A Sampling Criterion for Constrained Bayesian Optimization with Uncertainties

Reda El Amri, Rodolphe Le Riche, Céline Helbert,
Christophette Blanchet-Scalliet & Sébastien Da Veiga

CENTRE
MERSENNE

A Sampling Criterion for Constrained Bayesian Optimization with Uncertainties

Reda El Amri [1]
Rodolphe Le Riche [2]
Céline Helbert [3]
Christophette Blanchet-Scalliet [4]
Sébastien Da Veiga [5]

[1] Ecole Centrale de Lyon, now with IFP Energies Nouvelles, France
E-mail address: mohamed-reda.el-amri@ifpen.fr
[2] CNRS LIMOS (Mines St-Etienne and UCA), France
E-mail address: leriche@emse.fr
[3] Ecole Centrale de Lyon, Institut Camille Jordan, UMR CNRS 5208, France
E-mail address: celine.helbert@ec-lyon.fr
[4] Ecole Centrale de Lyon, Institut Camille Jordan, UMR CNRS 5208, France
E-mail address: christophette.blanchet@ec-lyon.fr
[5] Safran Tech, Modelling & Simulation, Magny-Les-Hameaux, France
E-mail address: sebastien.da-veiga@ensai.fr.

Abstract. We consider the problem of chance constrained optimization where it is sought to optimize a function and satisfy constraints, both of which are affected by uncertainties. The real world declinations of this problem are particularly challenging because of their inherent computational cost.

To tackle such problems, we propose a new Bayesian optimization method. It applies to the situation where the uncertainty comes from some of the inputs, so that it becomes possible to define an acquisition criterion in the joint optimized-uncertain input space. The main contribution of this work is an acquisition criterion that accounts for both the average improvement in objective function and the constraint reliability. The criterion is derived following the Stepwise Uncertainty Reduction logic and its maximization provides both optimal design variables and uncertain parameters. Analytical expressions are given to efficiently calculate the criterion. Numerical studies on test functions are presented. It is found through experimental comparisons with alternative sampling criteria that the adequation between the sampling criterion and the problem contributes to the efficiency of the overall optimization.

1. Introduction

Despite the long-term research efforts put into numerical optimization, many practical applications remain difficult. There are three main reasons: most real problems involve nonlinear constraints, the objective function or the constraints are numerically costly to evaluate (e.g., when nonlinear finite elements underlie the optimization criteria), and some of the parameters are uncertain.

To ease the computing load, Bayesian Optimization (BO) incorporates kriging surrogates to save calls to the objective function, as embodied in the archetypal EGO algorithm [46]. The original optimization problem is translated into a series of other problems, that of the acquisition of new points where the costly function will be calculated. The acquisition criterion is based on the kriging model and it mitigates the optimization of the function and the improvement of the kriging model [20]. BO has rapidly been extended to encompass constraints [17, 21, 37, 45].

In this article, the focus is not only on costly and general nonlinear constrained optimization but also on problems that are affected by uncertainties.

Uncertainties may originate from random environments such as weather, noise in sensors or uncontrolled boundary conditions. Many physical models also come with uncertainties in an attempt to describe a lack of knowledge about the true phenomena. Big data applications are confronted to uncertainties which reflect the part of the data that cannot be handled at a time, either because the data arrive as a dynamic flow, or because the volume is too large to be processed in a single step. Since uncertainties are so ubiquitous, robustness against uncertainties is becoming an important aspect of optimization problem solutions.

When the uncertainties cannot be characterized in stochastic terms such as probabilities, strong guarantees about the robustness of the solutions can be obtained with deterministic approaches based on worst-case scenarios over the set of possible uncertainties [8, 19, 29]. If this set is large (possibly infinite) and the problem non-convex, the conservatism of the solutions and the computational tractability are inherent difficulties of this family of methods. When the uncertainties are seen as stochastic, two situations may be distinguished.

In a first class of problems, the uncertainties are instantiated within the objective or constraint functions and cannot be chosen. Such uncertainties are a hard coded, not controllable, noise corrupting the objective and the constraint functions. A typical situation is when the functions resorts to random number generations, e.g., for a Monte Carlo simulation, and no access to the source code is granted. Stochastic algorithms can, under conditions about their own stochasticity, accomodate the noise in the observations and still converge to problem solutions: this has given rise in the early 50's to stochastic descent algorithms [3, 24] that have since then experienced great developments [2, 48] often in relation to machine learning [25]; well-performing versions of (stochastic) evolutionary algorithms for noisy functions have been identified [11, 28] thanks to competitions on reference benchmarks [4].

In a second class of problems, the uncertainties perturb parameters that are distinct from the optimization variables and can be chosen during the simulations. The separation between optimization (or design) variables and uncertain parameters was already underlying Taguchi's design of experiments in the 80's [27]. Because of this separation and providing that a probability of occurence of the uncertainties exists, a statistical modelling in the joint *design* × *uncertain* parameters space is possible. This will be the context of the current work.

A key step when optimizing in the presence of uncertainties is the formulation of the problem, i.e., the choice of the robustness criteria. Considering first unconstrained problems, relevant criteria are the expectation of the objective [23] or one of its (super-)quantiles [49, 50]. In Robust Optimization, the uncertainties are handled in terms of specific compromises between the average performance and its dispersion [9, 34] or by investigating all such Pareto optimal compromises through a multi-objective formulation [42].

When there are constraints that depend on the uncertainties, the feasibility of the solutions is typically measured in probability. Probabilistic models of the constraints are called chance constraints [32] or reliability constraints [10]. The field of Reliability-Based Design Optimization (RBDO) is concerned with the resolution of optimization problems that contain reliability constraints [5]. The optimization problems are formulated in terms of statistical criteria such as probabilities of satisfying the constraints, expectations or (super-)quantiles or conditional quantiles of the objective function [26, 40, 49]. When the statistical criteria cannot be calculated analytically, the bottleneck of the computational cost of the optimization is even more stringent since the statistical criteria must be numerically estimated within the optimization iterations. This is sometimes called the double loop issue i.e., for each design point visited, many simulations are needed to encompass the effects of the uncertainties. Several paths have been proposed to circumvent the double loop issue, some approximating the probability of feasibility by reliability indices [51], others improving the efficiency of the probability calculations

(e.g., stratified sampling in [53]), others decoupling the reliability estimation from the optimization. [47] gives a review of some of these techniques.

In the last decade, numerous contributions to the optimization of costly functions with uncertainties have relied on the learning of a metamodel of the true functions, in particular Gaussian processes (GP) [14, 31]. In [23] and [1], the GP not only helps for the optimization (or for the inversion) of the design variables, but it also serves to define an optimal sampling scheme.

In this article, the problem of minimizing the mean of a stochastic function under chance constraints is addressed. The objective function and the constraints are costly in the sense that they cannot be calculated more than a hundred times. Furthermore the problem is assumed to be nonconvex so that part of the solution process will not be analytical. Uncertainties are described by parameters different from the optimization variables and that can be chosen in the calculations. Generalizing [23], an optimization and sampling Bayesian procedure is proposed that accounts for probabilistic constraints. The acquisition criterion is based on the feasible improvement, which is known to be more myopic (less explorative) but also numerically more tractable than its information theoretic counterparts [21, 36]. Indeed, information theoretic criteria, which quantify reductions in the entropy of the predicted optima locations or values, imply internal optimizations of the GP trajectories. Note also that [21, 36] do not consider uncertain parameters. A less myopic feasible improvement criterion, which looks two-steps ahead, is given in [52], yet, again, this work does not address uncertain parameters. On the contrary, uncertainties are examined in [41]: thanks to a spectral decomposition of the GP covariance functions, the variance and the mean of the objective function under the uncertain parameters become a GP, which then opens the way to acquisition criteria based on improvement. The constraint bears on the objective variance. Our contribution differs from [41] essentially by the proposition of a policy to choose where to evaluate the uncertain parameters, and secondarily, by the consideration of general constraints.

After formulating the problem (Sections 2.1 and 2.2), a principle for devising robust bayesian problems is stated (Section 2.3) which applies to any given progress measure. In Section 3, this principle is applied to the feasible improvement as a specific progress measure. The associated sampling criterion is introduced in Section 4. It is a Stepwise Uncertainty Reduction (SUR) criterion [7], a one-step-ahead variance reduction, for which a proxy which is easier to compute is presented. The resulting algorithm is summarized in Section 5, and its performance assessed on an analytical and an industrial test case.

