (mean of three repeations) and erro bar at each injection wells. The bottom of error bar in this plot shows the $mean - sd$ and top of bar is $mean + sd$. As we can see that we do not see significant variation in the final solution in each repeations, also the plots recoomnds that in the case of repeating the optimization with more than three times (like 10 or 100), it can lead to lower variation in final solution.

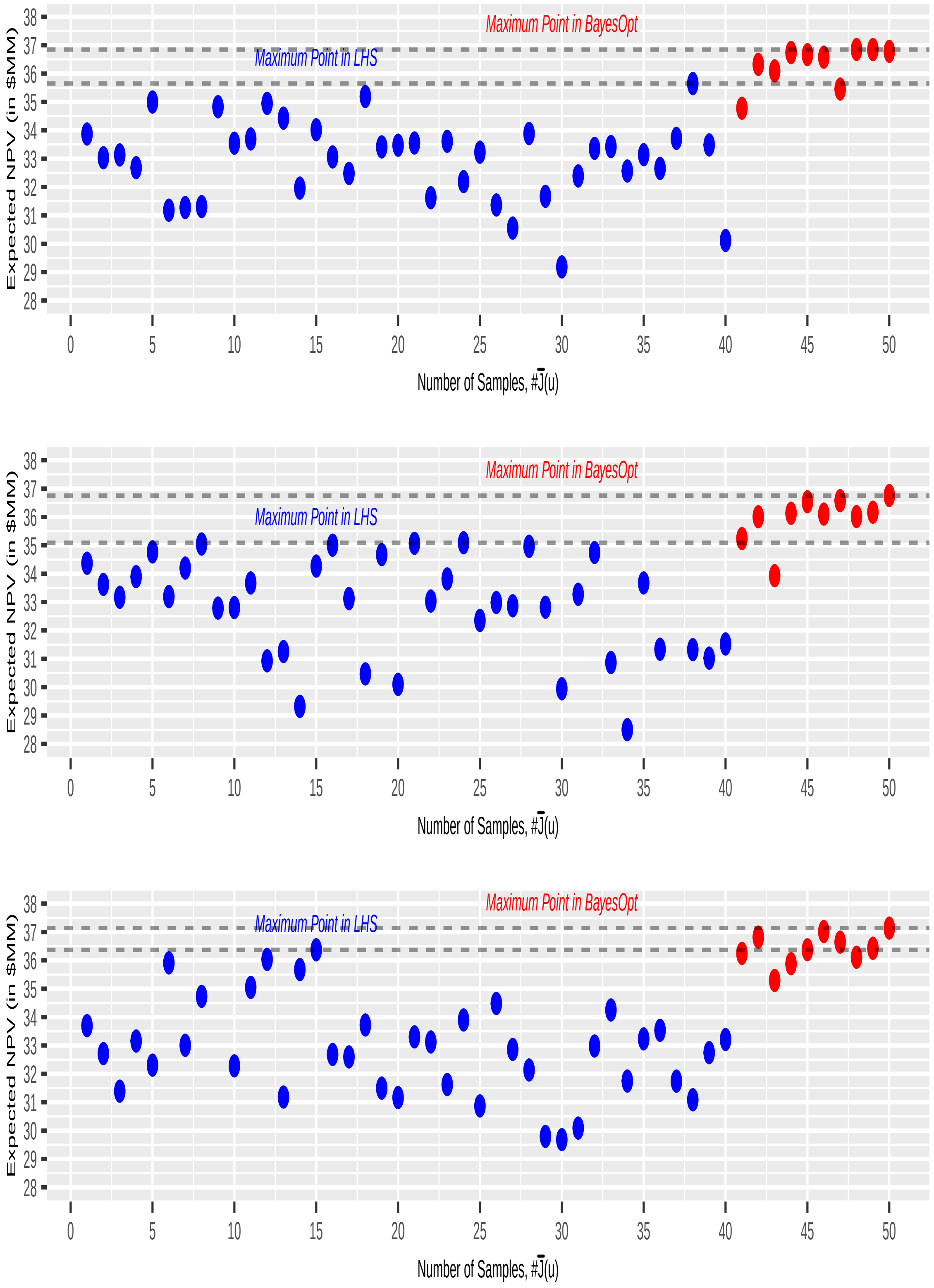


Figure 13: BayesOpt workflow applied to Syntetic 3D model, in three different initialization

