# Optimization of Well Placement in Carbon Capture and Storage (CCS): Bayesian Optimization Framework under Permutation Invariance

**Sofianos Panagiotis Fotias** [1], **Ismail Ismail** [1] **and Vassilis Gaganis** [1,2,*]

1 School of Mining and Metallurgical Engineering, National Technical University of Athens, 15773 Athens, Greece; sfotias@metal.ntua.gr (S.P.F.); i.ismail@herema.gr (I.I.)
2 Institute of Geoenergy, Foundation for Research and Technology, 73100 Chania, Greece
* Correspondence: vgaganis@metal.ntua.gr

**Abstract:** Carbon Capture and Storage (CCS) stands as a pivotal technological stride toward a sustainable future, with the practice of injecting supercritical $CO_2$ into subsurface formations being already an established practice for enhanced oil recovery operations. The overarching objective of CCS is to protract the operational viability and sustainability of platforms and oilfields, thereby facilitating a seamless transition towards sustainable practices. This study introduces a comprehensive framework for optimizing well placement in CCS operations, employing a derivative-free method known as Bayesian Optimization. The development plan is tailored for scenarios featuring aquifers devoid of flow boundaries, incorporating production wells tasked with controlling pressure buildup and injection wells dedicated to $CO_2$ sequestration. Notably, the wells operate under group control, signifying predefined injection and production targets and constraints that must be adhered to throughout the project's lifespan. As a result, the objective function remains invariant under specific permutations of the well locations. Our investigation delves into the efficacy of Bayesian Optimization under the introduced permutation invariance. The results reveal that it demonstrates critical efficiency in handling the optimization task extremely fast. In essence, this study advocates for the efficacy of Bayesian Optimization in the context of optimizing well placement for CCS operations, emphasizing its potential as a preferred methodology for enhancing sustainability in the energy sector.

**Keywords:** CCS; machine learning; Bayesian optimization; Gaussian process; regression; predictive modeling; well placement optimization

## 1. Introduction

In recent years, the discourse surrounding climate change has intensified, with a growing consensus on the imperative need to transition the global economy towards achieving net-zero greenhouse gas emissions by mid-century. This ambitious goal, essential to averting dangerous anthropogenic interference with the climate system, requires rapid investment increase in near-zero emission technologies across all sectors [1]. Among these technologies, Carbon Capture and Storage (CCS), also referred to as CCUS, where "U" stands for utilization, shows to be a pivotal solution to reducing carbon emissions, addressing climate change, and facilitating the transition to a carbon-neutral future [2,3]. Its significance lies in its potential to address emissions from various sources, including power-generating plants, hard-to-abate industries (such as cement and refineries), and hydrogen production. CCUS is the only technology capable of guiding industries to absolute net neutrality by capturing and treating emissions.

Once $CO_2$ is captured, it undergoes compression and transport to be permanently stored in geological formations, acting as a "sink" through injection. These geological formations must possess certain key characteristics [4]. Essential conditions for a formation to be considered a suitable carbon storage site include the formation of a pore unit with

sufficient capacity to store the intended $CO_2$ volume; pore interconnectivity allowing injection at the required rate; and the presence of an extensive cap rock or barrier at the top of the formation to retain the injected $CO_2$, prevent unwanted migration, and ensure long-term containment [5]. Saline water-bearing formations [6–8] are widely regarded as the most mature options for $CO_2$ storage, as demonstrated by numerous projects worldwide. Additionally, depleted oil and gas fields have recently emerged as potential storage sites, benefitting from operational experience gained through $CO_2$-EOR operations [9–11].

Deploying CCUS technology on a large scale in saline aquifers or hydrocarbon fields necessitates accurate characterization of the storage site and reservoir, involving a complex process, from screening to deep geological and geophysical characterization [12]. Numerical simulation and modeling using computational fluid dynamics (CFD), possibly coupled with reactive transport simulations, are employed to address complex subsurface geology, simulating the flow behavior of the injected and the in situ fluids. The goal of such models is to estimate the field sequestration capacity of the injected $CO_2$ under different trapping mechanisms, such as structural, dissolution, residual, and mineral storage, during all project phases [13,14].

Reservoir simulation is a crucial tool in the field of petroleum engineering that aids in modeling and predicting fluid flow behavior within subsurface reservoirs. The primary objective is to simulate the complex interactions among various components, such as rock, fluids, and wells. One of the fundamental principles underlying reservoir simulation is Darcy's law, which describes the flow of fluids through porous media. The Darcy equation (Equation (1)) relates fluid velocity to pressure gradient, permeability, and fluid viscosity [15]. The simulator employs numerical methods to solve the mass, momentum, and energy differential equations governing fluid flow (2) in the porous medium. The most widely used flow model in reservoir simulation is the black-oil one [16]. In the black-oil formulation [17], the model is considered to consist of three separate fluid phases or pseudocomponents (water, oil, and gas). Mixing among phases is allowed, and the amount of how much each phase is dissolved in the others and how it affects flow properties need to be kept tracked of.

$$\mathbf{v}_a = -\lambda_a \mathbf{K}(\nabla p_a - \rho_a \mathbf{g}) \tag{1}$$

$$\frac{\partial}{\partial t}(\phi A_a) + \nabla \mathbf{u}_a + q_a = 0 \tag{2}$$

where the subscript $a$ denotes the pseudocomponent oil (o), gas (g), or water (w). For each one, the variables are defined as follows: $\mathbf{v}_a$ is the phase flux, and $\lambda_a$ is the mobility of the phase. This is defined as $\lambda_a = \frac{k_{r,a}}{\mu_a}$, where $\mu_a$ and $k_{r,a}$ are the viscosity and the relative permeability of each phase, respectively. In short, relative permeability is a measure of the effective permeability reduction of a phase in the presence of other phases and is a function of saturation. $p_a$ is the pressure, $\rho_a$ is the density, and $q_a$ is the well outflux/influx of each phase, i.e., the volumetric flow rate at which each pseudocomponent is produced/injected from the system. $A_a = f(b_w, b_o, b_g, s_w, s_o, s_g)$ are accumulation terms dependent on phase saturation ($s_a$) and shrinkage phase factors ($b_a$). The flux terms $\mathbf{u}_a = f(\mathbf{v}_w, \mathbf{v}_o, \mathbf{v}_g, b_w, b_o, b_g)$ are known functions of shrinkage factor and phase fluxes. Finally, the terms $\phi$ and $\mathbf{K}$ are the porosity and permeability, respectively.

The other popular simulation model is the compositional one [18,19]. The difference between the black-oil and compositional models lies in that the former traces the flow of each separate phase; hence, it only considers single-phase (gas, oil, and water) density and compressibility effects, as well as dissolution of gas in the oil phase. The fact that the volumetric and dissolution factors can be easily described by simple functions of the pressure is based on the assumption that the overall composition does not change significantly during reservoir depletion or waterflooding. On the other hand, compositional modeling, facilitated by means of an EoS model, is used when some "alien" composition is

incorporated in the reservoir fluids. This is the typical case when handling EOR cases of $CO_2$, natural or acid gas injection, and injection of dry gas in gas condensate reservoirs.

Equations (1) and (2) are combined in a second-order partial differential equation (PDE) which can be analytically solved under simplifying assumptions (incompressible, single-phase system with constant porosity and permeability). However, when trying to model realistic subsurface formations, these assumptions do not hold. In this case, the discretization of the equation through linearization is a key step in numerical reservoir simulation. This process involves dividing the reservoir into a grid to represent the spatial distribution of rock properties and fluid flow and using a linearization scheme [20]. Common methods for discretization include finite differences, finite volumes, and finite elements. Finite volumes [21], in particular, are widely used in reservoir simulation due to their simplicity and efficiency. Gridding plays a vital role in the discretization process, involving defining the size and shape of the cells within the reservoir grid. Various types of grids, such as Cartesian, corner-point, and unstructured grids, are used based on the geological complexity of the reservoir. The choice of grid impacts the accuracy and computational efficiency of the simulation.

Petrophysical and fluid properties, such as porosity, permeability, fluid saturation, compressibility, and relative permeability, are crucial inputs for reservoir simulation [22,23]. Accurate characterization of these properties is essential for realistic simulation results. Furthermore, incorporating phase behavior models is essential to accurately representing the complex interactions among different phases of fluids and dissolved particles in multiphase flow simulations. For instance, in CCS simulations, it is crucial to capture the dynamics of $CO_2$ dissolution, which significantly depends on the concentration of salts in the aquifer [24].

Building upon the foundation of accurate simulation models, the optimization of well placement becomes paramount to maximizing the sequestration potential of the aquifer in CCS endeavors [25,26]. The primary objective of CCS projects is to maximize $CO_2$ storage while adhering to constraints and technical policies [27,28]. In its most commonly employed form, the CCS plan requires that brine producers be closed off once significant $CO_2$ breakthrough occurs, whereas the injectors rate is reduced when the bottomhole pressure reaches the safety limit. Consequently, optimal well placement significantly influences the overall success of the sequestration project by delaying the breakthrough time. The objective function maximization in this context is defined as achieving the highest attainable $CO_2$ storage, considering the spatial distribution of the injection and brine withdrawal wells. This multi-dimensional function has an unknown functional form and its point values are expensive to compute since each one requires a fully implicit simulation to be run. Therefore, it needs to be treated as a black-box function.

Well placement decisions cannot rely solely on static properties, since geological variables are mostly non-linearly correlated to production performance. This is demonstrated in Figure 1, where the objective function of oil production in the form of the expected net present value (NPV) is evaluated for the placement of a single oil producer in each cell. The complexity and non-linearity in CCS projects parallel those encountered in oil production, underscoring the importance of optimized well placement for maximizing storage efficiency while considering factors such as $CO_2$ dissolution dynamics and aquifer characteristics. In general, well placement optimization poses a formidable challenge due to the large number of the decision variables involved, the high non-linearity of the system response, the abundance of local optima, and the well placement constraints. Due to the discontinuities and the inherent non-smoothness of the objective, research has been mostly focused on derivative-free optimization methods such as Genetic Algorithms [29] and Particle Swarm Optimization [30] combined with various heuristics.

Guyaguler et al. (2002) [31] developed and applied the Hybrid Genetic Algorithm (HGA) [32] to determine the optimal location and number of vertical wells, while Badru et al. (2003) [33] extended this work to also include horizontal wells. In essence, HGA comprises Genetic Algorithm (GA), the polytope algorithm for selecting the fittest indi-

viduals and a kriging proxy to expedite the optimization process. The results showed that after a few simulation runs, the method was able to optimize the production's NPV compared with the base case. In another work, Emerick et al. (2009) [34] designed a tool for well placement optimization using a GA with non-linear constraints. They applied the GENOCOP III algorithm [35] which utilizes two separate populations where the evolution in one influences the evaluation of the other. The first population, named the "search population", consists of individuals that satisfy the linear constraints, whereas the second, called the "reference population", consists only of those that satisfy both linear and non-linear constraints. By using this procedure, the research team developed a tool that handled a total of eight decision variables per well (six in total that define the heel and toe coordinates of the well, one binary for producer or injector, and one defining active or inactive well). The results showed a significant increase in the NPV of the production plan in comparison with the base case proposed plans. Stopa et al. (2016) [36] applied both GA and PSO to optimize well placement and well rates. Their approach was different in that they utilized an objective function that measures the ratio of the volume of mobile $CO_2$ phase (plume) over the volume of $CO_2$ injected after 10 years of injection and 80 years of monitoring. Their goal was to minimize this ratio, since by minimizing the plume, leakage risk is also reduced. In essence, they tried to find configurations that maximized the residual and solubility trapping of $CO_2$. In their work, PSO was able to find the optimal value of 12.5% in 500 simulation runs, while GA needed more than 850. Similarly, Goda and Sato (2013) [37] utilized a similar objective function by using a global sampling method known as Latin Hypercube Sampling (LHS) [38]. Notably, in their work, they enhanced the definition of immobile $CO_2$ by taking into account the irreducibility of relative permeability curves. Besides the mentioned landmark works, there have been dozens more published on this optimization problem. Islam et al. (2020) [39] provided a detailed review of research conducted on the well placement optimization problem, focusing mostly on derivative-free and machine learning regression methods.

Bayesian Optimization (BO) [40] is a global optimization method that efficiently constructs and updates surrogate models representing black-box objective functions, rendering it suitable for addressing the well placement optimization problem. In this work, our goal is to apply this method to optimize the carbon sequestration capabilities of a saline aquifer. BO represents a cutting-edge methodology that dynamically balances the exploration of the solution space with the exploitation of promising regions to efficiently discover well placement that can lead to the global optimum configuration. BO relies on constructing a probabilistic surrogate model, often a Gaussian Process (GP), to approximate the underlying objective function, which, in this context, measures total carbon storage. This surrogate model not only captures the observed data but also quantifies the uncertainty associated with predictions, allowing for the incorporation of prior knowledge and continuous refinement as new information becomes available through successive simulation runs. By iteratively selecting well locations based on the surrogate model's predictions and subsequently updating the model with the actual reservoir response, BO intelligently adapts its search strategy. This adaptive nature makes it particularly well suited for navigating the high-dimensional and uncertain parameter space, providing a systematic and data-driven means of identifying optimal well configurations for enhanced carbon sequestration.