2. Problem formulation

2.1. Starting problem

Let $f(\mathbf{x}, \mathbf{u})$ be the scalar output of an expensive computer simulation and let $g_i(\mathbf{x}, \mathbf{u})$, $i = 1, \ldots, l$ be the set of constraints, where $\mathbf{x} \in \mathcal{S}_\mathcal{X}$ can be precisely chosen while $\mathbf{u} \in \mathcal{S}_\mathcal{U}$ is a realization of a vector of random variables $\mathbf{U}$ with a specified probability density function $\rho_\mathbf{U}$. Such a formulation with design and random variables is general to all optimization problems under uncertainty. The examples in this article belong to continuous optimization in the sense that $\mathcal{S}_\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{S}_\mathcal{U} \subset \mathbb{R}^m$. Nevertheless, it is important to note that the framework of our work, the Gaussian processes and the algorithms which will be introduced, generalize nicely to spaces that contain discrete variables (see for example [35, 43]).

Our goal is to find x which minimizes f while insuring that the g_i's lies under a failure threshold (0 in general). In the presence of uncertainties, Robust Optimization aims at controlling the impact of uncertainties on the performance of the optimal solution. The problem is that $f(\mathbf{x}, \mathbf{U})$ and $g_i(\mathbf{x}, \mathbf{U})$, $i = 1, \ldots, l$ are random quantities induced by $\mathbf{U}$. In order to perform optimization, we need to fall back on a deterministic form which is achieved by applying some statistical measures to $f(\mathbf{x}, \mathbf{U})$ and $g_i(\mathbf{x}, \mathbf{U})$, $i = 1, \ldots, l$ [26, 49].

In this article, the constrained optimization problem under uncertainties is formulated as the minimization of the expectation over $\mathbf{U}$ of f while all the constraints $g_i(\mathbf{x}, \mathbf{U}) \le 0$, $i = 1, \dots, l$ are satisfied with a high probability:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{S}_\mathcal{X}} \mathbb{E}[f(\mathbf{x}, \mathbf{U})] \quad \text{s.t.} \quad \mathbb{P}(g_i(\mathbf{x}, \mathbf{U}) \le 0, \ i = 1, \dots, l) \ge 1 - \alpha$$
$$\text{where } \mathbf{U} \sim \rho_\mathbf{U} \text{ with support } \mathcal{S}_\mathcal{U}. \tag{2.1}$$

α is a reliability parameter representing the allowed constraint violation level ($0 < \alpha < 1$).

In the formulation of Equation (2.1), the emphasis is on constraint satisfaction with a guaranteed reliability. This is common in RBDO where constraints are satisfied in probability and the objective function is deterministic [14]. Such problems are also said to have chance constraints [8]. In addition in Equation (2.1), random events affect the objective function and are taken into account through an expectation. Thanks to its linearity, the expectation is the simplest statistical measure. Besides, efficient approaches exist to estimate and optimize it [23]. Formulations such as Equation (2.1) with a mean objective function and chance constraints have been called a "model for qualitative failure" [2] in which the objective function and the constraints are taken as independent. Other formulations with quantiles conditioned by feasibility have been given in [26, 40] but they remain an open challenge for costly problems. In the current work, Equation (2.1) is addressed because it is a compromise between complete robust formulations and mathematical tractability.

By seeing the probability as an expectation, the constraint part of Equation (2.1) becomes

$$\mathbb{P}(g_i(\mathbf{x}, \mathbf{U}) \le 0, \ i = 1, \dots, l) \ge 1 - \alpha$$
$$\Leftrightarrow 1 - \alpha - \mathbb{E}[\mathbb{1}_{\{g_i(\mathbf{x}, \mathbf{U}) \le 0, \ i=1, \dots, l\}}] \le 0.$$

From the last expression, the problem (2.1) is equivalent to

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{S}_\mathcal{X}} \ z(\mathbf{x}) \text{ s.t. } c(\mathbf{x}) \le 0 \tag{2.2}$$

where $z(\cdot) := \mathbb{E}[f(\cdot, \mathbf{U})]$ and $c(\mathbf{x}) := 1 - \alpha - \mathbb{E}[\mathbb{1}_{\{g_i(\mathbf{x}, \mathbf{U}) \le 0, \ i=1, \dots, l\}}]$.

2.2. Gaussian Process regression framework

In the context of expensive computer simulation, the Problem (2.2) is approximated with Gaussian processes (GPs). Directly building a metamodel for z and c would need too many evaluations of f and g_i to estimate the expectation and the probabilities. Therefore, GP approximations of f and the g_i's are built in the joint space $\mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{U}$. Models for z and c in the design space $\mathcal{S}_\mathcal{X}$ are then deduced from them. More precisely, we suppose that f and the constraints $(g_i)_{i=1}^l$ are realizations of independent Gaussian processes F and G_i such that

$$F(\mathbf{x}, \mathbf{u}) \sim \mathcal{GP}(m_F(\mathbf{x}, \mathbf{u}), k_F(\mathbf{x}, \mathbf{u}, \mathbf{x}', \mathbf{u}')),$$
$$\forall i = \{1, \dots, l\} \ , \ G_i(\mathbf{x}, \mathbf{u}) \sim \mathcal{GP}(m_{G_i}(\mathbf{x}, \mathbf{u}), k_{G_i}(\mathbf{x}, \mathbf{u}, \mathbf{x}', \mathbf{u}')),$$

where m_F and m_{G_i} are the mean functions while k_F and k_{G_i} are the covariance functions.

Let $F^{(t)}$ and $G_i^{(t)}$ denote the Gaussian processes conditioned on the t observations, $f^{(t)} = (f(\mathbf{x}_1, \mathbf{u}_1), \dots, f(\mathbf{x}_t, \mathbf{u}_t))$ and $g_i^{(t)} = (g_i(\mathbf{x}_1, \mathbf{u}_1), \dots, g_i(\mathbf{x}_t, \mathbf{u}_t))$, obtained at points $D^{(t)} = \{(\mathbf{x}_k, \mathbf{u}_k), k = 1, \dots, t\}$. Since the expectation is a linear operator applied to f, it follows that $Z^{(t)}(\mathbf{x}) = \int_{\mathbb{R}^m} F^{(t)}(\mathbf{x}, \mathbf{u}) \rho_\mathbf{U}(\mathbf{u}) d\mathbf{u}$ is still a Gaussian process with known mean $m_Z^{(t)}$ and covariance function

$k_Z^{(t)}$ given by:

$$m_Z^{(t)}(\mathbf{x}) = \int_{\mathbb{R}^m} m_F^{(t)}(\mathbf{x}, \mathbf{u})\rho_{\mathbf{U}}(\mathbf{u})d\mathbf{u},$$

$$k_Z^{(t)}(\mathbf{x}, \mathbf{x}') = \iint_{\mathbb{R}^{2m}} k_F^{(t)}(\mathbf{x}, \mathbf{u}, \mathbf{x}', \mathbf{u}')\rho_{\mathbf{U}}(\mathbf{u}, \mathbf{u}')d\mathbf{u}d\mathbf{u}'. \tag{2.3}$$

The integrals that appears in (2.3) can be evaluated analytically for specific choices of ρ_U and k_F (see [23]). In the general case, a quadrature rule can be used to approximate these integrals or a Monte Carlo method if m is large.

We also introduce the process

$$C^{(t)}(\mathbf{x}) = 1 - \alpha - \int_{\mathbb{R}^m} \mathbb{1}_{\cap_{i=1}^l \{G_i^{(t)}(\mathbf{x},\mathbf{u}) \leq 0\}} \rho_{\mathbf{U}}(\mathbf{u})d\mathbf{u}$$

which is the statistical model of the constraint c (Equation (2.2)). Note that the process $C^{(t)}$ is not Gaussian.

To solve Problem (2.2), Bayesian optimization is used where functions z and c are considered realizations of processes Z and C. The procedure is detailed next.

2.3. **A general principle for devising robust Bayesian optimization algorithms**

Now that the problem has been formulated (Section 2.1) and the GP models introduced (Section 2.2), we present a Bayesian algorithm to solve Problem (2.1) within a restricted number of calls to f and the g_i's. Before going into the details of the method, it is important to understand the general principle that underlies the design of robust Bayesian Optimisation (BO) algorithms.