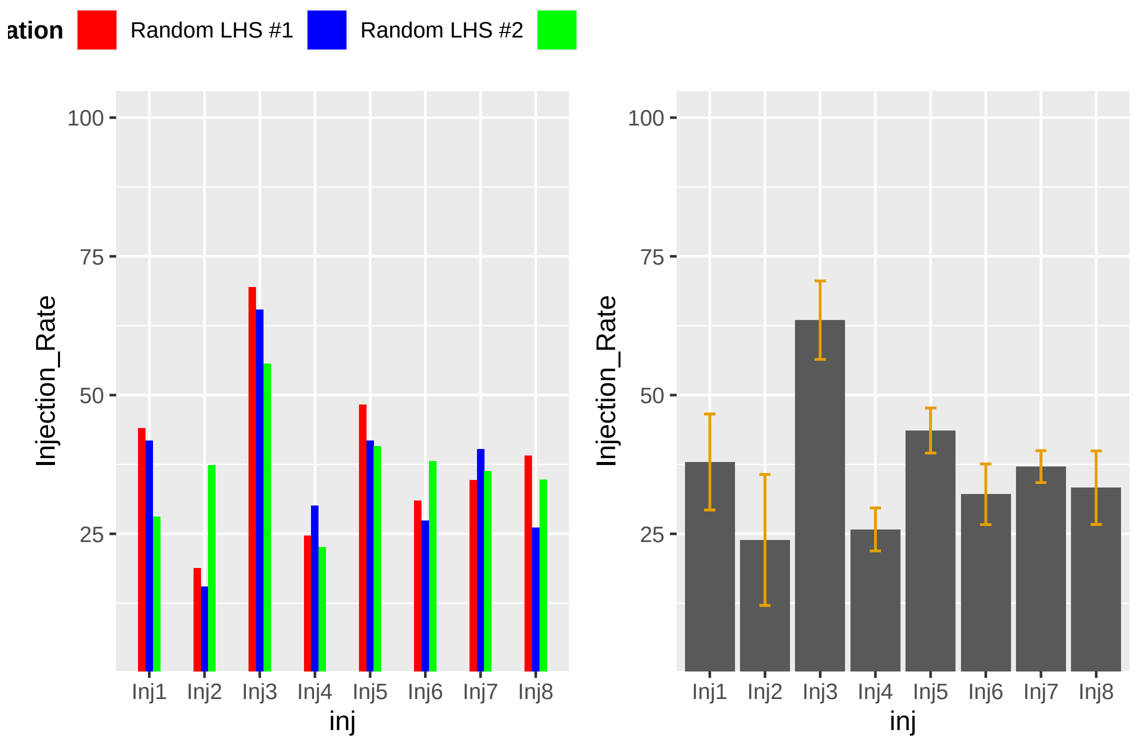


Figure 14: Left: final solution of optimization algorithm in three different initialization, Right: Mean and error bar of each injection rate at each injection wells

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:

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

where $i = 1, 2, ..., P$ and $j = 1, 2, ..., n$. 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, $gbestj$. 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)

$$X_{ij}^{t+1} = X_{ij}^t + V_{ij}^{t+1} \tag{27}$$

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 $\theta_1^{(0)}; \theta_2^{(0)}, \cdots, \theta_p^{(0)}$, (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 $\theta_1^{(k+1)}; \theta_2^{(k+1)}, \cdots, \theta_p^{(k+1)}$ 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 $\overline{J}(u)$. It means that optimization process could continue, until they use $\overline{J}(u) = 50$ function evaluations. It is worth to mention that in fact $\overline{J}(u) = 50$ is equal to 500 reservoir simulations, due to number of realization, $n_e = 10$ and ten computation per each $\overline{J}(u)$. 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 15 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 15 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. Morever, the Figure shows that in BayesOpt method, number of $\overline{J}(u)$ 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 $\overline{J}(u)$ grows as $n_{\text{popsize}} \times iteation$. As 15 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 16. In Figure 16 we further allowed the number of $\overline{J}(u)$ evaluations to 250, while keep the results of BayesOpt to the 50. Meaning that PSO and GA algorithm