BO for well placement optimization was implemented in the work of Balabaeva et al. (2020) [41]. In their work, the researchers attempted to optimize well locations by maximizing the cumulative drainage area of wells by using various derivative-free methods, with BO being one of them. In another work, Bordas et al. (2020) [42] utilized a BO framework for field development planning while quantifying the effect of geological uncertainty. They generated a surrogate model that was able to identify well placement positions that maximized NPV while accounting for uncertainty. Kumar [43,44] also presented works on novel BO realizations, such as trust region BO [45] and sparse-axis-aligned subspaces BO (SAASBO) [46]. BO has additionally been investigated for optimizing various decision variables crucial in reservoir exploitation besides well placement. Lu et al.

(2022) [47] utilized BO to optimize the time cycle of gas/surfactant injection in Water Alternating Gas (WAG) schemes. Their study conducted at the Cranfield reservoir site resulted in a 16% increase in gas storage volume and a 56% reduction in water/surfactant usage compared with the baseline WAG schedule. In an MSc thesis, Javed [48] employed a BO framework to optimize deviated well trajectories in a synthetic reservoir model located in the North Sea. The optimized trajectories led to a 28% enhancement in the objective function compared with the base case. Additionally, Wang and Chen (2017) [49] investigated the Cardium tight formation, where they successfully optimized fracturing design parameters by using BO, achieving a notable 55% improvement in the objective function. These studies prove BO to be a suitable framework in reservoir engineering. However, to our knowledge, BO has not been studied for the well placement optimization problem in CCS specifically.

The rest of the paper is organized as follows: In Section 2, we delve into a comprehensive explanation of the Bayesian Optimization method. Following that, Section 3 offers a detailed description of the specific problem, elucidating how BO was strategically applied to address the challenges encountered, such as non-linear constraints and permutation invariance in the input space. In Section 4, we present results stemming from various implementations of the BO method, comparing the performance of various acquisition functions and inspecting the exploration–exploitation trade-off. Section 5 initiates an expansive discussion of the obtained results. Finally, Section 6 draws insightful conclusions based on the findings and discussions presented throughout the study.
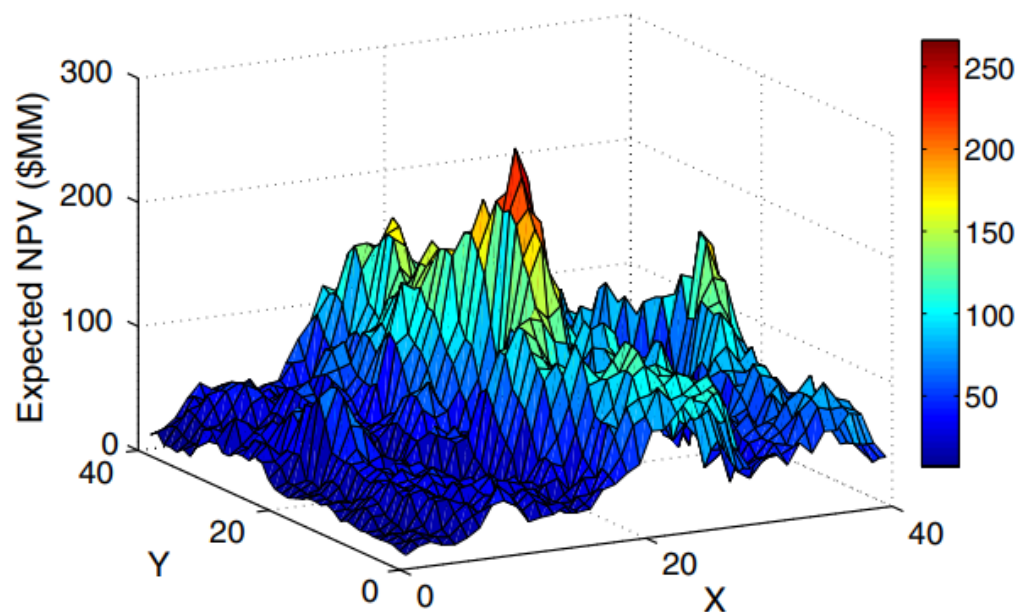
**Figure 1.** NPV evaluation for oil production with a single well [50].

## 2. Bayesian Optimization

### 2.1. Gaussian Process (GP) Regression

In order to properly understand BO, the GP framework needs to be explained in detail. The complete treatment of GP can be found in Rasmussen and Williams (2006) [51]. Most of the material in this section has been inspired by Frazier (2018) [40] and Dou (2015) [52] and the articles by Distill [53,54], which provide visual and interactive explanations of the materials discussed here.

Multivariate Gaussian distributions (MVN) [55] act as the building blocks of GP. They express the distribution of multiple random variables that are distributed normally and whose joint distribution is also Gaussian. They are defined by a mean vector **μ**, which describes the expected value of the distribution, and a positive semidefinite covariance

matrix $\Sigma$, which models the variance along each dimension and determines how the different random variables are correlated. The diagonal elements of $\Sigma$ express the variance of each random variable. The off-diagonal elements $\Sigma_{i,j}$ describe the correlation between the $i$th and $j$th random variables. The probability density of an MVN in $d$ dimensions is shown in Equation (3).

$$N(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right] \tag{3}$$

where $\boldsymbol{\mu}$ is the $d \times 1$ mean vector and $\boldsymbol{\Sigma}$ is the $d \times d$ covariance matrix. A GP is the extension of this MVN to the space of functions, defining a probability measure on a function space. This function space acts as the probabilistic surrogate objective function, which will eventually converge to the actual objective function through sampling and utilization of the GP regression algorithm. At first, the probability measure is interpreted as our prior knowledge about the unknown objective function before data are seen. A GP is defined via its mean and covariance just like the MVN; however, those now represent functions. Given a finite collection of $q$ points $\mathbf{x}_{1:q} \in \mathbb{R}^d$ that appropriately discretize the $d$-dimensional input space, the objective function's values $\mathbf{f}(\mathbf{x}_{1:q})$ are assumed to be drawn by an MVN.

$$\mathbf{f}(\mathbf{x}_{1:q}) \sim N(\mathbf{m}(\mathbf{x}_{1:q}), \mathbf{K}(\mathbf{x}_{1:q}, \mathbf{x}_{1:q})) \tag{4}$$

Compared with a multivariate normal, we have the following:

- A random function vector $\mathbf{f}(\cdot)$ instead of a random vector $\mathbf{x}$;
- A mean function vector $\mathbf{m}(\cdot)$ instead of a mean vector $\boldsymbol{\mu}$;
- A covariance (kernel) function matrix $\mathbf{K}(\cdot, \cdot)$ instead of a covariance matrix $\boldsymbol{\Sigma}$.

Above, $\mathbf{K}(\cdot, \cdot) = \{k(\mathbf{x}_i, \mathbf{x}_j)\}$ and $k(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function discussed below.

Equation (4) states that the vector of functions conditional on inputs $\mathbf{x}_{1:q}$ and the mean and covariance functions follows a $q$-dimensional MVN distribution. To avoid confusion, it needs to be stated that no assumption has been made on the distribution of the inputs. So far, the prior Gaussian distribution of the objective function has been defined. Function values selected in the prior are commonly referred to in the literature as the test data.

In the prior distribution, it is common to set the mean function to a constant value of zero. This is a reasonable assumption since no prior knowledge is assumed. If prior knowledge can be assumed to exist, then the mean function can be set to Equation (5).

$$m(\mathbf{x}) = \mu_0 + \sum_{i=1}^{p}\beta_i\Psi_i(\mathbf{x}) \tag{5}$$

where $\Psi_i$ are often parametric, low-order polynomials which summarize our prior knowledge on the objective function.

When it comes to choosing a kernel function, it is typically necessary to select it based on certain properties. First of all, kernel functions need to be positive semidefinite. Secondly, it is important that points closer in the input space are more strongly correlated, i.e., if $||\mathbf{x} - \mathbf{x}'|| < ||\mathbf{x} - \mathbf{x}''||$, then $k(\mathbf{x}, \mathbf{x}') > k(\mathbf{x}, \mathbf{x}'')$. There are many kernel functions commonly used in BO; however, the Matérn kernel [56] will be the main focus, since it is a more general form of kernel functions.

$$k(\mathbf{x}, \mathbf{x}') = \alpha_0\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}||\mathbf{x} - \mathbf{x}'||\right)^{\nu}K_\nu(\sqrt{2\nu}||\mathbf{x} - \mathbf{x}'||) \tag{6}$$

where $K_\nu$ is the modified Bessel function [57], $\Gamma(\nu)$ is the Gamma function, and $\alpha_0$ is a hyperparamter [58]. Certain values of $\nu$ in Equation (6) reduce the equation to other popular kernels, including the following:

- Radial basis function (RBF) kernel [59] for $\nu \to \inf$;
- Absolute exponential kernel for $\nu = 0.5$;

- Once-differentiable functions for $\nu = 1.5$;
- Twice-differentiable functions for $\nu = 2.5$.

So far, the prior distribution of functions has been constructed, but no observed information has been introduced to the model yet. To do so, the objective function's value has to be measured at some points, which are denoted by $\mathbf{y}_{1:n}$. These points, commonly referred to as training data, are observed by sampling the black-box objective function. By incorporating this information into our model, the posterior distribution can be defined. First of all, the distribution of $\mathbf{f}(\mathbf{x}_{1:q}, \mathbf{y}_{1:n})$ is now a $(q + n)$-dimensional MVN. To acquire the posterior distribution on a certain point of $\mathbf{x}_{1:q}$, conditioning and marginalization have to be applied. By using the Bayes rule, Equation (7) is obtained.

$$f(\mathbf{x})|\mathbf{f}(\mathbf{y}_{1:n}) \sim N(\mu_n(\mathbf{x}), \sigma_n^2(\mathbf{x})) \tag{7}$$

where the mean value $\mu_n(\mathbf{x})$ is given by Equation (8) and represents a weighted average between the prior and an estimate based on the training data.

$$\mu_n(\mathbf{x}) = \mathbf{k}(\mathbf{x}, \mathbf{y}_{1:n})\mathbf{K}(\mathbf{y}_{1:n}, \mathbf{y}_{1:n})^{-1}(\mathbf{f}(\mathbf{y}_{1:n}) - \boldsymbol{\mu}_0(\mathbf{y}_{1:n})) + \mu_0(\mathbf{x}) \tag{8}$$

and the posterior variance $\sigma_n^2(\mathbf{x})$ seen in Equation (9) is equal to the previous covariance, minus a term that corresponds to the variance removed by observing the training data.

$$\sigma_n^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x}, \mathbf{y}_{1:n})\mathbf{K}(\mathbf{y}_{1:n}, \mathbf{y}_{1:n})^{-1}\mathbf{k}(\mathbf{y}_{1:n}, \mathbf{x}) \tag{9}$$

This regression process updates the surrogate objective function's values by conditioning on the training data.

### 2.2. Kernel Functions

Kernel functions are essential in BO since they define the shape of the prior and posterior distributions. For a function to be a valid kernel, it needs to satisfy Mercer's theorem [60].

A symmetric function $k(\mathbf{x}, \mathbf{y})$ defined on $\mathcal{X} \times \mathcal{X}$, where $\mathcal{X}$ is some input space, is a valid kernel function if and only if it satisfies the following conditions:

1. Symmetry : $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$.
2. Positive semidefiniteness: For any finite set of points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathcal{X}$, where $i, j \in \{1, ..., n\}$, the corresponding kernel matrix $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)$ is positive semidefinite.

If there exists a mapping function $\phi(\mathbf{x})$ that maps $\mathbf{x}$ to a higher-dimensional space such that the inner product in that space is equivalent to the kernel function $k(\mathbf{x}, \mathbf{y}) = <\phi(\mathbf{x}), \phi(\mathbf{y})>$, then according to Mercer's theorem, $k(\mathbf{x}, \mathbf{y})$ is a valid kernel function. Kernel functions have important properties commonly used to modify and non-linearly combine existing kernels to obtain new ones better suited for handling specific problems, such as the following:

1. Linearity: If $k_1(\mathbf{x}, \mathbf{y})$ and $k_2(\mathbf{x}, \mathbf{y})$ are valid kernel functions, then for any constants $a, b \geq 0$, the linear combination $a \cdot k_1(\mathbf{x}, \mathbf{y}) + b \cdot k_2(\mathbf{x}, \mathbf{y})$ is also a valid kernel function.
2. Product: If $k_1(\mathbf{x}, \mathbf{y})$ and $k_2(\mathbf{x}, \mathbf{y})$ are valid kernel functions, then their product $k(\mathbf{x}, \mathbf{y}) = k_1(\mathbf{x}, \mathbf{y}) \cdot k_2(\mathbf{x}, \mathbf{y})$ is also a valid kernel function.
3. Exponential: If $k(\mathbf{x}, \mathbf{y})$ is a valid kernel function, then $\exp(k(\mathbf{x}, \mathbf{y}))$ is also a valid kernel function.
4. Function: If $k(\mathbf{x}, \mathbf{y})$ is a valid kernel and $f : \mathcal{X} \to \mathcal{R}$, then $g(\mathbf{x}, \mathbf{y}) = f(\mathbf{x})k(\mathbf{x}, \mathbf{y})f(\mathbf{y})$ is also a valid kernel.

### 2.3. Acquisition Functions

So far, the GP framework within which BO operates has been defined. This of course does not constitute a complete optimization algorithm since an iterative solution update rule is yet to be defined. This is where acquisition functions come into play. The main idea

is for BO to utilize the posterior GP to determine a new point of high expected value and make an additional valuable observation on that point. Subsequently, the posterior GP is conditioned against the new information, and the procedure is repeated until there is enough confidence that the algorithm has converged to the global optimum.