Proposition 2.1 (A general principle to devise robust BO algorithms)**.** *Robust Bayesian Optimization algorithms can be designed as follows:*

(A) *define a progress measure $P(x)$ in relation with the problem formulation and calculated from the GPs trajectories.*

(B) *The robust BO algorithm is:*

> *Define an initial space filling design in the joint space $\mathcal{S}_{\mathcal{X}} \times \mathcal{S}_{\mathcal{U}} : D^{(n_0)}$*
> *Initialize all conditional GPs: $F^{(n_0)}$, $G_i^{(n_0)}, i = 1, \ldots, l$ with $D^{(n_0)}$*
> ***while** stopping criterion not met **do***
> *Determine a desirable, targeted, $\mathbf{x}$ by maximizing the expectation of the progress measure,*

$$\mathbf{x}_{targ} = \arg\max_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} \mathbb{E}\left(P^{(t)}(\mathbf{x})\right). \tag{2.4}$$

> *The next iterate minimizes the one-step-ahead variance of the progress measure at $\mathbf{x}_{targ}$,*

$$(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) = \arg\min_{\tilde{\mathbf{x}}, \tilde{\mathbf{u}} \in \mathcal{S}_{\mathcal{X}} \times \mathcal{S}_{\mathcal{U}}} \mathbb{VAR}\left(P^{(t+1)}(\mathbf{x}_{targ})\right) \tag{2.5}$$

> *where $P^{(t+1)}$ is evaluated with GPs updated according to $D^{(t+1)} = D^{(t)} \cup \{(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})\}$.*
> *Calculate the simulator response, i.e., f and $g_i, i = 1, \ldots, l$, at the next point $(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$.*
>
> *Update the design $D^{(t+1)} = D^{(t)} \cup \{(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})\}$*
> *Update the Gaussian processes, $F^{(t+1)}$, $G_i^{(t+1)}, i = 1, \ldots, l$.*
> ***end while***

Various algorithms can be obtained by changing the measure of progress P. In this article it will be the feasible improvement, which will soon be presented. Other measures of progress are possible, for example $P(\mathbf{x}) = -F(\mathbf{x}) - \sum_{i=1}^{l} p_i \max(0 \,, \, G_i(\mathbf{x}, \mathbf{u}^{\mathrm{mod}}))$ where the p_i's are positive penalty parameters and $\mathbf{u}^{\mathrm{mod}}$ the mode of $\mathbf{U}$, or $P(\mathbf{x}) = \max([z_{\min}^{\mathrm{feas}} - Z(\mathbf{x}) \mid C(\mathbf{x}) \leq 0], 0)$ where $z_{\min}^{\mathrm{feas}}$ is the best objective function associated to a feasible point. The goal of the next sections is to present the methodology and the associated formulas when the progress measure is the feasible improvement, chosen for its closeness to the problem formulation and its computability. The one-step-ahead variance can be difficult to tackle so approximations are useful. In this text, the generic term "sampling criterion" relates to the one-step-ahead variance or its proxy. For costly problems, the stopping criterion in Proposition 2.1 is often a given number of calls to the simulator.

3. The progress measure and the associated targeted point $\mathbf{x_{targ}}$

Following Proposition 2.1, the first step consists in defining a progress measure $P^{(t)}$, which will be the cornerstone of the definition of what a most promising candidate for evaluation, $\mathbf{x}_{\mathrm{targ}}$, is. The maximization of its expectation should contribute to both solving the constrained optimization problem and improving the GPs. The most popular progress measure for optimization under constraints is the Feasible Improvement [44, 46] defined by

$$FI^{(t)}(\mathbf{x}) = I^{(t)}(\mathbf{x}) \; \mathbb{1}_{\{C^{(t)}(\mathbf{x}) \leq 0\}} \,,$$

where $I^{(t)}(x) = \left(z_{\min}^{\mathrm{feas}} - Z^{(t)}(\mathbf{x})\right)^{+}$ denotes the improvement over the current feasible minimum value.

In our case, $z_{\min}^{\mathrm{feas}}$ must be further explained because it is not directly observed. This will be the subject of the next section. The definition of $z_{\min}^{\mathrm{feas}}$ and the fact that $C^{(t)}(\mathbf{x})$ is not Gaussian is a difference between the FI of this article and those in [44, 46].

Following Proposition 2.1 and Equation (2.4) the promising point in the control space is obtained by maximizing the expectation of the progress measure. Here it corresponds to maximizing the *Expected Feasible Improvement* (EFI),

$$\mathbf{x}_{\mathrm{targ}} = \arg \max_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} \mathrm{EFI}^{(t)}(\mathbf{x}). \tag{3.1}$$

where $\mathrm{EFI}^{(t)}(\mathbf{x})$ is $\mathbb{E}\left(FI^{(t)}(\mathbf{x})\right)$. The independence of the GP processes implies that the EFI can be expressed as

$$\mathrm{EFI}(\mathbf{x}) = \mathrm{EI}^{(t)}(\mathbf{x})\mathbb{P}(C^{(t)}(\mathbf{x}) \leq 0).$$

The first term is the well known *Expected Improvement* (EI) for which an analytical expression is available,

$$\mathrm{EI}^{(t)}(\mathbf{x}) = (z_{\min}^{\mathrm{feas}} - m_Z^{(t)}(\mathbf{x}))\Phi\left(\frac{z_{\min}^{\mathrm{feas}} - m_Z^{(t)}(\mathbf{x})}{\sigma_Z^{(t)}(\mathbf{x})}\right) + \sigma_Z^{(t)}(\mathbf{x})\phi\left(\frac{z_{\min}^{\mathrm{feas}} - m_Z^{(t)}(\mathbf{x})}{\sigma_Z^{(t)}(\mathbf{x})}\right), \tag{3.2}$$

where $\sigma_Z^{(t)}(\mathbf{x}) = \sqrt{k_Z^{(t)}(\mathbf{x}, \mathbf{x})}$, Φ and ϕ are the normal cumulative distribution and density functions, respectively. The second term, $\mathbb{P}(C^{(t)}(\mathbf{x}) \leq 0)$, can be approximated with available numerical methods (see details in Section 5.2).

Definition of the current feasible minimum $z_{\min}^{\mathbf{feas}}$

To solve problem (3.1), we need to define $z_{\min}^{\mathrm{feas}}$. We extend the definition of the current minimum for a non-observed process Z introduced in [23] to a problem with constraints. $z_{\min}^{\mathrm{feas}}$ is defined as the

minimum of the mean of the process $Z^{(t)}$ such that the constraint is satisfied in expectation

$$z_{\min}^{\text{feas}} = \min_{\mathbf{x} \in \mathcal{X}_t} m_Z^{(t)}(\mathbf{x}) \text{ s.t. } \mathbb{E}[C^{(t)}(\mathbf{x})] \leq 0 . \tag{3.3}$$

Under Fubini's condition and as the constraints are conditionally independent given $\mathbf{x}$ and $\mathbf{u}$, the expectation of $C^{(t)}$ is an integral over the uncertain space of a product of univariate Gaussian cumulative distribution functions

$$\begin{aligned} \mathbb{E}[C^{(t)}(\mathbf{x})] &= \mathbb{E}\left[1 - \alpha - \int_{\mathbb{R}^m} \mathbb{1}_{\cap_{i=1}^l \{G_i^{(t)}(\mathbf{x},\mathbf{u}) \leq 0\}} \rho_{\mathbf{u}}(\mathbf{u}) d\mathbf{u} \right] \\ &= 1 - \alpha - \int_{\mathbb{R}^m} \prod_{i=1}^l \mathbb{E}[\mathbb{1}_{\{G_i^{(t)}(\mathbf{x},\mathbf{u}) \leq 0\}}] \rho_{\mathbf{u}}(\mathbf{u}) d\mathbf{u} \\ &= 1 - \alpha - \int_{\mathbb{R}^m} \prod_{i=1}^l \Phi\left(\frac{-m_{G_i}^{(t)}(\mathbf{x}, \mathbf{u})}{\sigma_{G_i}^{(t)}(\mathbf{x}, \mathbf{u})} \right) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u}. \end{aligned} \tag{3.4}$$

If there is no solution to problem (3.3), we choose the most feasible point in expectation,

$$z_{\min}^{\text{feas}} = m_Z^{(t)}(\mathbf{x}^{\text{mf}}) \text{ where } \mathbf{x}^{\text{mf}} = \arg\max_{\mathbf{x} \in \mathcal{X}_t} \int_{\mathbb{R}^m} \prod_{i=1}^l \mathbb{P}(G_i^{(t)}(\mathbf{x}, \mathbf{u}) \leq 0) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u}.$$

4. Extending the acquisition criterion

The sequential methodology introduced in Proposition 2.1, Equation (2.5), requires to choose a couple $(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$ such that the variance of the one-step-ahead feasible improvement is minimal, i.e.,

$$(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) = \arg \min_{(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) \in \mathcal{S}_{\mathcal{X}} \times \mathcal{S}_{\mathcal{U}}} \mathbb{VAR}\left(I^{(t+1)}(\mathbf{x}_{\text{targ}}) \, \mathbb{1}_{\{C^{(t+1)}(\mathbf{x}_{\text{targ}}) \leq 0\}} \right), \tag{4.1}$$

where $I^{(t+1)}$ and $C^{(t+1)}$ are the updated $I^{(t)}$ and $C^{(t)}$ taking into account the observations at the point $(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$. This choice increases the most the information provided at the current point of interest, $\mathbf{x}_{\text{targ}}$, where the information now includes both the probabilistic constraints and the improvement.