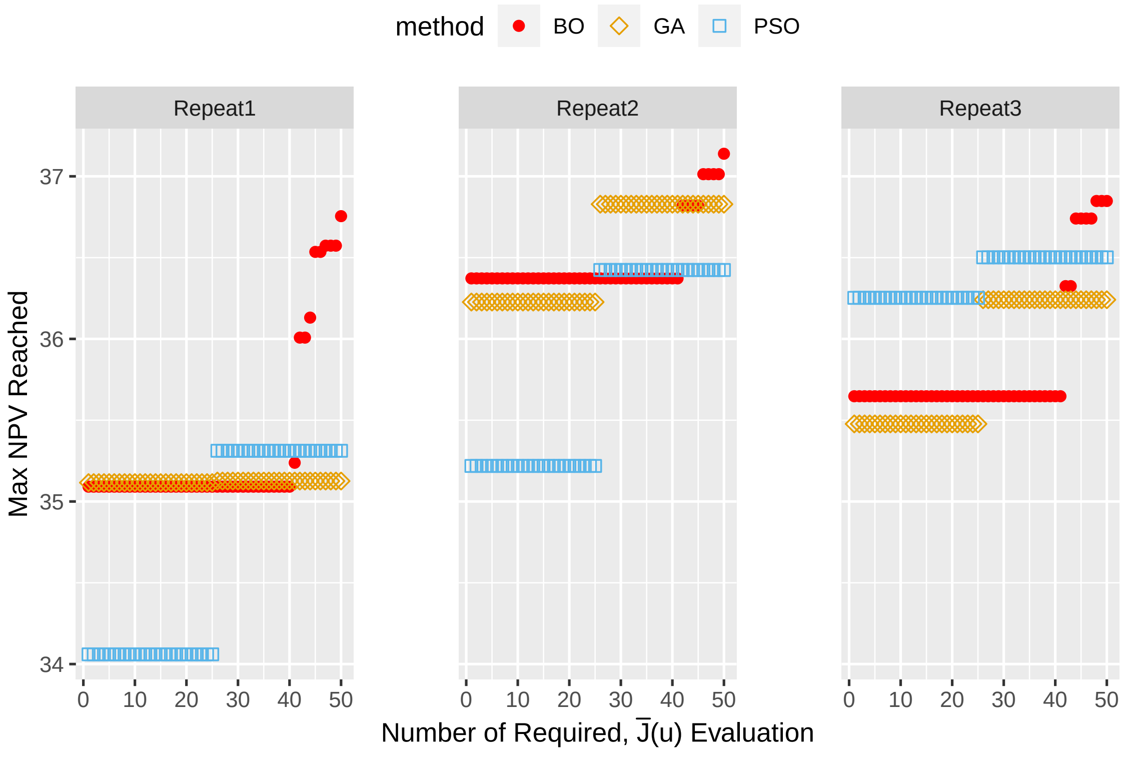


Figure 15: Comparison of GA, PSO and BayesOpt performance at fixed function evaluation budget

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

Optimization Method	Maximum Achieved NPV (median)	$\overline{J}(u)$ 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 16 does not convey a single message about performance of these methods. In **??** median value of three algorithm was compared. The value in second column of **??** is mean value of each optimization method. (In three repetitions, the maximum achieved NPV is a<b<c, b was selected). As the **??** 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 $\overline{J}(u)$ 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)