BO is specifically tailored to problems where data acquisition is considered computationally intractable. Therefore, BO's goal is to find good sequential acquisition policies which converge to the optimum in as few objective function evaluations as possible. These policies inherently include the exploration–exploitation trade-off. In BO, exploration is related to looking for the next iteration estimate in areas of the solution space where the variance is high, while exploitation is related to searching areas where the current maximum is located and trying to find even better solutions.

Given a posterior distribution based on $n$ observed function values, the goal is to find the most valuable point $\mathbf{x}_{n+1}$ for receiving the next objective function calculation. The proposed solution is founded on the concept of the value of information [61]. Let us assume that a hypothetical new observation at $\mathbf{x}$ has an objective function value of $y$. The value of making this observation is denoted by $v(\mathbf{x}, y; f(\mathbf{x}_{1:n}))$. By integrating $y$ out of the picture by using the point predictive distribution of the posterior GP, we can define the expected value of observing $\mathbf{x}$ (Equation (10)).

$$v(\mathbf{x}; f(\mathbf{x}_{1:n})) = \mathbb{E}_{y|\mathbf{x}; f(\mathbf{x}_{1:n})}[v(\mathbf{x}, y; f(\mathbf{x}_{1:n}))] \tag{10}$$

As a result, the data acquisition policy is defined by solving the optimization problem $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} v(\mathbf{x}; f(\mathbf{x}_{1:n}))$, i.e., the next trial point should be the one exhibiting the maximum expected value. A natural stopping criterion for such a policy is to stop when the maximum expected value of the current iteration is smaller than a threshold $v(\mathbf{x}_{n+1}; f(\mathbf{x}_{1:n})) < \epsilon$. There are many choices for the value function but the following three in particular are the most commonly utilized ones:

1. Probability of improvement (PI) : PI (Equation (11)) picks the point that is most likely to yield an improvement over the current maximum $\hat{y}_n$[62].

$$v(\mathbf{x}; f(\mathbf{x}_{1:n})) = PI(\mathbf{x}) = P[f(\mathbf{x}) \geq \hat{y}_n] = \Phi\left(\frac{\hat{y}_n - m_n(\mathbf{x})}{\sigma_n}\right) \tag{11}$$

   This is considered to be an exploitation-oriented policy.

2. Expected improvement (EI): EI (Equation (12)) involves computing how much improvement we expect, especially over a highly uncertain domain [63]. It picks the point with the highest expectation of difference between predicted function value and current maximum.

$$\begin{aligned} v(\mathbf{x}; f(\mathbf{x}_{1:n})) = EI(\mathbf{x}) &= \mathbb{E}[\max(0, y - \hat{y}_n)] = \\ &= \sigma_n(\mathbf{x})\varphi\left(\frac{m_n(\mathbf{x}) - \hat{y}_n}{\sigma_n(\mathbf{x})}\right) + (m_n(\mathbf{x}) - \hat{y}_n)\Phi\left(\frac{m_n(\mathbf{x}) - \hat{y}_n}{\sigma_n(\mathbf{x})}\right) \end{aligned} \tag{12}$$

   where $\varphi$ and $\Phi$ are the normal density and cumulative distribution.

3. Upper confidence bound (UCB): UCB (Equation (13)) is one of the simplest acquisition functions, with a tunable parameter ($\lambda$) designed to favor either exploration or exploitation.

$$v(\mathbf{x}; f(\mathbf{x}_{1:n})) = UCB(\mathbf{x}) = \mu(\mathbf{x}) + \lambda\sigma(\mathbf{x}) \tag{13}$$

## 3. Problem Description

### 3.1. Reservoir System

In this study, the sequestration potential of a deep and highly permeable synthetic aquifer (Figure 2) is investigated. The aquifer exhibits an inclined profile, with a pronounced steepness near the crest resembling an anticline, expanding horizontally across a wide area, and comprises four sedimentary layers, each one with varying average permeability

values. There is no apparent faulting or fracturing in the formation, and it is assumed to be sealed tightly by shales. The aquifer's salinity is about 65,000 ppm. Furthermore, it maintains a temperature profile consistent with typical thermal gradient expectations at the reservoir depth. The characteristics of the aquifer can be seen in Table 1. Due to its high permeability, the pressure distribution within the aquifer adheres to the pressure gradient, remaining isotropic on the x-y plane. The aquifer is considered to be slightly overpressurized, prompting the simultaneous production of brine during CCS operations to postpone reaching the pressure upper safety limit indicated by the technical and safety policies in place as 9000 psi. The significance of this approach is underscored by the zero-flow conditions at the aquifer's sides, as dictated by no-flow boundary conditions in the simulation model [64].
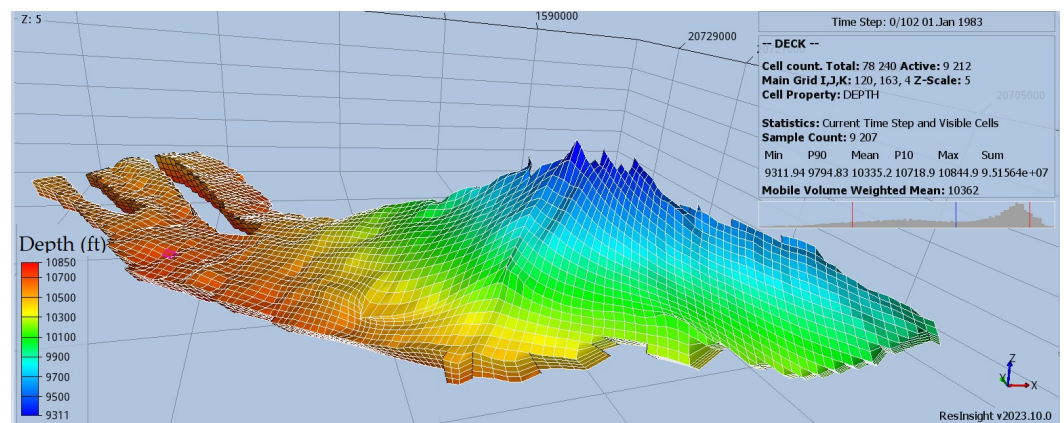
**Figure 2.** Depth value per cell in the aquifer. The z-axis is scaled by a factor of 5 for visualization purposes. The tool used is ResInsight, which is an open-source cross-platform 3D visualization, curve plotting, and post-processing tool for reservoir models and simulations.

**Table 1.** Aquifer characteristics.

| Parameter | Value | Unit |
|---|---|---|
| Average pressure ($P$) | 5000 | Psi |
| Temperature ($T$) | 200 | $^\circ$F |
| Porosity ($\phi$) | 0.25 | |
| Average depth ($D$) | 10,180 | ft |
| Average xy permeability ($k$) | 300 | mD |
| Bulk volume ($V$) | $2.4 \cdot 10^{11}$ | scf |
| Water in place | $1.7 \cdot 10^9$ | STB |

### 3.2. BO Framework

To formulate a BO framework, the objective function, its constraints, and the decision variables have to be defined. The target is to maximize the total amount of $CO_2$ sequestered in the aquifer after continuous injection for 40 years and concurrent brine production with constant target rates. This is handled by introducing two types of groups, injectors and producers, with each group consisting of a number of identical wells.

Constraints are set to the maximum and minimum bottomhole pressure thresholds, as well as to the maximum afforded breakthrough rate of $CO_2$ recycling. High non-linearity is introduced to the problem by the effect of the wells' location on the objective function and of the dynamic schedule changes occurring when operational constraints are violated. If, for example, an injector is placed close to a producer, then it is likely that $CO_2$ breakthrough will occur way sooner than otherwise. Subsequently, the producer will be closed off, and pressure will build up at a faster rate so that the constraint of maximum bottomhole pressure for the injector will be reached sooner, resulting in reduced sequestration. The depth at which each well is situated also plays a huge part in the breakthrough time. Since

$CO_2$ tends to migrate upwards, $CO_2$ travels faster to a producer situated near the crest. In contrast, if the producer is situated closer to the reservoir bottom, $CO_2$ breakthrough is delayed. The theoretical maximum amount of $CO_2$ stored is easily calculated as the target rate of the injectors over the timespan of injection. However, this maximum may be unobtainable due to the constraints already mentioned.

The well placement optimization problem can be formulated as shown in Equation (14).

$$
\begin{aligned}
\underset{\mathbf{x}}{\text{maximize}} \quad & f(\mathbf{x}) \\
\text{subject to} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \ldots, m
\end{aligned}
\tag{14}
$$

where $f(\mathbf{x})$ corresponds to the $CO_2$ mass sequestered, which can be sampled by simulation runs, and $g_i(\mathbf{x}) \leq 0$ are the scheduling constraints, where $\mathbf{x}$ represents the decision variables, which, in this case, are the coordinates $(x, y)$ of each vertical well. Most commercial and research reservoir simulators can embed scheduling constraints into their runs, thus ensuring that the simulation is operationally feasible. This way, the scheduling constraints are handled by the simulator and thus do not need to be explicitly handled by the optimizer. There are, however, additional constraints related to the feasibility of the input space, as discussed in the next subsection. For the optimization process, the python package Bayesian Optimization [65] was utilized.

### 3.3. Decision Variables and Constraints

Constraints need to be imposed on the input space to define the feasible region of the objective function. For a CCS system with $N$ wells, the input space consists of $2N$ decision variables and the wells' location may be written as in Equation (15).

$$
\mathbf{x} = (x_1, y_1, x_2, y_2, \ldots, x_N, y_N)
\tag{15}
$$

The first constraint to be imposed on the input space is that two wells cannot take up the same cells on the grid, since the simulator issues a simulation error instantly. As such, the optimizer is instructed to penalize such cases by giving a very negative value for the objective function when the simulator crashes.

The second and much more troublesome-to-handle constraint is the one regarding the valid position of each well. Since the aquifer's shape is not a square box, there are coordinate values within the grid-enclosing volume that correspond to null (non-existent) grid cells, as seen in Figure 3. The recommended solution would be to add a severe penalty term by introducing a large negative objective function value for inputs in the infeasible area. For a reservoir with an active-to-total-cell ratio of 50%, this ratio corresponds to the probability of a well being placed within the feasible area. If the location of a single well were sought, the optimizer would be able to map the infeasible area quite quickly. However, when N wells have to be placed rather than just one, the probability of selecting all of them inside the feasible space is only $(0.5)^N$. Clearly, the timeframe for the determination of the feasible area increases exponentially with the number of wells selected.
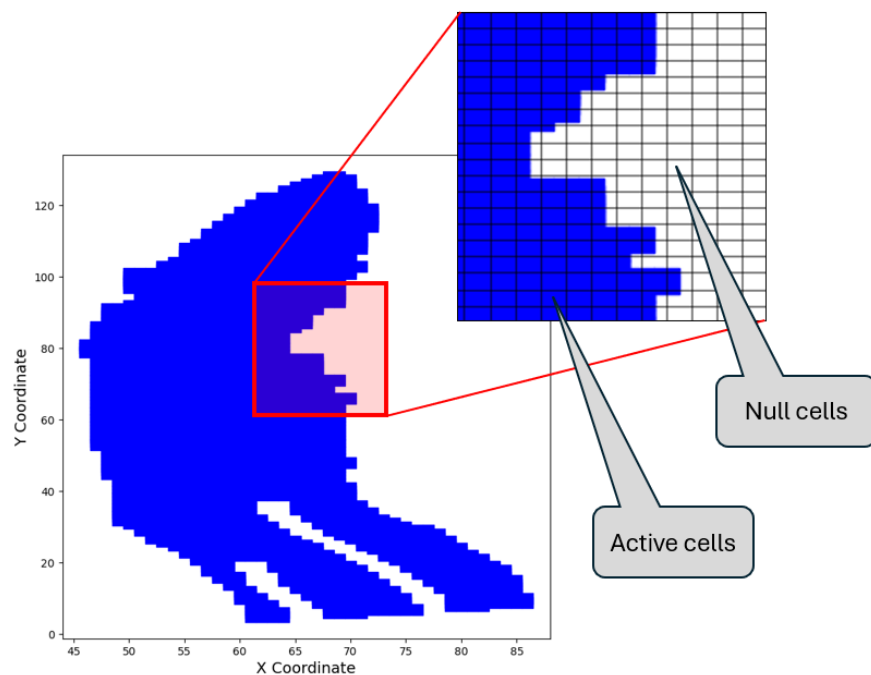
**Figure 3.** Feasible region for wells on reservoir grid top view.

To overcome this challenge, a mapping from this 2D input space to a 1D space is employed while maintaining a one-to-one relationship, using a transformation based on the cell distance from the origin. Subsequently, an ascending-order-sorting operation is applied to the Euclidean norms of each cell to arrange them. Algorithm 1 of the map can be seen below.

---

**Algorithm 1:** Mapping function $f : \mathbb{R}^2 \to \mathbb{R}$.

---

**Input:** Set of 2D cells $C = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
**Output:** Mapped one-dimensional values
Calculate Euclidean distance from the origin $(0,0)$ for each cell and store them in array $D$;
Sort array $D$ in ascending order;
Initialize $i = 1$;
**for** *d in D* **do**
    Assign $f(x_i, y_i) = i$;
    Increment $i$ by 1;
**end**

---

This mapping is one to one (1-1) because each point $(x, y) \in \mathbb{R}^2$ is uniquely associated with a real number $f(x, y) \in \mathbb{R}$. Note that this is not a homeomorphism, as it does not preserve topological properties such as closeness of points.