In [23], the authors noted that $\mathbf{x}_{t+1}$ is usually very close to $\mathbf{x}_{\text{targ}}$ so instead of looking for the couple $(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$, we assume that $\mathbf{x}_{t+1} = \mathbf{x}_{\text{targ}}$. This simplifies the optimization of the sampling criterion because it reduces the dimension of the minimization (Equation (4.1)). As the one-step-ahead variance of the feasible improvement is difficult to evaluate, a computationally more tractable quantity, built with a product of variances, is now proposed.

The sampling criterion

We introduce a new sampling criterion specific to the robust optimization Problem (2.1), denoted S for Sampling as it is used to generate a new value for $\mathbf{u}$. It is inspired by the variance of the one-step-ahead feasible improvement but easier to assess.

Proposition 4.1. *The sampling criterion, given a new observation at point $(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$, is*

$$\begin{aligned} S(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) &= \mathbb{VAR}(I^{t+1}(\mathbf{x}_{targ})) \int_{\mathbb{R}^m} \mathbb{VAR}(\mathbb{1}_{\cap_{i=1}^l \{G_i^{(t+1)}(\mathbf{x}_{targ},\mathbf{u}) \leq 0\}}) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u}, \\ &= \mathbb{VAR}((z_{\min}^{feas} - Z^{(t+1)}(\mathbf{x}_{targ}))^+) \int_{\mathbb{R}^m} \mathbb{VAR}(\mathbb{1}_{\cap_{i=1}^l \{G_i^{(t+1)}(\mathbf{x}_{targ},\mathbf{u}) \leq 0\}}) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u}. \end{aligned} \tag{4.2}$$

In Equation (4.2) the first part of the expression forces the sampling to reduce the uncertainty in the improvement value while the second part focuses on the averaged predicted feasibility variance. For the sake of calculation simplicity, it might seem preferable to replace the improvement by the

objective process $Z^{(t+1)}$ and the variance of feasibility by the variance of the constraints. However, such variances are not those of the quantities of interest. In optimization, it is not important to reduce the variance of the constraints if it is clear that points are feasible or infeasible. However it is important to reduce the variance when there is a large uncertainty on feasibility, which is achieved by considering the variance of the Bernoulli variable $\mathbb{1}_{\cap_{i=1}^{l}\{G_i^{(t+1)}(\mathbf{x}_{\text{targ}},\mathbf{u})\leq 0\}}$. In the same way, the variance of $Z^{(t+1)}$ in regions where it is clearly above the target does not matter, but the variance of the process where improvement is uncertain should be reduced.

The ideal sampling criterion for the optimization problem at hand, the variance of the feasible improvement (Equation (4.1)), is the variance of the product of the improvement times a feasibility with confidence. The criterion S bears some resemblance but is not equivalent to the variance of the feasible improvement. It is a product of variances, which is not equivalent to the variance of a product[1]. Furthermore, the second term in the product for S (Equation (4.2)) is an averaged variance of feasibility $\mathbb{1}_{\cap_{i=1}^{l}\{G_i^{(t+1)}(\mathbf{x}_{\text{targ}},\mathbf{u})\leq 0\}}$ as opposed to a variance of feasibility with $1-\alpha$ confidence, $\mathbb{1}_{C^{(t+1)}(\mathbf{x}_{\text{targ}})\leq 0}$.

So, we choose the best candidate uncertainty $\mathbf{u}_{t+1}$ as

$$\mathbf{u}_{t+1} = \arg\min_{\tilde{u}\in\mathcal{S}_{\mathcal{U}}} S(\mathbf{x}_{\text{targ}},\tilde{u}). \tag{4.3}$$

The calculation of both terms of S is now presented.

Calculation of $\mathbb{VAR}(I^{t+1}(\mathbf{x_{targ}}))$

The expression of $\mathbb{VAR}\left(I^{t+1}(\mathbf{x}_{\text{targ}})\right)$ given a new observation at point $(\tilde{\mathbf{x}},\tilde{\mathbf{u}})$ is

$$\begin{aligned}
\mathbb{VAR}\left(I^{t+1}(\mathbf{x}_{\text{targ}})\right) &= \mathbb{VAR}\left((z_{\min}^{\text{feas}} - Z^{(t+1)}(\mathbf{x}_{\text{targ}}))^{+}\right) \\
&= \mathbb{VAR}\left((z_{\min}^{\text{feas}} - Z(\mathbf{x}_{\text{targ}}))^{+} \,\Big|\, F(D^{(t)}) = f^{(t)}, F(\tilde{\mathbf{x}},\tilde{u}) = f(\tilde{\mathbf{x}},\tilde{u})\right)
\end{aligned} \tag{4.4}$$

The expression of the expected improvement in terms of PDF and CDF of Gaussian distribution is well-known. The closed-form expression of the *Variance of the Improvement* stated in Proposition 4.2 can be obtained using the expression of the generalized expected-improvement criterion introduced in [46]. For the convenience of the readers, we provide a complete proof in Appendix A.1.

Proposition 4.2 (The *Variance of the Improvement*)**.**

$$\mathbb{VAR}\left(I^{(s)}(\mathbf{x})\right) = EI^{(s)}(\mathbf{x})(z_{\min}^{feas} - m_Z^{(s)}(\mathbf{x}) - EI^{(s)}(\mathbf{x})) + (\sigma_Z^{(s)})^2(\mathbf{x})\Phi\left(\frac{z_{\min}^{feas} - m_Z^{(s)}(\mathbf{x})}{\sigma_Z^{(s)}(\mathbf{x})}\right),$$

where $\sigma_Z^{(s)}(\mathbf{x}) = \sqrt{k_Z^{(s)}(\mathbf{x},\mathbf{x})}$.

As $f(\tilde{\mathbf{x}},\tilde{\mathbf{u}})$ is unknown, we cannot apply Proposition 4.2 to Equation (4.4). We use that $F^{(t)}(\tilde{\mathbf{x}},\tilde{\mathbf{u}}) \sim \mathcal{N}\left(m_F^{(t)}(\tilde{\mathbf{x}},\tilde{\mathbf{u}}), k_F^{(t)}(\tilde{\mathbf{x}},\tilde{\mathbf{u}};\tilde{\mathbf{x}},\tilde{\mathbf{u}})\right)$ and the law of total variance

$$\begin{aligned}
\mathbb{VAR}\left(I^{t+1}(\mathbf{x}_{\text{targ}})\right) &= \mathbb{E}\left[\mathbb{VAR}\left((z_{\min}^{\text{feas}} - Z(\mathbf{x}_{\text{targ}}))^{+} \,\Big|\, F(D^{(t)}) = f^{(t)}, F(\tilde{\mathbf{x}},\tilde{u})\right)\right] \\
&\quad + \mathbb{VAR}\left[\mathbb{E}\left((z_{\min}^{\text{feas}} - Z(\mathbf{x}_{\text{targ}}))^{+} \,\Big|\, F(D^{(t)}) = f^{(t)}, F(\tilde{\mathbf{x}},\tilde{u})\right)\right]. \quad (4.5)
\end{aligned}$$

To compute Equation (4.5), notice that the inside of the brackets have closed-form expressions in terms of $m_Z^{(t+1)}$ and $\sigma_Z^{(t+1)}$, which are given by Proposition 4.2 and Equation (3.2) for the first and second brackets, respectively. The external $\mathbb{E}$ and $\mathbb{VAR}$ only concern $m_Z^{(t+1)}(\mathbf{x}_{\text{targ}})$ whose randomness

---

[1]Let U and V be two random variables, $U \perp V$, $\mathbb{VAR}(UV) = \mathbb{E}U^2\mathbb{E}V^2 - (\mathbb{E}U)^2(\mathbb{E}V)^2$.
$\mathbb{VAR}U\,\mathbb{VAR}V = (\mathbb{E}U^2 - (\mathbb{E}U)^2)(\mathbb{E}V^2 - (\mathbb{E}V)^2) = \mathbb{E}U^2\mathbb{E}V^2 - (\mathbb{E}U)^2(\mathbb{E}V^2 - (\mathbb{E}V)^2) - \mathbb{E}U^2(\mathbb{E}V)^2$. The variance of the product is the product of the variances only if $\mathbb{E}U = \mathbb{E}V = 0$ when $\mathbb{VAR}(UV) = \mathbb{VAR}U\,\mathbb{VAR}V = \mathbb{E}U^2\mathbb{E}V^2$.