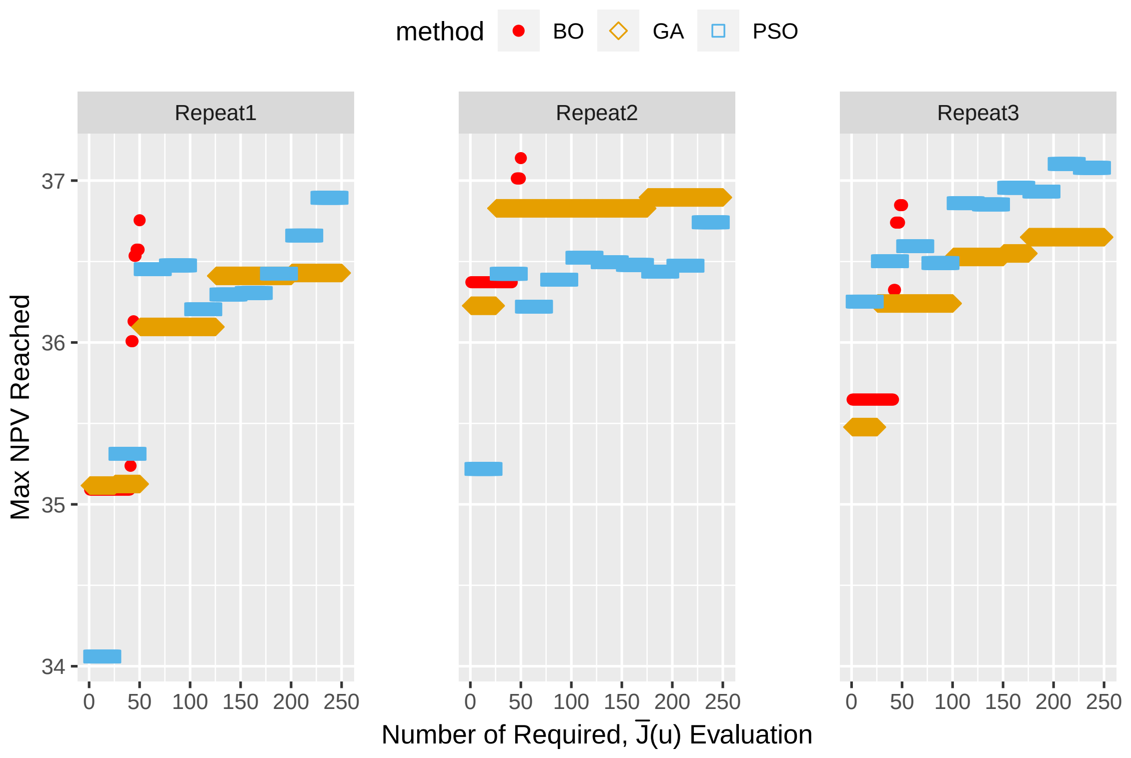
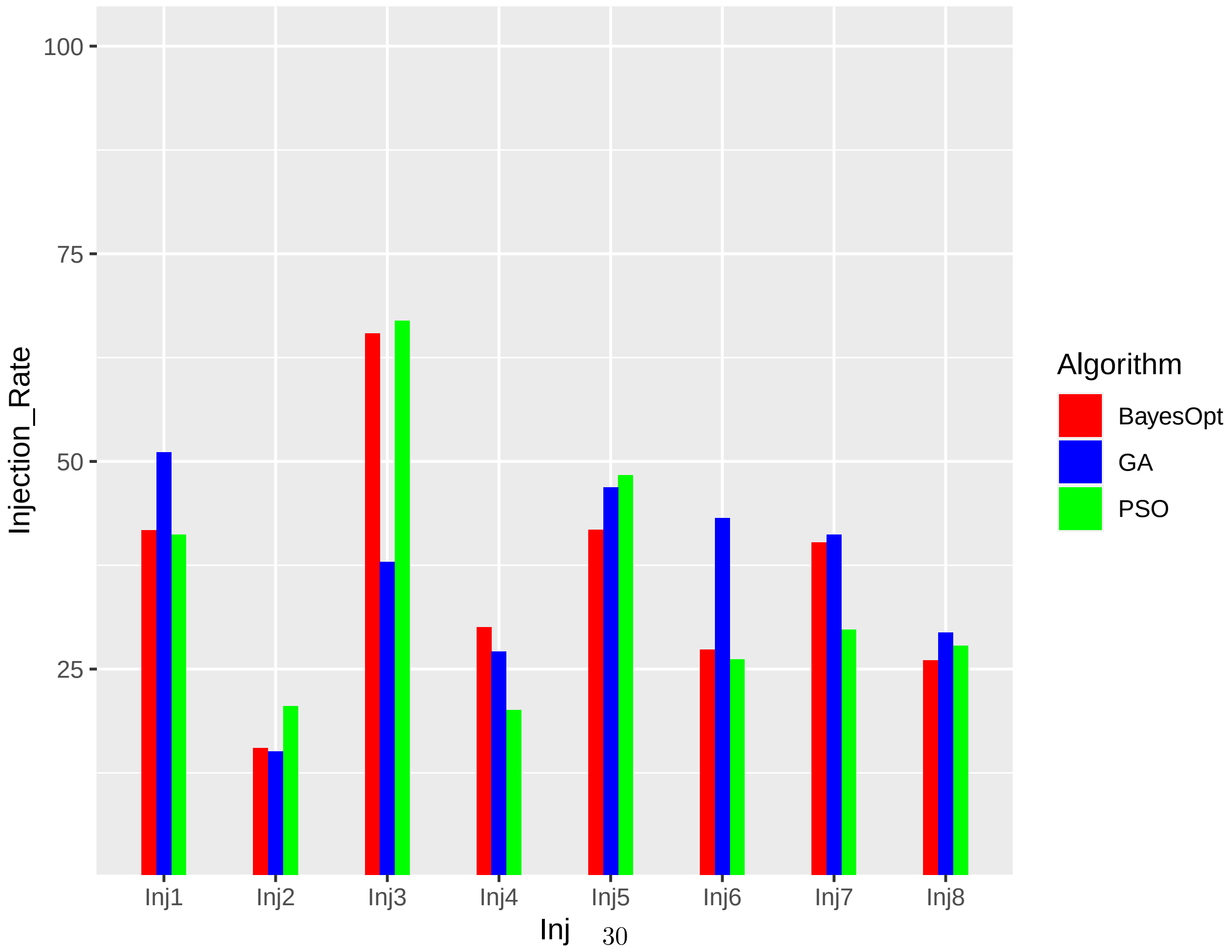