This mapping allows for dimensionality reduction from a $2N$-dimensional input space to an $N$-dimensional one. Furthermore, all values in the new input space represent the feasible region, so no more constraints on the input region need to be handled. The mapped values of each cell are shown in Figure 4.
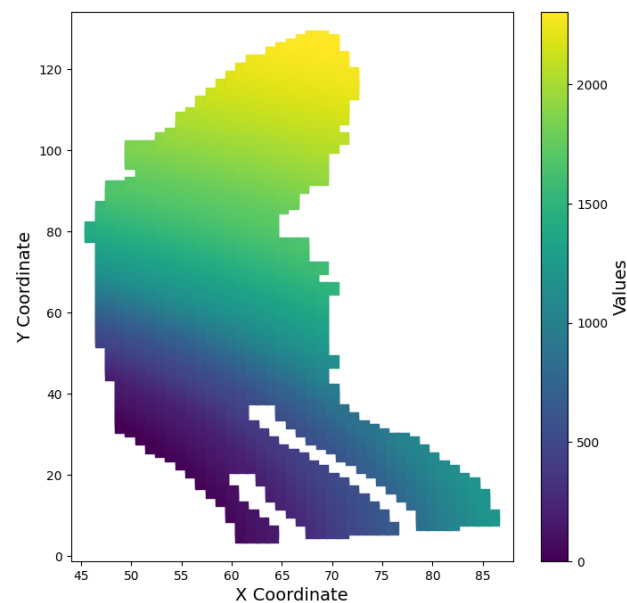
**Figure 4.** Mapped value of each cell.

### 3.4. Permutation Invariance Of Input Space

As mentioned earlier, the simulator splits wells into two groups, injectors and producers, managing them collectively under group control policies. Targets and constraints are set for each group, with an embedded optimizer in the simulator responsible for adjusting schedule parameters for individual wells within each group. Essentially, a group comprises wells of the same type, distinguished solely by their location on the grid. On the other hand, the wells' locations serve as input variables in the BO framework, ordered in a vector. Therefore, the simulator handles the wells' positions as two sets of unordered objects, partitioned on the respective group, while BO treats them as a vector. Consequently, for the simulator, the input space is invariant under permutations for well positions within a group, whereas for BO, it is not.

Let us consider a scenario following the mapping described in Algorithm 1. If variables $m$ and $n$ represent the positions of injectors and producers, respectively, the input space's invariance implies that a sampled input exerts the same effect on the simulator, thus also on the objective function, across $m! \cdot n!$ vectors, generated by all of the permutations of the inputs within the same group.

To illustrate this concept, let us consider a simple case with two wells of the same type placed on a $3 \times 3$ Cartesian grid (refer to Figure 5). In this scenario, the top-left cell is denoted as [1,1]. Moving right from [1,1] brings us to [1,2], while moving downwards from [1,1] leads to [2,1]. This structure mirrors the conventions of matrix elements, facilitating a straightforward understanding of the grid's layout. The coordinates for the left sample are $x = (f([2,1]), f([3,2]))$, and for the right sample, $x = (f([3,2]), f([2,1]))$, where $f$ is the mapping defined by Algorithm 1. Despite the differing configurations, the simulator produces identical outputs for both samples, since they belong to the same group, and the simulator handles them as a set, regardless of their order. Each sample has its counterpart, or "doppelganger". As the number of wells increases, the number of potential configurations grows exponentially, since for $N$ wells, each unordered configuration can result from $N!$ different orderings. This example highlights the fundamental principle of group control and the resultant invariance in the input space, offering insights into the system's behavior regardless of the specific arrangement of wells.
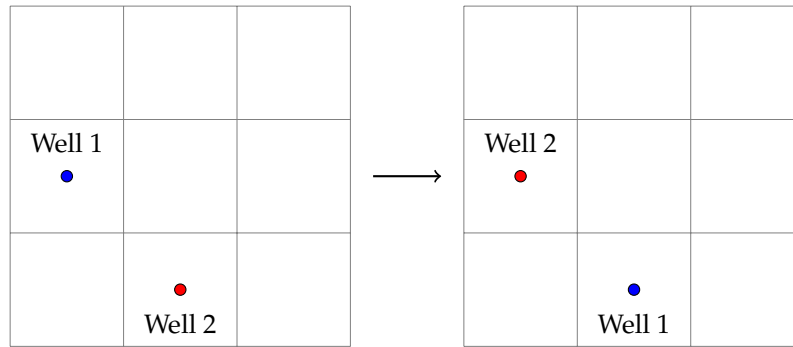
**Figure 5.** Both ordered configurations produce the same output.

### 3.5. Modified Kernel Function

The observation of permutation invariance is crucial to better acknowledging the principal idea introduced in this work. Since the black-box function of the BO framework (i.e., the simulator) considers the inputs to be sets, it would be extremely beneficial for this to also be incorporated in the BO framework. By sampling an input vector, information is obtained not only regarding what its objective function value is but also regarding all input vectors that are derived from same-group permutations of its elements. For instance, in the $N$-dimensional input space, where $N = n + m$, and $n$ and $m$ correspond to the dimensions of the groups, a single sample would give information about $n! \cdot m!$ vectors. It is now obvious that incorporating this information in BO would greatly benefit its efficiency.

To incorporate this in BO, we need to recall that BO builds a surrogate objective function that is mostly influenced by samples and the covariance matrix which is derived from a kernel function. Modifying the kernel function is a clear pathway for improving the efficiency of BO. The kernel function calculates the covariance between points of the input space and acts as a measure of how similar the surrogate objective function's values are based on the distance between inputs. The Matérn kernel utilized in this work is a stationary kernel, i.e., invariant to translations where the covariance of two input vectors is only dependent on their relative position $k(\mathbf{x}, \mathbf{y}) = k(||\mathbf{x} - \mathbf{y}||)$. Inputs have high covariance when their distance approaches zero, whereas it decreases as they move further apart.

The modified input space generated by the mapping of Algorithm 1 is useful for generating inputs inside the feasible region, but the caveat is that it is not homogeneous. Points that are close in the modified input space might be very far apart when translated to the original input space. The lack of preservation of this closeness property in the modified input space can be expressed as follows:

$$\text{If } ||\mathbf{c}_1 - \mathbf{c}_2|| \leq ||\mathbf{c}_1 - \mathbf{c}_3||$$

it does not necessarily imply that $||f(\mathbf{c}_1) - f(\mathbf{c}_2)|| \leq ||f(\mathbf{c}_1) - f(\mathbf{c}_3)||$.

where $\mathbf{c}_i$ are cells coordinates and $f$ is the mapping from Algorithm 1. In other words, the mapping function is not an increasing function of the well distance. It follows that if the kernel function utilizes the distances of the mapped input space directly, there will likely be a discrepancy between the calculated covariance and the actual covariance between inputs when this translates to the original input space.

To overcome this problem, the kernel function was modified so that covariance is calculated based on the inverse mapping of the input space, i.e., from the $N$-dimensional input space to the $2N$-dimensional one which now represents the actual distance between two cells $(\mathbf{x}, \mathbf{y}) = k(||f^{-1}(\mathbf{x}) - f^{-1}(\mathbf{y})||)$, where $f^{-1}(\mathbf{x}) = (f^{-1}(x_1), f^{-1}(x_2), \ldots f^{-1}(x_{11}))$. By utilizing this technique, the kernel function now considers the actual distance between two cells.

The last issue to be addressed is the permutation invariance. Since each input vector is invariant under $m! \cdot n!$ permutations, to calculate the representative covariance between two inputs, the permutation ordering that minimizes the distance of the one input to the

other needs to be applied. This is a key modification that transforms BO from considering inputs as vectors to now view them as sets. By permuting, it is now able to minimize the distance between sets rather than vectors.

To solve this, a matrix **D** is generated, with each value $d_{ij}$ representing the distance of the *i*th well in **x** from the *j*th well in **y**, as given by Equation (16).

$$\mathbf{D} = d_{ij} = \{||f^{-1}(x_i) - f^{-1}(y_j)||\} \tag{16}$$

Since the goal is to find the permutation that minimizes the distance between wells in the two samples, the problem can be expressed as finding the permutation of rows (or columns) that minimizes the trace of matrix **D**, shown in Table 2, which has complexity $O(N!)$. Since permutation invariance only affects injectors and producers separately, the problem can be reduced to the two counterparts, i.e., finding the right ordering of the injectors which has complexity $O(m!)$ and finding the right ordering of the producers with complexity $O(n!)$. This is a very well-known problem in the literature called the assignment problem. Algorithm A1 (Appendix A), known as the Hungarian method [66], or Munkres algorithm [67], is commonly utilized to reduce its complexity to the order of $O(N^3)$, or in our case, $O(m^3) + O(n^3)$, saving valuable computational time. Algorithm 2, describing the full Kernel modification, is shown below.

**Table 2.** Distance matrix **D** .

|  | $y_1$ | $y_2$ | $\cdots$ | $y_n$ |
|---|---|---|---|---|
| $x_1$ | $d_{11}$ | $d_{12}$ | $\cdots$ | $d_{1n}$ |
| $x_2$ | $d_{21}$ | $d_{22}$ | $\cdots$ | $d_{2n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| $x_n$ | $d_{n1}$ | $d_{n2}$ | $\cdots$ | $d_{nn}$ |

---

**Algorithm 2:** Covariance calculation.

> **Input:** Modified input space
> **Output:** Modified kernel function
> **Function** `GenerateDistanceMatrix`(*Modified input space*):
>> Calculate the inverse mapping $f^{-1}$ for each sample in the modified input space using the 1-1 dictionary values produced from Algorithm 1;
>> Initialize an empty distance matrix;
>> **for** *each pair of samples* $\mathbf{x}_i$ *and* $\mathbf{y}_j$ **do**
>>> Define $f^{-1}(\mathbf{x}_i)$ and $f^{-1}(\mathbf{y}_j)$ as the inverse mappings of $\mathbf{x}_i$ and $\mathbf{y}_j$ respectively;
>>> Calculate the Euclidean distance $d_{ij}$ using the inverse mapping:
>>> $d_{ij} = ||f^{-1}(\mathbf{x}_i) - f^{-1}(\mathbf{y}_j)||$;
>>> Store $d_{ij}$ in the distance matrix;
>>
>> **end**
>> **return** Distance matrix;
>
> **Function** `HungarianMethod`(*Distance matrix*):
>> Use the Hungarian method A1 to find the optimal permutation ordering that minimizes the trace of the distance matrix;
>> **return** Optimal permutation ordering;
>
> **Function** `ModifiedKernel`(*Distance matrix, Optimal permutation ordering*):
>> Calculate the modified kernel function using the optimal permutation ordering;
>> **return** Modified kernel function;

The modified kernel function is represented by a functional form, $k(\mathbf{x}, \mathbf{y}) = k(\min_m || f^{-1}(\mathbf{x}) - f^{-1}(\mathbf{P}_m \mathbf{y})||)$, where $\mathbf{P}_m$ denotes the permutation matrix. This functional form introduces significant non-linearity, due to the involvement of inverse mapping $f^{-1}$ and permutation matrix $\mathbf{P}_m$. This non-linearity poses challenges in directly establishing a proof of satisfaction of Mercer's conditions, which typically rely on simpler, more linear forms of kernel functions. However, despite the inherent complexity introduced by the functional form, extensive sampling efforts across a large number of input samples consistently yield a positive semidefinite covariance matrix. This empirical observation is crucial, as it provides compelling evidence supporting the validity of the modified kernel function. To elaborate, the positive semi-definite covariance matrix implies that the resulting kernel function preserves important properties required for Mercer's conditions, such as symmetry and non-negativity. The positive semi-definite nature of the covariance matrix indicates that any linear combination of kernel evaluations over the sampled inputs remains non-negative, thus aligning with the requirements of Mercer's theorem. Furthermore, although direct analytical proof may be challenging due to the non-linearity of the modified kernel, the empirical validation through covariance matrix analysis corroborates the functional form's adherence to Mercer's conditions. This empirical evidence, coupled with the observed positive semi-definite covariance matrix, lends strong support to the validity of the modified kernel function despite its inherent non-linearity.

Importantly, due to the permutation invariance inherent in the function, it can be construed as a kernel function operating between sets rather than individual vectors. Algorithm 3 delineates the primary workflow of the Bayesian Optimization (BO) framework employed in this study.

---

**Algorithm 3:** Bayesian Optimization framework.

**Input:** Objective function $f$, initial set of samples $X$, maximum iterations $N$
**Output:** Optimal solution $x^*$
Use Algorithm 1 (Mapping Function $f : \mathbb{R}^2 \to \mathbb{R}$) to map the 2D cells to one-dimensional values;
Initialize the set of samples $X$ with initial samples;
Initialize the covariance matrix $K$ using the modified kernel function;
**for** *i from 1 to N* **do**
    Use Algorithm 2 (Covariance calculation) to compute the modified kernel function;
    Update the surrogate model with the modified input space and kernel function;
    Select the next sample point using an acquisition function;
    Evaluate the objective function at the selected sample point;
    Update the set of samples $X$ with the new sample point and corresponding function value;
    Update the covariance matrix $K$ with conditioning based on the new sample point;
**end**
**return** Optimal solution $x^*$ based on the observed samples and their function values;

---

## 4. Case Study

To properly solve the multi-phase compressible fluid flow problem, reservoir simulation is utilized through OPM's open-source Flow software v2023.10 [16] (https://opm-project.org/?page_id=19, (accessed on 17 April 2024)). The aquifer is detailed in Section 3.1, while the BO framework is delineated in Sections 3.2–3.5. Specifically, $m = 3$ injectors and $n = 8$ producers are selected. The injector selection is based on the chosen target rates, aligning with commonly utilized practices in CCS projects [68,69]. Conversely, the decision

to increase the number of producers is motivated by the requirement for pressure maintenance, particularly in a closed system. This assumption is pragmatic, as it is anticipated that most producers function as lateral extensions of a few multi-segment horizontal wells. A safety upper limit for pressure buildup is set to 9000 psi, while a lower limit of 3000 psi is set to mitigate the need for costly pumping procedures.