comes from $F(\tilde{\mathbf{x}}, \tilde{u})$, $\sigma_Z^{(t+1)}(\mathbf{x}_{\text{targ}})$ being not dependent on the evaluation of the function. It is proved in Appendix A.2 that $m_Z^{(t+1)}(\mathbf{x}_{\text{targ}})$ follows

$$m_Z^{(t+1)}(\mathbf{x}_{\text{targ}}) \sim \mathcal{N}\left(m_Z^{(t)}(\mathbf{x}_{\text{targ}}), \left(\frac{\int_{\mathbb{R}^m} k_F^{(t)}(\mathbf{x}_{\text{targ}}, \mathbf{u}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}}) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u}}{\sqrt{k_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})}}\right)^2\right). \tag{4.6}$$

Finally, the external $\mathbb{E}$ and $\mathbb{VAR}$ in Equation (4.5) are numerically evaluated. Details are given in Section 5.2.

Calculation of $\int_{\mathbb{R}^m} \mathbb{VAR}\left(\mathbb{1}_{\cap_{i=1}^l \{G_i^{(t+1)}(\mathbf{x}_{\text{targ}}, \mathbf{u}) \leq 0\}}\right) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u}$

Regarding the second term, we follow the *Kriging Believer* principle, leading to suppose that $\forall i = 1, \ldots, l$; $m_{G_i}^{(t+1)}(\mathbf{x}_{\text{targ}}, \mathbf{u}) = m_{G_i}^{(t)}(\mathbf{x}_{\text{targ}}, \mathbf{u})$. Under the hypothesis of independence of the constraint GPs, we have

$$\int_{\mathbb{R}^m} \mathbb{VAR}\left(\mathbb{1}_{\cap_{i=1}^l \{G_i^{(t+1)}(\mathbf{x}_{\text{targ}}, \mathbf{u}) \leq 0\}}\right) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u} = \int_{\mathbb{R}^m} p(\mathbf{u})(1 - p(\mathbf{u})) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u},$$

where

$$p(\mathbf{u}) = \prod_{i=1}^l \Phi\left(\frac{-m_{G_i}^{(t)}(\mathbf{x}_{\text{targ}}, \mathbf{u})}{\sqrt{k_{G_i}^{(t+1)}(\mathbf{x}_{\text{targ}}, \mathbf{u}, \mathbf{x}_{\text{targ}}, \mathbf{u})}}\right),$$

and $k_{G_i}^{(t+1)}(\mathbf{x}_{\text{targ}}, \mathbf{u}, \mathbf{x}_{\text{targ}}, \mathbf{u}) = k_{G_i}^{(t)}(\mathbf{x}_{\text{targ}}, \mathbf{u}, \mathbf{x}_{\text{targ}}, \mathbf{u}) - \frac{(k_{G_i}^{(t)}(\mathbf{x}_{\text{targ}}, \mathbf{u}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}}))^2}{k_{G_i}^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})}$ (cf. [13]). Further details about the numerical estimation of the above integral are given in Section 5.2.

The steps of the proposed methodology are summarized in Algorithm 1, called EFISUR for Expected Feasible Improvement with Stepwise Uncertainty Reduction sampling.

Algorithm 1 : EFISUR(Expected Feasible Improvement with Stepwise Uncertainty Reduction sampling)

Create an initial Design of Experiments (DoE) of size t in the joint space and calculate simulator responses:
$D^{(t)} = \{(\mathbf{x}_i, \mathbf{u}_i) , i = 1, \ldots, t\}$, and associated $f^{(t)}$ and $g_i^{(t)}$
while $t \leq$ maximum budget **do**
    Create the GPs of the objective and the constraints in the joint space: $F^{(t)}$ and $(G_i^{(t)})_{i=1}^l$
    Calculate the GP of the mean objective, $Z^{(t)}$, in the search space $\mathcal{S}_{\mathcal{X}}$
    Optimize EFI to define $\mathbf{x}_{\text{targ}} = \arg\max_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} \text{EFI}^{(t)}(\mathbf{x})$ (Eq. (3.1))
    Set $\mathbf{x}_{t+1} = \mathbf{x}_{\text{targ}}$
    Sample the next uncertain point by solving
    $\mathbf{u}_{t+1} = \arg\min_{\tilde{u} \in \mathcal{S}_{\mathcal{U}}} S(\mathbf{x}_{\text{targ}}, \tilde{u})$ (Eq. (4.3))
    Calculate simulator responses at the next point $(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$
    Update the DoE: $D^{(t+1)} = D^{(t)} \cup (\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$, $f^{(t+1)} = f^{(t)} \cup f(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$,
    $g_i^{(t+1)} = g_i^{(t)} \cup g_i(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) , i = 1, \ldots, l, t \leftarrow t+1$
end while

---

5. Numerical experiments

In this section the performance of the EFISUR method is studied first on an analytical test case, and then on an industrial application. The results are compared to two alternative procedures which are described below.

The code and data generated or used during the current study are available in the first author's GitHub repository [16].

5.1. **Competing algorithms**

Two algorithms serve as bases for comparison for `EFISUR`. To the best of our knowledge, although inspired by previous works, these algorithms are new. First, the `EFIrand`algorithm is identical to `EFISUR`to the exception of the point $\mathbf{u}_{t+1}$ which is simply sampled from its distribution. This `EFIrand`algorithm will be useful to judge the usefulness of the sampling criterion S in `EFISUR`.

Algorithm 2 : `EFIrand`Expected Feasible Improvement with random sampling

Create an initial DoEof size t in the joint space and calculate simulator responses:
$D^{(t)} = \{(\mathbf{x}_i, \mathbf{u}_i)\ ,\ i = 1, \dots, t\}$, and associated $f^{(t)}$ and $g_i^{(t)}$
while $t \leq$ maximum budget **do**
    Create the GPs of the objective and the constraints in the joint space: $F^{(t)}$ and $(G_i^{(t)})_{i=1}^l$
    Calculate the GP of the mean objective, $Z^{(t)}$, in the search space $\mathcal{S}_{\mathcal{X}}$
    Optimize EFIto define $\mathbf{x}_{t+1} = \arg\max_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} \text{EFI}^{(t)}(\mathbf{x})$ (Eq. (3.1))
    Sample the next uncertain point randomly, $\mathbf{u}_{t+1} \sim \rho_{\mathbf{U}}$
    Calculate simulator responses at the next point $(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$
    Update the DoE: $D^{(t+1)} = D^{(t)} \cup (\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$, $f^{(t+1)} = f^{(t)} \cup f(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$,
    $g_i^{(t+1)} = g_i^{(t)} \cup g_i(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})\ ,\ i = 1, \dots, l,\ t \leftarrow t + 1$
end while

The other algorithm to which `EFISUR`will be compared uses the quantile as an alternative way of measuring the failure probability. For the first constraint, the quantile writes,

$$\mathbb{P}(g_1(\mathbf{x}, \mathbf{U}) \leq 0) \geq 1 - \alpha \iff q_{1-\alpha}(g_1(\mathbf{x}, \mathbf{U})) \leq 0.$$

When dealing with several constraints, $\mathbb{P}(g(\mathbf{x}, \mathbf{U}) \leq 0)$ is then replaced by individuals constraints with quantile level $1 - \alpha/l$. The quantile of the constraint $i = 1, \dots, l$ is approximated using the predictive mean of the GP model $G^{(t)}$,

$$q_{1-\alpha/l}(g_i(\mathbf{x}, \mathbf{U})) \approx q_{1-\alpha/l}(m_{G_i}^{(t)}(\mathbf{x}, \mathbf{U})). \tag{5.1}$$

Further implementation details about the quantile estimation are given in Section 5.2.