Figure 16: Comparison of GA PSO and BayesOpt performance at fixed function evaluation budget



377

6. Conclusing Remarks

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

Albertoni, A., Lake, L.W., 2003. Inferring Interwell Connectivity Only From Well-Rate Fluctuations in Waterfloods. SPE Reservoir Evaluation & Engineering 6, 6–16. doi:10.2118/83381-PA

Almeida, B.S.G. de, Leite, V.C., 2019. Particle Swarm Optimization: A Powerful Technique for Solving Engineering Problems. IntechOpen.

Bruce, W.A., 1943. An Electrical Device for Analyzing Oil-reservoir Behavior. Transactions of the AIME 151, 112–124. doi:10.2118/943112-G

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

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

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.

Fedutenko, E., Yang, C., Card, C., Nghiem, L.X., 2014. SPE Heavy Oil Conference-Canada. SPE, Calgary, Alberta, Canada, p. D021S008R001. doi:10.2118/170085-MS

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.

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

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.

Mitchell, M., 1998. An introduction to genetic algorithms. MIT press.

Mohaghegh, S.D., Guruswamy, S., 2006. Development of Surrogate Reservoir Models (SRM) for Fast-Track Analysis of Complex Reservoirs. San Antonio, Texas, U.S.A, p. 9.

Murphy, K.P., 2022. Probabilistic machine learning: An introduction. MIT Press.

Nwachukwu, A., Jeong, H., Sun, A., Pyrcz, M., Lake, L.W., 2018. SPE Improved Oil Recovery Conference. OnePetro. doi:10.2118/190239-MS

Sampaio, T.P., 2009. An Application of Feed Forward Neural Network as Nonlinear Proxies for the Use During the History Matching Phase 11.

Sayarpour, M., 2008. Development and application of capacitance-resistive models to water/CO? floods (PhD thesis).

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

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

Yousef, A.A., 2006. A Capacitance Model To Infer Interwell Connectivity From Production- and Injection-Rate Fluctuations 17.

Zhao, H., Kang, Z., Zhang, X., Sun, H., Cao, L., Reynolds, A.C., 2015. SPE Reservoir Simulation Symposium. Society of Petroleum Engineers, Houston, Texas, USA. doi:10.2118/173213-MS