### 4.1. Exploration–Exploitation Trade-Off

The first phase of testing the BO algorithm involves the utilization of all three candidate acquisition functions introduced in Section 2.3, shown in Equations (11)–(13). Each function is utilized twice, once with explorative hyperparameters values and once with exploitative ones. In total, six BO models are generated, with each one being trained through 50 iterations, by an equivalent amount of simulation runs conducted for each model to refine the surrogate objective function. Details of the selected schedule are outlined in Table 3.

**Table 3.** Schedule targets and constraints.

| Parameter | Value |
|---|---|
| (Target) Injection rate ($CO_2$) | 110 MMscf/day |
| (Target) Production rate (water) | 50 Mstb/day |
| (Constraint) Max production rate ($CO_2$) | 5 MMscf/day |
| (Constraint) Max bottomhole pressure | 9000 psi |
| (Constraint) Min bottomhole pressure | 3000 psi |
| Duration | $\approx 40$ years |
| Theoretical maximum sequestration volume ($CO_2$) | 1.58 Tscf |

In the table above, "M" denotes 1000, "MM" denotes $10^6$, and "T" corresponds to $10^{12}$ while "stb" and "scf" represent units of volume, i.e., stock tank barrel and standard cubic feet, respectively, measured under surface conditions. The theoretical maximum volume shown in Table 3 was calculated by multiplying the target injection rate with the duration of the simulation and represents the target value for the optimizer to reach. However, it is crucial to note that this value may not be feasible without violating the imposed constraints; therefore, it may not be achievable under any set of well configurations. Mass-wise, the target injection $CO_2$ rates correspond to 2.3 MMtn/yr and the maximum sequestration volume to 90 MMtn.

At first, the performance of all six models is compared in terms of maximum storage value attained by timestep, as seen in Figure 6.
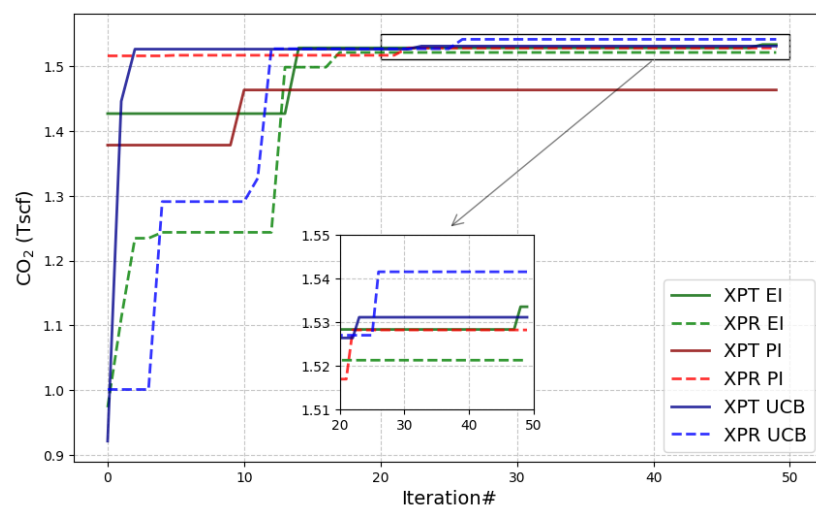
**Figure 6.** Maximum obtained $CO_2$ storage value vs. iteration number.

The first component of the series names signifies the hyperparameter value selection guiding the model towards either exploration (XPR) or exploitation (XPT), while the second component indicates the acquisition function employed (see Equations (11)–(13)). It is notable that all models approach a storage result close to the theoretical maximum already by iteration 15. Certain models (XPT EI, XPT PI, and XPR PI) are fortunate to nearly achieve their optimum value early on, possibly due to a lucky initial guess. This suggests that the targeted objectives of the schedule are relatively easy to maintain throughout the simulation. In contrast to other exploitative models, XPT UCB attains an optimal value by the third iteration. Both XPT EI and XPT UCB demonstrate their ability to refine their predictions in later iterations. However, XPT PI peaks at the 10th iteration but fails to reach the $1.5 \cdot 10^9$ mark, likely because PI inherently leans towards exploitation. In contrast, XPR PI continues to exhibit improvements even after 20 iterations. Notably, both XPR UCB and XPR EI consistently find enhancements throughout the iterations, owing to their explorative nature as well as initially exploring sub-optimal regions.

Since the results produced for the max value seem to be heavily reliant on the initial estimation, the distribution of the storage results that each model generated at every iteration is investigated, as illustrated in Figure 7, to offer valuable insights into the impact of the strategy on model performance. Notably, exploratory strategies (illustrated in the right-hand-side plots) demonstrate a higher susceptibility to unfavorable outputs, which arises from their emphasis on exploring uncharted territories within the input space. This trend is clearly evidenced by the elongated tail of the distribution estimation line (kde) and the behavior of the 20th percentile line (dashed green) of the distribution, indicating that these strategies occasionally obtain sub-optimal outcomes. Conversely, exploitative strategies, as indicated by the 80th percentile line (dashed blue), tend to sample inputs that enable them to remain in proximity to the maximum value. However, an exception emerges with XPT PI, which appears to have become stuck in a local optimum, failing to fully capitalize on potential improvements. This divergence underscores the nuanced interplay between exploitation and exploration strategies and their respective impacts on model behavior and performance.
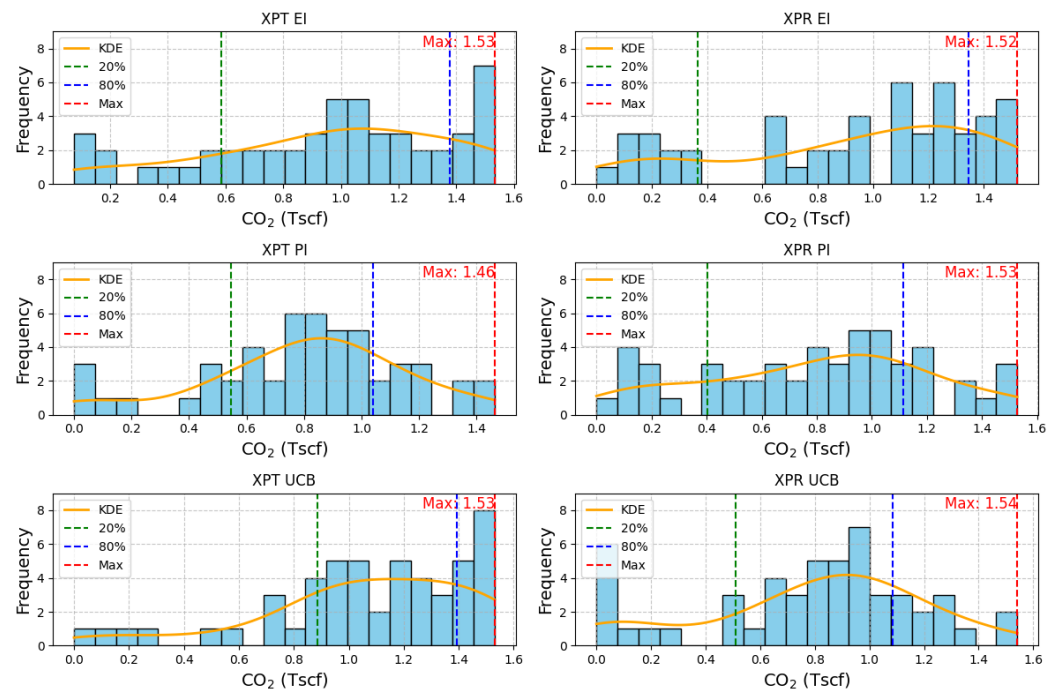
**Figure 7.** Histograms and probability density estimation of storage values.

Finally, to give a measure of the well location change per iteration, in Figure 8, the $\mathcal{L}_1$ norm between the locations in subsequent iterations is depicted as "Vector Distances"

defined as the trace of matrix **D**, as shown in Table 2, aiming to offer insights into the level of exploration across the models. However, since distance alone is not the optimal metric due to permutation invariance, akin to the modification of the kernel function, the "Set Distances" value between subsequent runs is calculated, utilizing Algorithm A1 on **D** to yield the minimum distance. This approach offers a more accurate depiction of the exploration dynamics among the models, mitigating the limitations associated with the traditional distance metric.
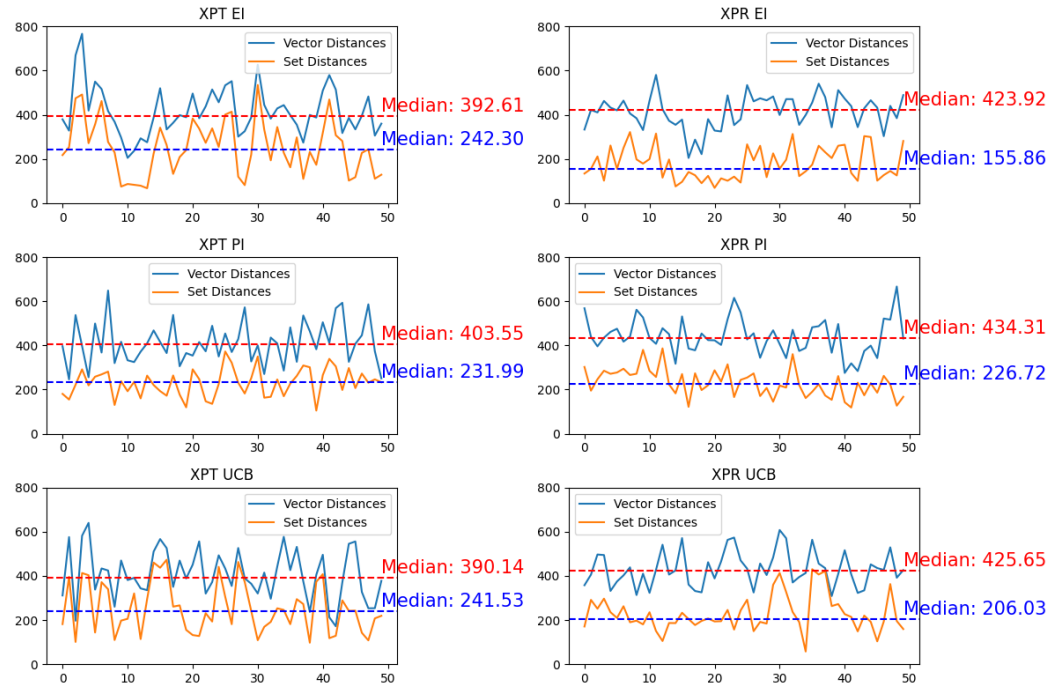
**Figure 8.** "Vector Distances" and "Set Distances" between subsequent inputs for all six investigated models.

The median of the "Vector Distances" series of exploration strategies is clearly higher than that of exploitation ones, as exploration inherently involves probing a wider range of input space. However, it is noteworthy that due to the acquisition function not being directly dependent on the kernel, the Set Distances value between the inputs is considerably higher in exploitative strategies. This observation suggests that exploitative strategies ultimately explore the space more efficiently. In essence, while exploration strategies cover more ground in terms of distance, exploitative strategies manage to navigate the space effectively, potentially due to their focused search around promising areas guided by the acquisition function.

### 4.2. Kernel Comparisons

To further evaluate the performance of the proposed method, a comparison between the modified kernel against the original Matérn kernel and an approach that randomly selects well positions is presented. This test is performed on a more strenuous schedule having a higher target injection rate, increased by more than 50%, as seen in Table 4. The target injection rate now corresponds to 3.6 MMtn/yr and the maximum sequestration volume to 142 MMtn.

**Table 4.** Schedule targets and constraints.

| Parameter | Value |
| --- | --- |
| (Target) Injection rate ($CO_2$) | 170 MMscf/day |
| (Target) Production rate (water) | 77 Mstb/day |
| (Constraint) Max production rate ($CO_2$) | 5 MMscf/day |
| (Constraint) Max bottomhole pressure | 9000 psi |
| (Constraint) Min bottomhole pressure | 3000 psi |
| Duration | ≈40 years |
| Theoretical maximum sequestration volume ($CO_2$) | 2.44 Tscf |

This comparison aims to elucidate the impact of the two modifications made to the kernel described in Section 3 on the final outcome. Specifically, we seek to understand their influence on the end result, as well as their ability to expedite the optimization process. The acquisition function utilized is EI with moderate hyperparameter values, striking a balance between exploration and exploitation. By evaluating these aspects, valuable insights can be gained into the efficacy and efficiency of the proposed modified kernel in comparison with both the original kernel and a random selection strategy.