In order to choose the next point in the design space, $\mathbf{x}_{t+1}$, this last competing methodology maximizes the expected improvement under constraints about the empirical quantiles:

$$\begin{aligned} \mathbf{x}_{t+1} = \arg\max_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} \text{EI}^{(t)}(\mathbf{x}) \\ \text{such that } \forall i \in \{1, \dots, l\},\ q_{1-\alpha/l}(m_{G_i}^{(t)}(\mathbf{x}, \mathbf{U})) \leq 0. \end{aligned} \tag{5.2}$$

The sampling in the uncertain space is based on the *deviation number* developed in [15, 18] for the Active Kriging Monte Carlo simulation technique. In these works, the uncertainty that is most likely to improve the GP is the one that minimizes the following DN(Deviation Number) function,

$$\text{DN}_i(\mathbf{u}) = \frac{|m_{G_i^{(t)}}(\mathbf{x}_{t+1}, \mathbf{u})|}{\sigma_{G_i}^{(t)}(\mathbf{x}_{t+1}, \mathbf{u})}. \tag{5.3}$$

The points that have a low deviation number are either close to the constraint threshold (null in Equation (5.3)), or they have a high GP variance. To handle multiple constraints, the constraint with

the minimum value of DN is selected (as in [30]),

$$\mathrm{DN}_c(\mathbf{u}) = \min_{i=1,\ldots,l} \mathrm{DN}_i(\mathbf{u}). \tag{5.4}$$

The whole algorithm is called `cEIdevNum` for Constrained EI plus Deviation Number. The `cEIdevNum` algorithm is an alternative to `EFISUR` for handling chance constraints in the augmented space. During the optimization step for $\mathbf{x}_{t+1}$, the constraints are handled explicitly through an ancilliary constrained optimization algorithm, as opposed to the EFI that aggregates them in a single objective. However, the reliability is defined independently for each constraint, which is not equivalent to the real problem given in Equation (2.1). The sampling step (the DN minimization) accounts only for the constraints satisfaction and not for the objective function. The `cEIdevNum` algorithm bears some resemblance to the approach described in [31]: in both cases, the reliability constraint is estimated with a kriging model used to estimate quantiles, and sampling occurs through the minimization of the deviation. The generalization of EGO to constrained problems by approximating the constraints through kriging models and keeping them separated from the objective, as it is done in Equation (5.2), can be found in other papers, e.g., [6]. However, to the authors' knowledge, `cEIdevNum` integrates these techniques within an EGO-like algorithm in an original manner.

Algorithm 3 : `cEIdevNum`

Create an initial DoE of size t in the joint space and calculate simulator responses:
$D^{(t)} = \{(\mathbf{x}_i, \mathbf{u}_i) \,,\ i = 1, \ldots, t\}$, and associated $f^{(t)}$ and $g_i^{(t)}$
while $t \leq$ maximum budget **do**

    Create the GPs of the objective and the constraints in the joint space: $F^{(t)}$ and $(G_i^{(t)})_{i=1}^l$
    Calculate the GP of the mean objective, $Z^{(t)}$, in the search space $\mathcal{S}_{\mathcal{X}}$
    Optimize the expected improvement under quantile constraints to determine the next iterate

$$\mathbf{x}_{t+1} = \arg \max_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} \mathrm{EI}^{(t)}(\mathbf{x})$$

    such that $\forall i \in \{1, \ldots, l\}$, $q_{1-\alpha/l}(m_{G_i}^{(t)}(\mathbf{x}, \mathbf{U})) \leq 0$,   ($Eq.$ (5.2))

    Sample the next uncertainty by minimizing the deviation number,

$$\mathbf{u}_{t+1} = \arg \min_{\mathbf{u}} \mathrm{DN}_c(\mathbf{u})\qquad (Eq.\ (5.4))$$

    Calculate simulator responses at the next point $(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$
    Update the DoE: $D^{(t+1)} = D^{(t)} \cup (\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$, $f^{(t+1)} = f^{(t)} \cup f(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})$,
$g_i^{(t+1)} = g_i^{(t)} \cup g_i(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) \,,\ i = 1, \ldots, l, t \leftarrow t + 1$
end while

5.2. Implementation details

Two strategies are adopted depending on the numerical cost of the integrals.

Common Random Numbers for $\mathbf{u}$ *samples*

All three algorithms include Monte Carlo simulations with respect to $\mathbf{u}$ samples. The efficiency of all three algorithms is enhanced by a common random numbers (CRN) technique. This means that the same seed is used to generate all random variables throughout the optimization. In particular, the same realizations of the uncertain variables $\{\mathbf{u}_1, \ldots, \mathbf{u}_M\}$, obtained by a quantization technique [33],

are considered in all iterations. The CRN produces more stable optimizations and reduces the variance of the estimated probabilities since the induced error is consistent within different designs. There is however a bias in the Monte Carlo estimations, which we keep small here by choosing relatively large Monte Carlo number of simulations.

More precisely, the `EFISUR`algorithm uses the CRN at different steps:

- In the `EFI`formula, the calculation of the quantity $\mathbb{P}(C^{(t)}(\mathbf{x}) \leq 0)$ is approximated by

$$\mathbb{P}(C^{(t)}(\mathbf{x}) \leq 0) \approx \frac{1}{N} \sum_{k=1}^{N} \mathbb{1}_{\left(1-\alpha-\frac{1}{M}\sum_{j=1}^{M} \mathbb{1}_{\left(G_i^{(t)}(\mathbf{x},\mathbf{u_j},\omega_k)\leq 0,\ i=1,\dots,l\right)} \leq 0\right)} \tag{5.5}$$

  where N realizations of the GPs are needed.

- In the $z_{\min}^{\text{feas}}$ formula, the calculation of $\mathbb{E}[C^{(t)}(\mathbf{x})]$ is approximated by :

$$\mathbb{E}[C^{(t)}(\mathbf{x})] \approx 1 - \alpha - \frac{1}{M} \sum_{j=1}^{M} \prod_{i=1}^{l} \Phi\left(\frac{-m_{G_i}^{(t)}(\mathbf{x},\mathbf{u_j})}{\sigma_{G_i}^{(t)}(\mathbf{x},\mathbf{u_j})}\right).$$

- In the second term of the sampling criterion, $\int_{\mathbb{R}^m} p(\mathbf{u})(1-p(\mathbf{u}))\rho_{\mathbf{U}}(\mathbf{u})d\mathbf{u}$ is approximated by $\frac{1}{M}\sum_{j=1}^{M} p(\mathbf{u_j})(1-p(\mathbf{u_j}))$.

The experiments reported in this article have as default $N = 1000$ trajectories of the GPs and $M = 300$ common random numbers.

The `EFIrand`algorithm is identical to `EFISUR`to the exception of the sampling of the next uncertain point which is random, hence it has a lower computational complexity.

Concerning the `cEIdevNum`algorithm, the quantiles making the constraints, $q_{1-\alpha}(m_{G_i}^{(t)}(\mathbf{x},\mathbf{U})) \leq 0$, are approximated by the corresponding order statistic associated to the sample $\{m_{G_i}^{(t)}(\mathbf{x},\mathbf{u}_j)\}_{j=1}^{M}$. For sharper comparisons, we use the same seed (M = 300 common random numbers) to estimate the quantiles.

Computational complexity

Remember that t is the iteration number and the number of data points, $t \leq$ budget, M is the number of $\mathbf{u}$ samples and N is the number of GP trajectories in the simulations, l the number of constraint GPs, d and m the dimensions of $\mathbf{u}$ and $\mathbf{x}$, respectively. The memory usage of the GPs grows in $\mathcal{O}(t^2)$ and is essentially the same for the `EFISUR`, `EFIrand`and `cEIdevNum`algorithms. The details of the calculations of the time complexities of the three algorithms are provided in Appendix B. The time complexities of the `EFISUR`and `EFIrand`algorithms are identical and of the order of $\max[\mathcal{O}(\text{budget}^3) \times (l+1) \times (d+m)\ ,\ d \times N \times \mathcal{O}(M^3)]$, where the first term in $\mathcal{O}(\text{budget}^3)$ comes from learning the GPs in the augmented space while the second term in $\mathcal{O}(M^3)$ comes from calculating the EFI. Learning the GPs is the critical component of `cEIdevNum`which therefore is computationally cheaper than `EFISUR`and `EFIrand`in most situations where $M > \text{budget}$. For large budgets however, the cost of GPs construction will dominate that of the EFI, making all three algorithms equivalent in terms of complexity.

Quantization for $m_Z^{(t+1)}$ samples

The calculation of the first term of the criterion S requires realizations of $m_Z^{(t+1)}$ (see Equation (4.5)). To this end, we approximate the continuous distribution of $m_Z^{(t+1)}$ (given in Equation (4.6)) by a discrete one. As $m_Z^{(t+1)}$ follows a Gaussian distribution, we approximate it using the inverse transformation of the Sobol sequence. Thus, the external variance and expectation of Equation (4.5) are

discretized at values representing the distribution of $m_Z^{(t+1)}$. In the upcoming experiments, a quantizer of size 20 is chosen.