As illustrated in Figure 9, the modified kernel surpasses the two alternatives in terms of maximum $CO_2$ sequestered. Furthermore, it demonstrates consistent improvement, initiating enhancement as early as at the third timestep and continuing to refine its performance a total of four times before the 15th timestep. In contrast, the original kernel demonstrates improvement only twice, and its performance is comparable only to that of the random process. This stark discrepancy underscores the critical importance of effectively managing kernel functions. Failure to comprehend and address the nuances of optimization caveats would have hindered our ability to achieve such promising outcomes.
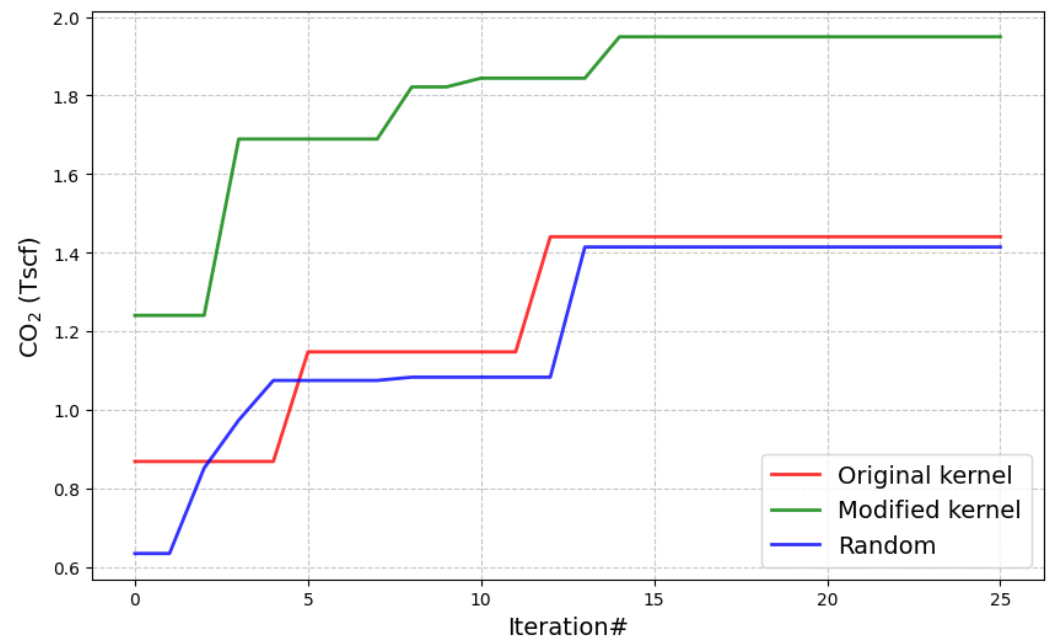
**Figure 9.** Maximum $CO_2$ sequestered volume achieved vs. iteration number.

Notably, the original kernel function's performance is akin to that of a random process. This outcome is attributable not only to improper distance calculations based on the 1D modified input space in the covariance but also to the lack of comprehensive information regarding permutation invariance and the need to manage a considerably larger input space as a consequence. Figure 10 further supports this argument by demonstrating that the density distribution of the results of the original kernel and the random optimization

are very close, as opposed to the modified kernel's results, where the density function is clearly skewed towards the optimal value.
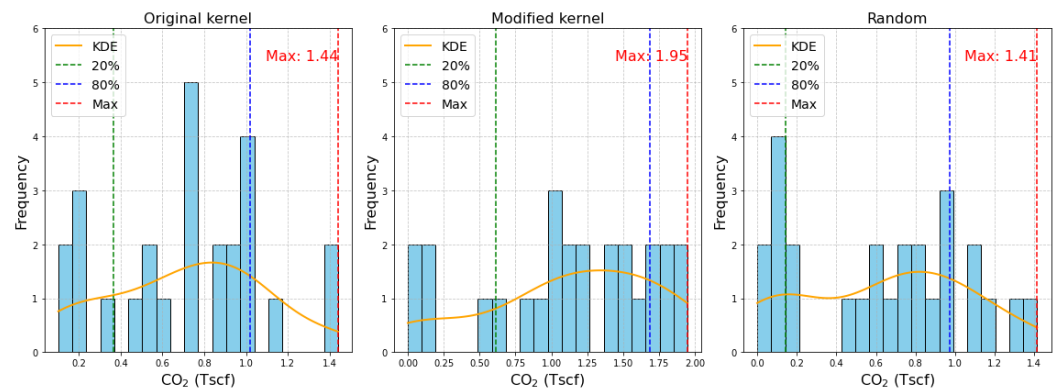
**Figure 10.** Maximum obtained $CO_2$ volume values vs. iteration number.

## 5. Discussion

To facilitate a broad discussion on the modifications and the results presented so far, a brief synopsis of what has transpired thus far needs to be presented. The objective of this work is to apply a BO framework to optimize well placement in a CCS project. Rather than applying BO in a "out of the shelf" straightforward fashion, several key modifications and analyses outlined throughout Section 3 were employed, aiming at enhancing optimization efficiency and effectiveness. Firstly, by utilizing a formation with complex geometry rather than a simple "sugar box", the issue of infeasible areas within the input space emerged and had to be addressed by implementing a transformation technique. This step allowed us to circumvent infeasible regions that took up the majority of the original input space.

This modification was not tax-free, as it introduced a severe issue to the kernel. The non-homeomorphic nature of the map implied that kernel closeness of points in the new modified input space did not translate to the same closeness on the original, thus violating the need for the kernel function to calculate the covariance properly. To address this, modifications were further introduced to the Matérn kernel, utilizing inverse mapping to capture real distances between well positions. By recalculating covariances based on actual distances, this adapted kernel accurately represented the spatial relationships crucial for optimal well placement.

A significant advancement in the proposed methodology was the integration of our knowledge of permutation invariance into the kernel function. From a mathematical point of view, this innovation enabled the kernel to operate on sets rather than conventional vectors, effectively considering the minimum distances between well positions under permutations. The resulting kernel showed to be instrumental in improving optimization outcomes, as demonstrated through rigorous experimentation in Section 4.2, where BO was run with and without the proposed modifications, with the results being significantly better in the modified case.

Clearly, the modified kernel exhibited superior performance, highlighting its pivotal role in driving significant improvements in well placement optimization outcomes. It is our strong belief that a solid foundation has been laid for future research endeavors aimed at optimizing the well placement problem with enhanced precision and efficiency. The proposed method is generalizable to other applications of well placement optimization, not necessarily restricted to a BO framework as long as wells operate under group control, making the input space invariant under permutations. These modifications can also find applications in other learning algorithms characterized by permutation invariant inputs.

In addition to the comparisons presented thus far, it is important to address the optimal placement identified by the algorithms and consider what insights engineers' intuition could offer. Generally, in a closed system, producers and injectors should be positioned as far apart as possible. For this aquifer, an alternative configuration could involve injecting

CO$_2$ at the crest and placing the producers around the periphery, which would potentially enforce the resulting plume to remain relatively stationary and not descend as rapidly. As depicted in Figure 11, the solution derived by the BO algorithm reflects a blend of both engineering perspectives and considerations.
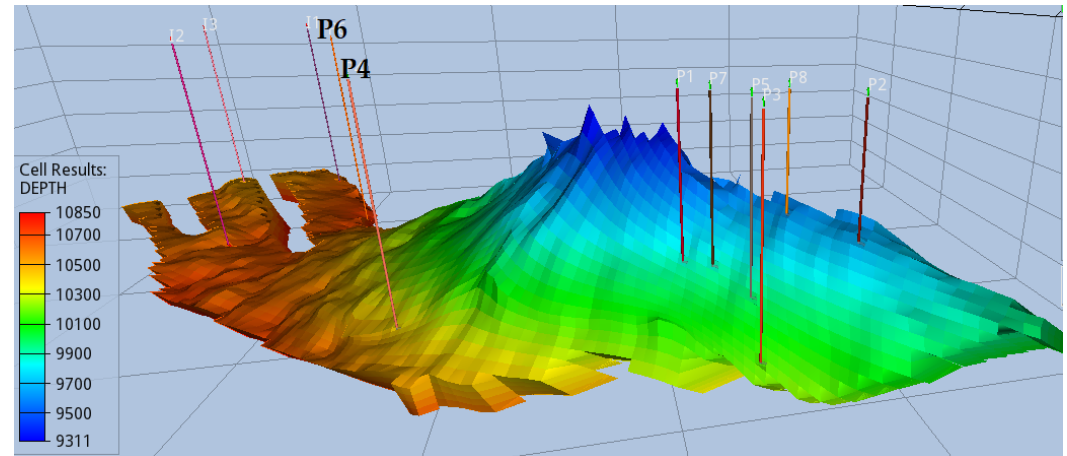
**Figure 11.** Optimal well placement achieved by the proposed algorithm.

Injectors are strategically positioned as far away as possible from the majority of the producers, situated in the region resembling a fork at the base of the aquifer. Conversely, most producers are sited on the opposing side of the crest that resembles a "hill". CO$_2$ naturally migrates upwards, facilitated by a clear pathway to the crest. It is reasonable to assume that producer P6 is strategically placed along the anticipated path of the plume for pressure maintenance, given the substantial injection rate from day 1. Producer P4 is strategically located to ensure a uniformly distributed plume. Considering the gravitational pressure differential that causes CO$_2$ to migrate towards the crest, the advancement of the resulting plume's front would be uneven. However, P4 counteracts this by moderating the pressure in its vicinity, resulting in even plume migration by the end of the CCS period, as illustrated in Figure 12.
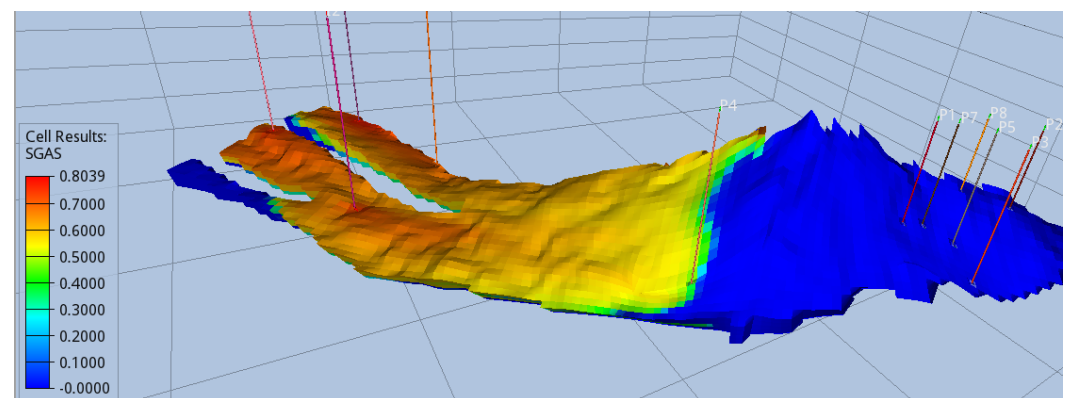
**Figure 12.** Even plume front when CO$_2$ reaches the crest.

Finally, the remaining six producers are positioned behind the crest, in accordance with the second intuition, that is, the deceleration of the CO$_2$ plume once it reaches the crest, primarily due to density differences between CO$_2$ and the brine. At this juncture, breakthrough to the producers becomes inevitable, marking the conclusion of the injection period. It is noteworthy that the simulation ran successfully for approximately 40 years, during which the injection rate naturally decelerated due to pressure buildup. However, as depicted in Figure 13, the plume had not yet reached the producers by this point.
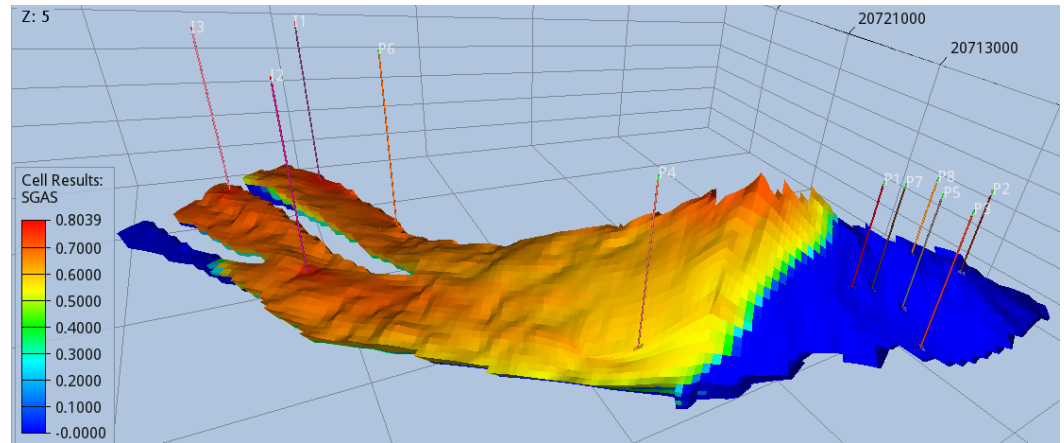
**Figure 13.** CO$_2$ plume migration at the end of the optimal scenario's simulation.

Field pressure, depicted in Figure 14, is shown to emphasize that even after 40 years of continuous development, the aquifer has not yet reached the safety limit pressure. Pressure gradient varies due to the constraints imposed on production/injection and breakthrough rates.
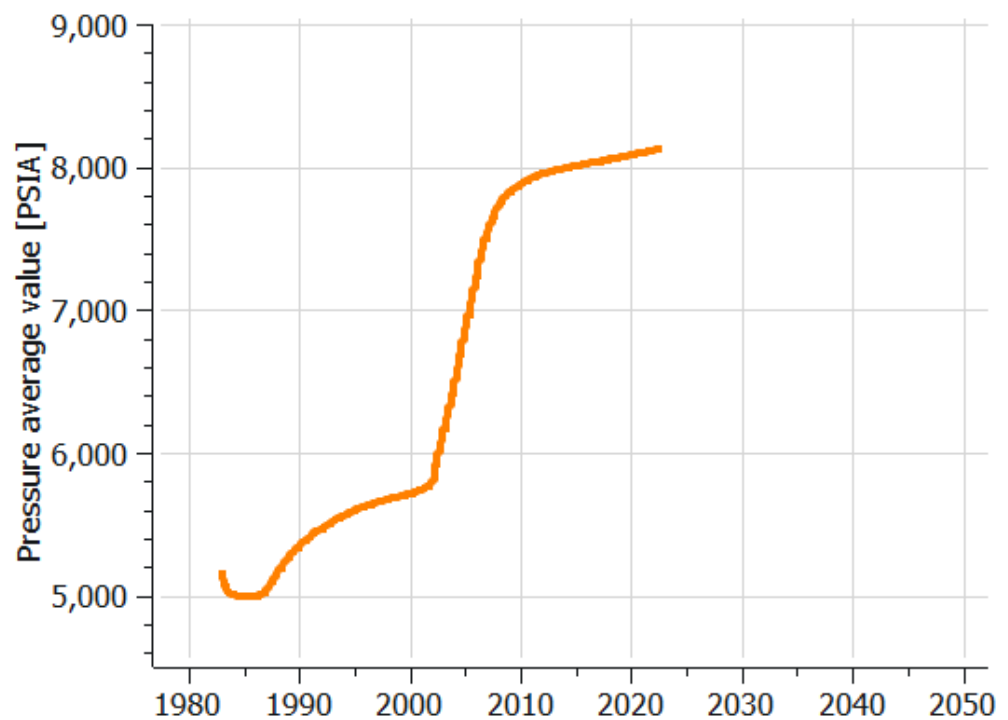
**Figure 14.** Field pressure vs. time.

Overall, BO optimization successfully captured, without any explicit guidance and in a very limited number of simulation runs, the need for producers and injectors to be spaced as further apart as possible, while positioning the producers below the crest to slow down CO$_2$ plume breakthrough. These are the two primary insights crucial for injection optimization, which an experienced engineer would typically independently deduce. Configurations deviating from these principles are deemed sub-optimal, leading to earlier CO$_2$ breakthrough in producers. Consequently, producers are prematurely closed off, pressure builds up faster, and injection rates decline sooner, resulting in reduced total

$CO_2$ injection and less effective plume sweep efficiency. This trend is illustrated in Figure 15, where the optimal case's injection rate (blue line) is compared to a sub-optimal one (red line). Despite the optimal case having a notably higher target injection rate, it maintains this rate for over a decade longer and exhibits a more gradual decline compared with the sub-optimal scenario. In the latter, injectors are not efficiently spaced with respect to the producers; therefore, the pressure reaches the safety limit of 9000 psi in less than a decade of development. The number of producers and injectors, as well as the aquifer system, are identical in both cases.
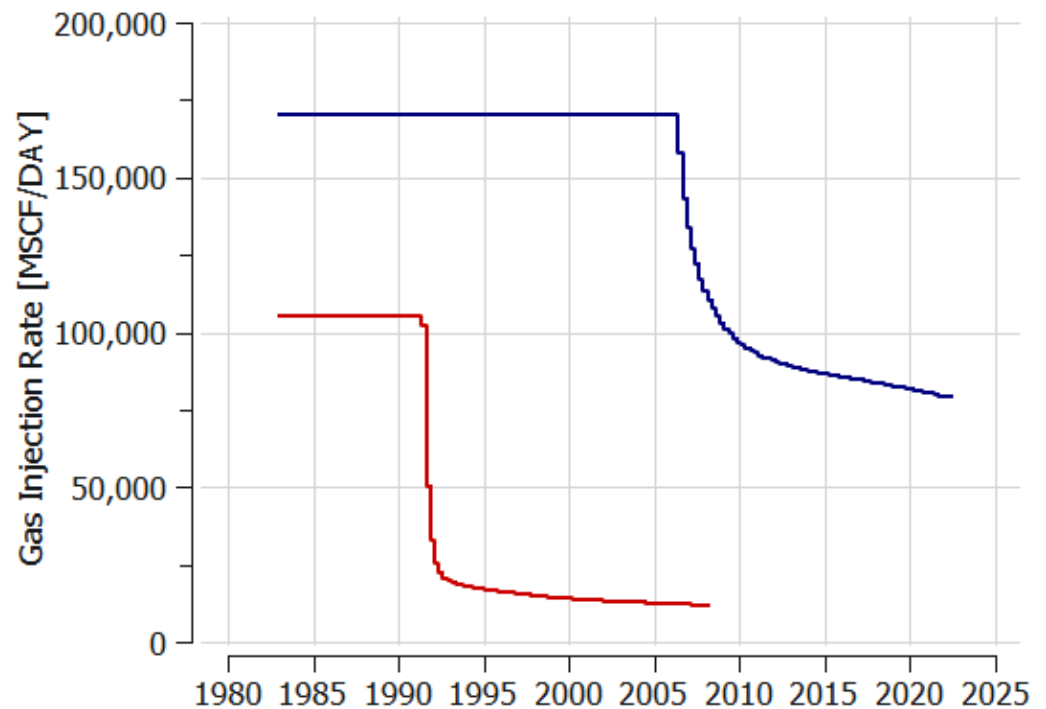
**Figure 15.** Injection rates: optimal case (blue line) vs. sub-optimal case (red line).

Given the engineering-oriented approach of the optimizer's solution, there is a high degree of confidence that the proposed solution is indeed optimal. However, it is essential to acknowledge the complexity introduced by the multi-dimensional nature of the input space. While the solution obtained represents a significant achievement, it is plausible that there exist alternative configurations capable of matching or even surpassing this optimal value. Furthermore, an interesting consideration arises from the spatial proximity of the six producers. Given their close positioning, there is potential for aggregation into one or two total producers during the project's design phase. Consolidating these producers could streamline operational logistics and potentially optimize resource utilization, contributing to further efficiency gains. After performing multiple simulations, aggregating the six wells into one or two and considering the rates of the optimized case, it was seen that the total volume of sequestered $CO_2$ slightly dropped by 3–7%, which could of course be negated by the reduced costs of drilling so many wells. However, a complete evaluation of this trade-off is beyond the scope of this work.

### 6. Conclusions

In this study, the well placement optimization problem was handled as a Bayesian Optimization one. To our knowledge, this was the first attempt of applying BO in a well placement CCS setting. The analysis focused on several key modifications and comparisons to enhance the efficiency and effectiveness of the optimization process. Firstly, the issue of the input space's infeasible area was addressed by introducing a mapping to a modified input space. This adjustment allowed the algorithm not to be concerned with exploring

the infeasible regions by itself, a process which would require hundreds of thousands of iterations. Furthermore, a novel kernel modification was proposed, aiming at achieving invariance under permutations of the input vectors. This adaptation not only enhanced the robustness of the optimization process but also ensured that the model's performance remained consistent across different permutations, a crucial consideration in real-world well optimization.

Subsequently, comprehensive analyses were conducted by using various acquisition functions, including expected improvement (EI), upper confidence bound (UCB), and probability of improvement (POI). Through comparative graphs, the performance differences among these functions was illustrated, providing valuable insights into their respective strengths and limitations in the context of CCS well placement optimization. One of the primary contributions of this study was the comparative evaluation of the developed modified kernel against its unmodified counterpart and a random function. The obtained findings unequivocally demonstrated the substantial impact of the modified kernel on optimization efficacy.

After extensive testing, we successfully optimized the sequestration of $CO_2$ in a saline aquifer. The resulting scenario reflects both intuitive engineering principles and computational efficiency. Our efforts culminated in a comprehensive case study outlining the optimal sequestration of 115 million tons of $CO_2$ over a four-decade period. This achievement represents 80% of the theoretically maximum value of 142 million tons, signifying exceptional efficiency within a closed system. Typically, breakthrough and pressure buildup pose significant challenges in such processes, often limiting their duration and effectiveness.

In conclusion, this study significantly enriches the ongoing discourse surrounding Bayesian Optimization and its practical implementation in complex domains like CCS. Through the introduction of bespoke modifications and comprehensive comparative analyses, a solid foundation for future progress in optimization methodologies has been established, tailored specifically to the nuanced challenges of real-world scenarios. Our efforts pave the way for further refinement and expansion of these techniques, bringing us ever closer to harnessing the complete capabilities of Bayesian Optimization in addressing the intricate well placement problem. Moving forward, the exploration of advanced exotic Bayesian Optimization procedures and frameworks becomes a priority, thereby pushing the boundaries of optimization capabilities in CCS and related fields. This includes extending our methodology to accommodate various types of wells, such as horizontal, inclined, and multi-segment wells, which pose unique challenges due to their non-traditional geometries and flow characteristics. Additionally, the importance of addressing geological uncertainty and the complexities of diverse formations is recognized; thus, future research endeavors will focus on incorporating these factors into the optimization framework. Furthermore, to enhance the industry relevance and real-life applicability of this research, the integration of a net present value (NPV) objective function into the optimization model is intended to be incorporated. This addition will enable the direct optimization of well placement decisions based on economic considerations, aligning our approach more closely with the priorities and objectives of stakeholders in the CCS industry. By integrating NPV considerations, we can ensure that our optimization solutions not only meet technical requirements but also deliver optimal economic outcomes, thereby facilitating more informed decision making in real-world CCS projects.

**Author Contributions:** Conceptualization, S.P.F. and V.G.; methodology, S.P.F.; software, S.P.F.; validation, S.P.F. and V.G.; formal analysis, S.P.F.; investigation, S.P.F.; resources, S.P.F.; data curation, S.P.F.; writing—original draft preparation, S.P.F.; writing—review and editing, S.P.F., I.I. and V.G.; visualization, S.P.F.; supervision, V.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Special Account for Research Funding (E.L.K.E.) of National Technical University of Athens (N.T.U.A.) and by the Hellenic Foundation for Research and Innovation (HFRI) under the 3rd Call for HFRI PhD Fellowships (fellowship number 61/513800).

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A

---

**Algorithm A1:** Hungarian method (Munkres algorithm).

---

**Input:** Cost matrix *costMatrix*
**Output:** Assignment matrix $M$
**Function** `HungarianMethod(`*costMatrix*`):`
    Initialize a matrix $M$ of the same size as *costMatrix* with all zeros;
    Initialize arrays *rowCovered* and *colCovered* of size equal to the number of
     rows and columns of *costMatrix* respectively, filled with **False**;
    **while** *there exists an uncovered zero in costMatrix* **do**
        Select an uncovered zero $Z$ in *costMatrix*;
        **Star** $Z$ and cover its row and column;
        **while** *there exists a starred zero in the same row* **do**
            Find a starred zero $Z'$ in the same row as $Z$;
            Cover the column of $Z'$ and uncover the column of the starred zero in
             the same row as $Z'$;
            **Star** the uncovered zero in the same row as $Z'$;
            $Z \leftarrow$ the uncovered zero just starred;
        **end**
        **if** *no starred zero is found in the same row* **then**
            `AugmentPath(`$Z$`);`
        **end**
    **end**
    **return** $M$;
**Function** `AugmentPath(`$Z$`):`
    Initialize a stack *path* and push $Z$ onto it;
    **while** *true* **do**
        Find a primed zero in the same column as the top element of *path*, say $Z'$;
        **if** *no primed zero is found* **then**
            Break;
        **end**
        Push $Z'$ onto *path*;
        Find a starred zero in the same row as $Z'$, say $Z''$;
        Push $Z''$ onto *path*;
    **end**
    **for** *each element E in path* **do**
        **if** *E is starred* **then**
            **Unstar** $E$;
        **end**
        **else**
            **Star** $E$;
        **end**
    **end**