Internal optimizations

The EFISUR algorithm and its two competitors involve several internal optimization problems that must be solved at every iteration. Some of these problems are unconstrained: the maximization of EFI with respect to $\mathbf{x}$ (EFISUR and EFIrand algorithms) or the minimization of S or U_c with respect to $\mathbf{u}$ (EFISUR and cEIdevNum). Such internal continuous unconstrained optimization problems are handled with the derivative-free solver BOBYQA [39].

The Problem (5.2) in the algorithm cEIdevNum further requires a solver that can handle constraints. It is addressed with the COBYLA program [38].

5.3. Analytical test case

A first test-case is now considered to compare the three competing algorithms in moderate dimensions. A fully random search, Random, is added for reference. The problem has two design variables, two uncertain parameters and a single reliability constraint:

$$\text{minimize} \quad \mathbb{E}[f(\mathbf{x}, \mathbf{U})]$$
$$\text{such that} \quad \mathbb{P}(g(\mathbf{x}, \mathbf{U}) \leq 0) \geq 1 - \alpha$$
$$\text{where } f(\mathbf{x}, \mathbf{u}) = 5(x_1^2 + x_2^2) - (u_1^2 + u_2^2) + x_1(u_2 - u_1 + 5) + x_2(u_1 - u_2 + 3)$$
$$g(\mathbf{x}, \mathbf{u}) = -x_1^2 + 5x_2 - u_1 + u_2^2 - 1$$
$$\text{with } \mathbf{x} \in [-5, 5]^2$$
$$\mathbf{U} \sim \mathcal{U}([-5, 5]^2)$$

By setting the target probability of failure to $\alpha = 0.05$, the computed reference solution is $\mathbf{x}^* = (-3.62069, -1.896552)$. This reference was found semi-analytically because some of the calculations are manually tractable in the above problem. Figure 5.1 shows the contour plots of the functions $\mathbb{E}[f(\,\cdot\,, \mathbf{U})]$ and $\mathbb{P}(g(\,\cdot\,, \mathbf{U}) \leq 0)$ obtained from a 40×40 grid experiments, where at each grid point the expectation and the probability are approximated by a Monte Carlo method over 10^4 realizations of $\mathbf{U}$.

To account for the inherent statistical variability of the algorithms, the runs are repeated 30 times for each method. The inital Design of Experiments of each method is a random Latin hypercube of $4 + (d + m) = 8$ points. An additional budget of 56 iterations[2] is used as a stopping criterion. The default Gaussian Process has a Matérn 5/2 covariance function and a constant trend. The performance of the various methods is measured by the average Euclidean distance between the optimum given by the method and the true minimum, at each iteration. The distance to the solution, averaged over the 30 runs, is plotted in Figure 5.2.

EFISUR and EFIrand converge faster to the optimum than cEIdevNum. After 40 iterations, EFISUR approaches the solution more closely than EFIrand does. A complementary view of the convergence, with dispersions, can be found in Figure 5.3 which shows the boxplots of the distances to the reference solution at iterations 15 (left panel), 40 (middle) and 60 (right). It is observed that EFISUR leads to an accurate solution from 40 iterations onwards with a small deviation between the runs. At all iterations, cEIdevNum has a larger mean distance to the solution (but the median is better at 60 iterations) and, mainly, a larger spread in results. Figure 5.4 shows the "a posteriori" probability, calculated by Monte Carlo, of satisfying the constraint at the current point seen as feasible by the

---

[2] An iteration encompasses one call to the objective function and one call to all the constraints.
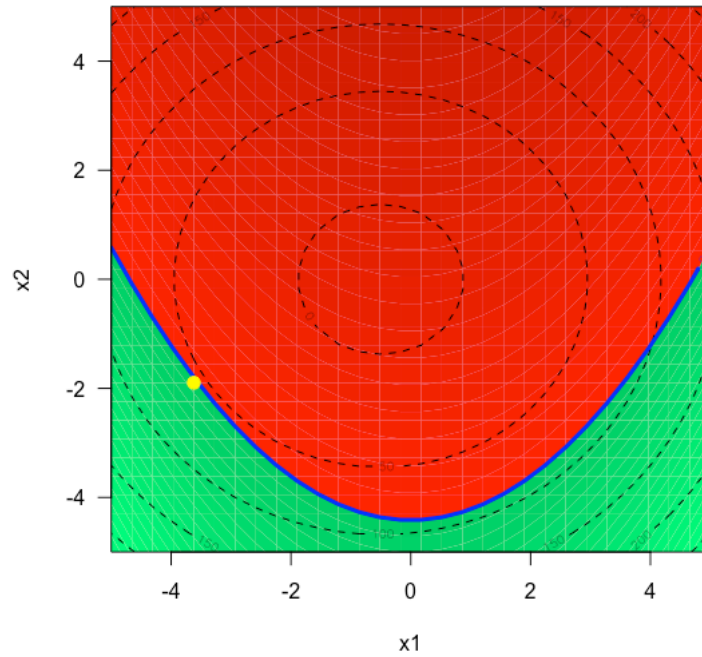
FIGURE 5.1. Contour plots of $\mathbb{P}(g(\,\cdot\,, \mathbf{U}) \leq 0)$ and $\mathbb{E}[f(\,\cdot\,, \mathbf{U})]$: failure and feasible regions in red and green, respectively, the limit-state function in blue, objective function in dashed black lines. The solution is the yellow bullet.
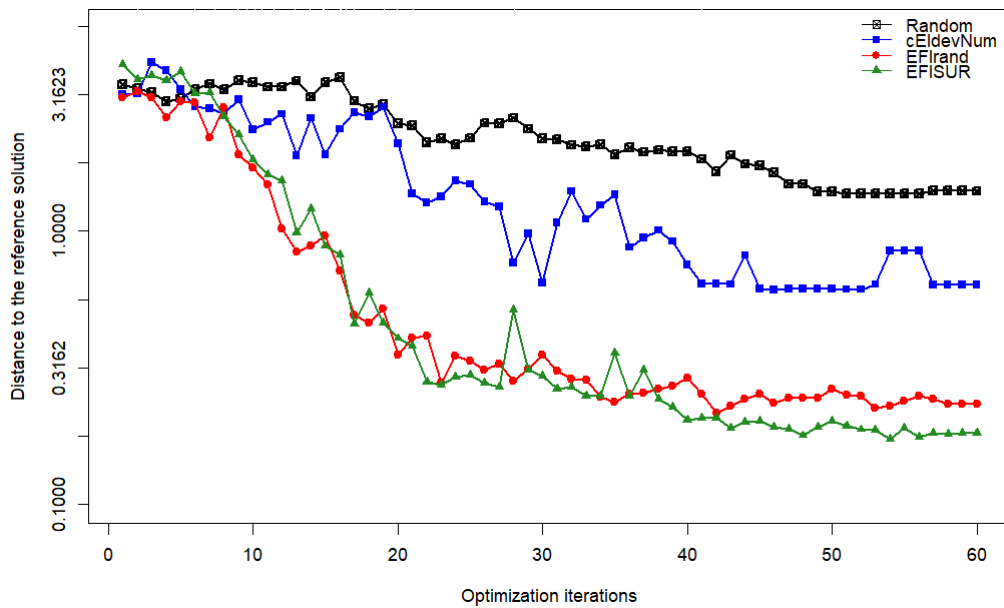

FIGURE 5.2. Mean convergence rates (Euclidean distance to the solution) on the analytical test case. The `EFISUR` method is plotted with green triangles, `EFIrand` with red rounds, `cEIdevNum` with blue squares and `Random` with the black crosses. The initial DoE which precedes these iterations is not represented.