---

## References

1. Jarvis, S.M.; Samsatli, S. Technologies and infrastructures underpinning future $CO_2$ value chains: A comprehensive review and comparative analysis. *Renew. Sustain. Energy Rev.* **2018**, *85*, 46–68. [CrossRef]
2. Gabrielli, P.; Gazzani, M.; Mazzotti, M. The role of carbon capture and utilization, carbon capture and storage, and biomass to enable a net-zero-$CO_2$ emissions chemical industry. *Ind. Eng. Chem. Res.* **2020**, *59*, 7033–7045. [CrossRef]
3. Bui, M.; Puxty, G.D.; Gazzani, M.; Soltani, S.M.; Pozo, C. *The Role of Carbon Capture and Storage (CCS) Technologies in a Net-Zero Carbon Future*; Frontiers Media: Lausanne, Switzerland , 2021.
4. Rackley, S.; Rackley, S. Introduction to geological storage. In *Carbon Capture and Storage*; Elsevier: Amsterdam, The Netherlands, 2017; pp. 285–304.
5. Tomić, L.; Maričić, V.K.; Danilović, D.; Crnogorac, M. Criteria for $CO_2$ storage in geological formations. *Podzemn. Rad.* **2018**, 61–74. [CrossRef]
6. Ji, X.; Zhu, C. $CO_2$ storage in deep saline aquifers. In *Novel Materials for Carbon Dioxide Mitigation Technology*; Elsevier: Amsterdam, The Netherlands, 2015; pp. 299–332.
7. Bachu, S. Review of $CO_2$ storage efficiency in deep saline aquifers. *Int. J. Greenh. Gas Control* **2015**, *40*, 188–202. [CrossRef]
8. Michael, K.; Golab, A.; Shulakova, V.; Ennis-King, J.; Allinson, G.; Sharma, S.; Aiken, T. Geological storage of $CO_2$ in saline aquifers—A review of the experience from existing storage operations. *Int. J. Greenh. Gas Control* **2010**, *4*, 659–667. [CrossRef]
9. Hannis, S.; Lu, J.; Chadwick, A.; Hovorka, S.; Kirk, K.; Romanak, K.; Pearce, J. $CO_2$ storage in depleted or depleting oil and gas fields: What can we learn from existing projects? *Energy Procedia* **2017**, *114*, 5680–5690. [CrossRef]
10. Mohammadian, E.; Jan, B.M.; Azdarpour, A.; Hamidi, H.; Othman, N.H.B.; Dollah, A.; Hussein, S.N.B.C.M.; Sazali, R.A.B. $CO_2$-EOR/Sequestration: Current Trends and Future Horizons. In *Enhanced Oil Recovery Processes-New Technologies*; IntechOpen: London, UK, 2019.
11. Li, Z.; Dong, M.; Li, S.; Huang, S. $CO_2$ sequestration in depleted oil and gas reservoirs—Caprock characterization and storage capacity. *Energy Convers. Manag.* **2006**, *47*, 1372–1382. [CrossRef]
12. Ismail, I.; Gaganis, V. Carbon Capture, Utilization, and Storage in Saline Aquifers: Subsurface Policies, Development Plans, Well Control Strategies and Optimization Approaches—A Review. *Clean Technol.* **2023**, *5*, 31. [CrossRef]
13. Pruess, K.; García, J.; Kovscek, T.; Oldenburg, C.; Rutqvist, J.; Steefel, C.; Xu, T. Code intercomparison builds confidence in numerical simulation models for geologic disposal of $CO_2$. *Energy* **2004**, *29*, 1431–1444. [CrossRef]
14. Class, H.; Ebigbo, A.; Helmig, R.; Dahle, H.K.; Nordbotten, J.M.; Celia, M.A.; Audigane, P.; Darcis, M.; Ennis-King, J.; Fan, Y.; et al. A benchmark study on problems related to $CO_2$ storage in geologic formations: Summary and discussion of the results. *Comput. Geosci.* **2009**, *13*, 409–434. [CrossRef]
15. Whitaker, S. Flow in porous media I: A theoretical derivation of Darcy's law. *Transp. Porous Media* **1986**, *1*, 3–25. [CrossRef]
16. Rasmussen, A.F.; Sandve, T.H.; Bao, K.; Lauser, A.; Hove, J.; Skaflestad, B.; Klöfkorn, R.; Blatt, M.; Rustad, A.B.; Sævareid, O.; et al. The open porous media flow reservoir simulator. *Comput. Math. Appl.* **2021**, *81*, 159–185. [CrossRef]
17. Trangenstein, J.A.; Bell, J.B. Mathematical structure of the black-oil model for petroleum reservoir simulation. *Siam J. Appl. Math.* **1989**, *49*, 749–783. [CrossRef]
18. Islam, M.R.; Hossain, M.E.; Mousavizadegan, S.H.; Mustafiz, S.; Abou-Kassem, J.H. *Advanced Petroleum Reservoir Simulation: Towards Developing Reservoir Emulators*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
19. Jamal, H.; Islam, M.R.; Farouq Ali, S.M. *Petroleum Reservoir Simulation: A Basic Approach*; Elsevier: Amsterdam, The Netherlands, 2006.
20. Møyner, O.; Lie, K.A. A multiscale two-point flux-approximation method. *J. Comput. Phys.* **2014**, *275*, 273–293. [CrossRef]
21. Eymard, R.; Gallouët, T.; Herbin, R. Finite volume methods. In *Handbook of Numerical Analysis*; Elsevier: Amsterdam, The Netherlands, 2000; Volume 7, pp. 713–1018.
22. Samnioti, A.; Gaganis, V. Applications of Machine Learning in Subsurface Reservoir Simulation—A Review—Part I. *Energies* **2023**, *16*, 6079. [CrossRef]
23. Samnioti, A.; Gaganis, V. Applications of Machine Learning in Subsurface Reservoir Simulation—A Review—Part II. *Energies* **2023**, *16*, 6727. [CrossRef]
24. Zhang, Y.; Lashgari, H.R.; Sepehrnoori, K.; Di, Y. Effect of capillary pressure and salinity on $CO_2$ solubility in brine aquifers. *Int. J. Greenh. Gas Control* **2017**, *57*, 26–33. [CrossRef]
25. Cihan, A.; Birkholzer, J.T.; Bianchi, M. Optimal well placement and brine extraction for pressure management during $CO_2$ sequestration. *Int. J. Greenh. Gas Control* **2015**, *42*, 175–187. [CrossRef]
26. Cameron, D.A.; Durlofsky, L.J. Optimization of well placement, $CO_2$ injection rates, and brine cycling for geological carbon sequestration. *Int. J. Greenh. Gas Control* **2012**, *10*, 100–112. [CrossRef]
27. Bachu, S. $CO_2$ storage in geological media: Role, means, status and barriers to deployment. *Prog. Energy Combust. Sci.* **2008**, *34*, 254–273. [CrossRef]
28. De Coninck, H.; Flach, T.; Curnow, P.; Richardson, P.; Anderson, J.; Shackley, S.; Sigurthorsson, G.; Reiner, D. The acceptability of $CO_2$ capture and storage (CCS) in Europe: An assessment of the key determining factors: Part 1. Scientific, technical and economic dimensions. *Int. J. Greenh. Gas Control* **2009**, *3*, 333–343. [CrossRef]
29. Mirjalili, S.; Mirjalili, S. Genetic algorithm. In *Evolutionary Algorithms and Neural Networks: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 43–55.

30.  Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995 ; Volume 4, pp. 1942–1948.
31.  Guyaguler, B.; Horne, R.N.; Rogers, L.; Rosenzweig, J.J. Optimization of well placement in a Gulf of Mexico waterflooding project. *Spe Reserv. Eval. Eng.* **2002**, *5*, 229–236. [CrossRef]
32.  El-Mihoub, T.A.; Hopgood, A.A.; Nolle, L.; Battersby, A. Hybrid Genetic Algorithms: A Review. *Eng. Lett.* **2006**, *13*, 124–137.
33.  Badru, O.; Kabir, C. Well placement optimization in field development. In Proceedings of the SPE Annual Technical Conference and Exhibition. OnePetro, San Antonio, TX, USA, 16–18 October 2003.
34.  Emerick, A.A.; Silva, E.; Messer, B.; Almeida, L.F.; Szwarcman, D.; Pacheco, M.A.C.; Vellasco, M.M. Well placement optimization using a genetic algorithm with nonlinear constraints. In Proceedings of the SPE Reservoir Simulation Conference, Abu Dhabi, United Arab Emirates, 19–21 October 2009; p. SPE–118808.
35.  Michalewicz, Z.; Nazhiyath, G. Genocop III: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In Proceedings of the 1995 IEEE International Conference on Evolutionary Computation, Perth, WA, Australia, 29 November–1 December 1995; Volume 2, pp. 647–651.
36.  Stopa, J.; Janiga, D.; Wojnarowski, P.; Czarnota, R. Optimization of well placement and control to maximize $CO_2$ trapping during geologic sequestration. *AGH Drilling Oil Gas* **2016**, *33*, 93–104. [CrossRef]
37.  Goda, T.; Sato, K. Global optimization of injection well placement toward higher safety of $CO_2$ geological storage. *Energy Procedia* **2013**, *37*, 4583–4590. [CrossRef]
38.  Loh, W.L. On Latin hypercube sampling. *Ann. Stat.* **1996**, *24*, 2058–2080. [CrossRef]
39.  Islam, J.; Vasant, P.M.; Negash, B.M.; Laruccia, M.B.; Myint, M.; Watada, J. A holistic review on artificial intelligence techniques for well placement optimization problem. *Adv. Eng. Softw.* **2020**, *141*, 102767. [CrossRef]
40.  Frazier, P.I. A tutorial on Bayesian optimization. *arXiv* **2018**, arXiv:1807.02811.
41.  Balabaeva, K.; Akmadieva, L.; Kovalchuk, S. Optimal Wells Placement to Maximize the Field Coverage Using Derivative-Free Optimization. *Procedia Comput. Sci.* **2020**, *178*, 65–74. [CrossRef]
42.  Bordas, R.; Heritage, J.; Javed, M.; Peacock, G.; Taha, T.; Ward, P.; Vernon, I.; Hammersley, R. A Bayesian Optimisation Workflow for Field Development Planning Under Geological Uncertainty. In Proceedings of the ECMOR XVII. European Association of Geoscientists & Engineers, Online, 14–17 September 2020; Volume 2020, pp. 1–20.
43.  Kumar, A. Search space partitioning, MCTS and trust-region Bayesian optimization for joint optimization of well placement and control. In Proceedings of the 83rd EAGE Annual Conference & Exhibition. European Association of Geoscientists & Engineers, Madrid, Spain, 6–9 June 2022; Volume 2022, pp. 1–5.
44.  Kumar, A. High-Dimensional Bayesian Optimization using Sparse-Axis Aligned Subspaces for Joint Optimization of Well Control and Placement. In Proceedings of the 84th EAGE Annual Conference & Exhibition. European Association of Geoscientists & Engineers, Vienna, Austria, 5–8 June 2023; Volume 2023; pp. 1–5.
45.  Eriksson, D.; Pearce, M.; Gardner, J.; Turner, R.D.; Poloczek, M. Scalable global optimization via local Bayesian optimization. *Adv. Neural Inf. Process. Syst.* **2019**, *32* .
46.  Eriksson, D.; Jankowiak, M. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In Proceedings of the Uncertainty in Artificial Intelligence. PMLR, Online, 27–30 July 2021; pp. 493–503.
47.  Lu, X.; Jordan, K.E.; Wheeler, M.F.; Pyzer-Knapp, E.O.; Benatan, M. Bayesian Optimization for Field-Scale Geological Carbon Storage. *Engineering* **2022**, *18*, 96–104. [CrossRef]
48.  Javed, M.A. Bayesian Optimization of Deviated Well Trajectories under Geological Uncertainty. Master's Thesis, Heriot-Watt University, Edinburgh, UK, 2020.
49.  Wang, S.; Chen, S. A Novel Bayesian Optimization Framework for Computationally Expensive Optimization Problem in Tight Oil Reservoirs. In Proceedings of the SPE Annual Technical Conference and Exhibition, Anaheim, CA, USA, 8–10 May 2017; p. D021S014R007.
50.  Onwunalu, J.E.; Durlofsky, L.J. Application of a particle swarm optimization algorithm for determining optimum well location and type. *Comput. Geosci.* **2010**, *14*, 183–198. [CrossRef]
51.  Williams, C.K.; Rasmussen, C.E. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006; Volume 2.
52.  Dou, Z. Bayesian Global Optimization Approach to the Oil Well Placement Problem with Quantified Uncertainties. Ph.D. Thesis, Purdue University, West Lafayette, IN, USA, 2015.
53.  Agnihotri, A.; Batra, N. Exploring bayesian optimization. *Distill* **2020**, *5*, e26. [CrossRef]
54.  Görtler, J.; Kehlbeck, R.; Deussen, O. A visual exploration of gaussian processes. *Distill* **2019**, *4*, e17. [CrossRef]
55.  Do, C.B. The multivariate Gaussian distribution. In *Section Notes, Lecture on Machine Learning, CS*; Stanford University: Stanford, CA, USA, 2008 ; Volume 229.
56.  Genton, M.G. Classes of kernels for machine learning: A statistics perspective. *J. Mach. Learn. Res.* **2001**, *2*, 299–312.
57.  Olver, F.W.; Maximon, L.C. Bessel functions. In *NIST Handbook of Mathematical Functions*; Cambridge University Press: Cambridge, UK, 2010.
58.  Artin, E. *The Gamma Function*; Courier Dover Publications: Mineola, NY, USA, 2015.
59.  Buhmann, M.D. Radial basis functions. *Acta Numer.* **2000**, *9*, 1–38. [CrossRef]
60.  Minh, H.Q.; Niyogi, P.; Yao, Y. Mercer's theorem, feature maps, and smoothing. In Proceedings of the International Conference on Computational Learning Theory, Pittsburgh, PA, USA, 22–25 June 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 154–168.

61. Lawrence, D.B. *The Economic Value of Information*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
62. Jones, D.R. A taxonomy of global optimization methods based on response surfaces. *J. Glob. Optim.* **2001**, *21*, 345–383. [CrossRef]
63. Vazquez, E.; Bect, J. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *J. Stat. Plan. Inference* **2010**, *140*, 3088–3095. [CrossRef]
64. Peaceman, D.W. *Fundamentals of Numerical Reservoir Simulation*; Elsevier: Amsterdam, The Netherlands, 2000.
65. Nogueira, F. Bayesian Optimization: Open Source Constrained Global Optimization Tool for Python. 2014. Available online: https://github.com/bayesian-optimization/BayesianOptimization (accessed on 17 April 2024).
66. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
67. Munkres, J. Algorithms for the assignment and transportation problems. *J. Soc. Ind. Appl. Math.* **1957**, *5*, 32–38. [CrossRef]
68. Rock, L.; O'Brien, S.; Tessarolo, S.; Duer, J.; Bacci, V.O.; Hirst, B.; Randell, D.; Helmy, M.; Blackmore, J.; Duong, C.; et al. The Quest CCS project: 1st year review post start of injection. *Energy Procedia* **2017**, *114*, 5320–5328. [CrossRef]
69. Eiken, O.; Ringrose, P.; Hermanrud, C.; Nazarian, B.; Torp, T.A.; Høier, L. Lessons learned from 14 years of CCS operations: Sleipner, In Salah and Snøhvit. *Energy Procedia* **2011**, *4*, 5541–5548. [CrossRef]