EFISURand EFIrandstrategies at different iterations. Unlike EFIrandalgorithm, starting from iteration 25, all the points proposed by the EFISURalgorithm satisfy the probabilistic constraint. Thus, the points proposed by EFISURare not only close to the reference solution, but they are also inside the safe region.
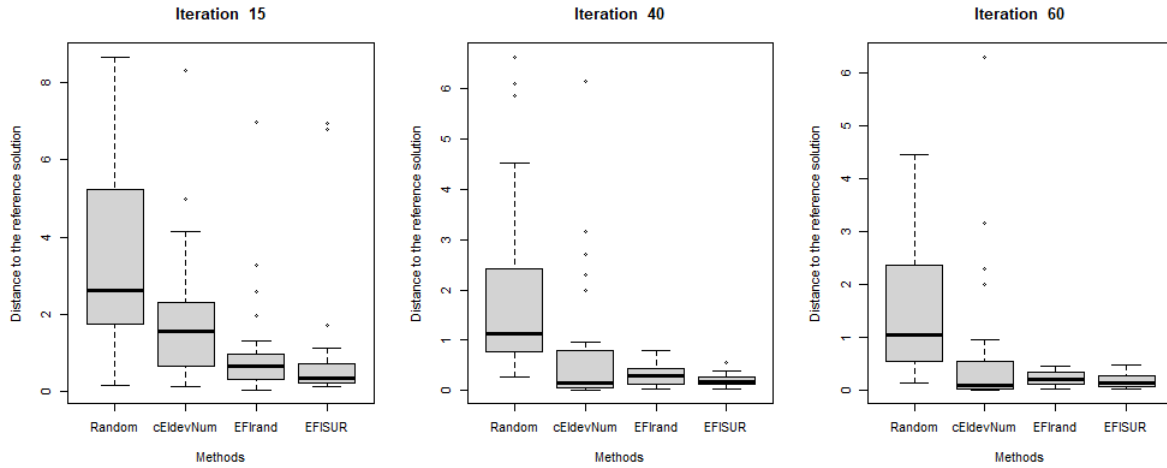

FIGURE 5.3. Distance to the reference solution at iteration 15 (left panel), 40 (middle panel) and 60 (right panel) for the three strategies. The boxplots summarize 30 replications of the runs.
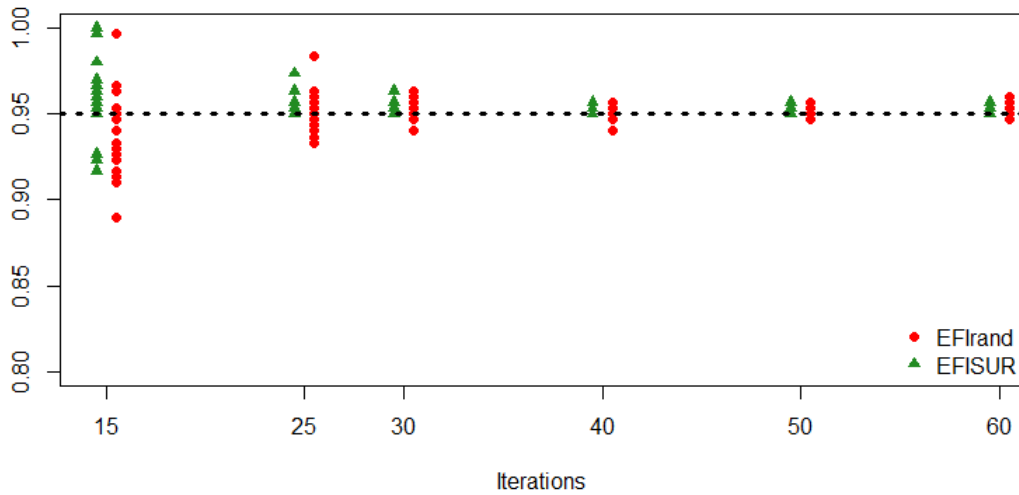

FIGURE 5.4. A posteriori probability of satisfying the constraint at the current "feasible" point $z_{\min}^{\text{feas}}$ for different iterations of the EFISURand EFIrandstrategies.

Figure 5.5 shows the enrichment in the uncertain space $\mathcal{S}_{\mathbf{U}}$ for all methods and all runs. It is clearly visible that EFIrand, in the middle plot, samples the $\mathbf{u}$'s randomly. The Random search does the same, of course. EFISUR(left) and cEIdevNum(right) both sample large $|u_2|$'s because they contribute to constraint violation irrespectively of $\mathbf{x}$ through the "$+u_2^2$" term (cf. $g(\mathbf{x}, \mathbf{u})$ expression above). In addition, cEIdevNumsamples large values $|u_1|$'s for varied u_2's because they are on the edge of the domain where the kriging variance is large (hence DNsmall). EFISURbreaks this symmetry and more

sparingly tries small (negative) u_1's associated to varied u_2's because its criterion also accounts for low objective function values: in the optimal region, $\mathbf{x}^* \approx (-3.6, -1.9)$, a small negative u_1 provides improvement through the term "$-x_1 u_1$".
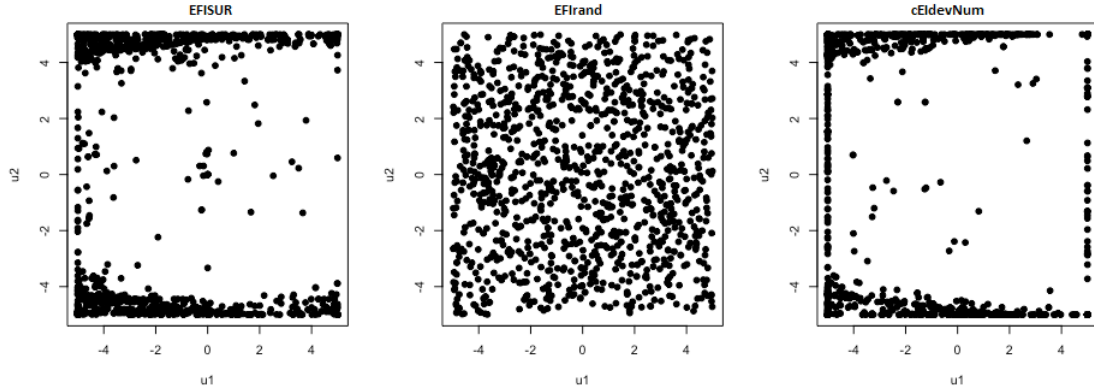

FIGURE 5.5. Enrichment in the uncertain space, $\mathcal{S}_{\mathbf{U}}$, for the three methods.

Two partial conclusions can be presumed from this set of experiments. First, the algorithms that rely on the EFIaggregated criterion to choose the next set of design variables, i.e., `EFIrand`and `EFISUR`, converge better than `cEIdevNum`and its constrained problem handled through `COBYLA`. Second, `EFISUR`provides an additional gain in efficiency thanks to its sampling criterion S which properly mixes the uncertainty about constraint satisfaction and improvement.

5.4. Industrial test case

We now report the application of the `EFISUR`method to an aeronautical test case. The NASA rotor 37 is a representative transonic axial-flow compressor that has been used extensively in the computational fluid dynamics (CFD) community to test optimization algorithms and validate CFD codes (see [22]). The optimization of the NASA rotor 37 compressor blade is a challenging test case primarily because of its high dimensionality: it has 20 design variables and 7 uncertain parameters. As such, to the best of the author's knowledge, such optimization has never been attempted using global metamodels. Furthermore, the design of the NASA rotor 37 compressor blade is highly nonlinear. And, as is common in CFD, each evaluation of the optimization criteria involves costly finite elements analyses. Formally, the optimization problem reads as follows:

$$\text{minimize: } \mathbb{E}[f(\mathbf{x}, \mathbf{U})]$$
$$\text{satisfying: } \mathbb{P}(g_i(\mathbf{x}, \mathbf{U}) \leq 0, \forall i \in \{1, \ldots, 5\}) \geq 1 - \alpha$$
$$\text{with } \alpha = 5\%,$$
$$\mathbf{x} \in \mathcal{S}_{\mathcal{X}} \quad , \quad \mathcal{S}_{\mathcal{X}} = [0,1]^{20} \subset \mathbb{R}^{20},$$
$$\mathbf{U} \sim \mathcal{U}(\mathcal{S}_{\mathcal{U}}), \quad , \quad \mathcal{S}_{\mathcal{U}} = [0,1]^7 \subset \mathbb{R}^7.$$

Because of the dimensionality and the numerical cost, the use of surrogate models is the only path to perform such an optimization. To highlight the importance of carefully sampling in the joint-space during the optimization, we compare our approach to the `EFIrand`algorithm. For comparison purposes, we also carry out a random search. All the methods are started with a DoEof 100 points drawn from an optimal Latin Hypercube Sampling. The enrichment then proceeds by adding 137 points, for a total of 237 model evaluations. Figure 5.6 shows the convergences of the feasible minimum in terms of

mean objective function. We first note that the `EFIrand` and `EFISUR` methods very clearly give access to better designs than the random search does. We observe that, after a warm-up phase of about 140 analyses, `EFISUR` finds better designs than `EFIrand` does. This behavior is attributed to the only difference between `EFISUR` and `EFIrand`, that is the sampling in the uncertain space. Properly choosing the random variables $\mathbf{u}$ results in a more accurate estimation of the criterion of merit, the feasible improvement, in the target region, which translates into more frequent progress.
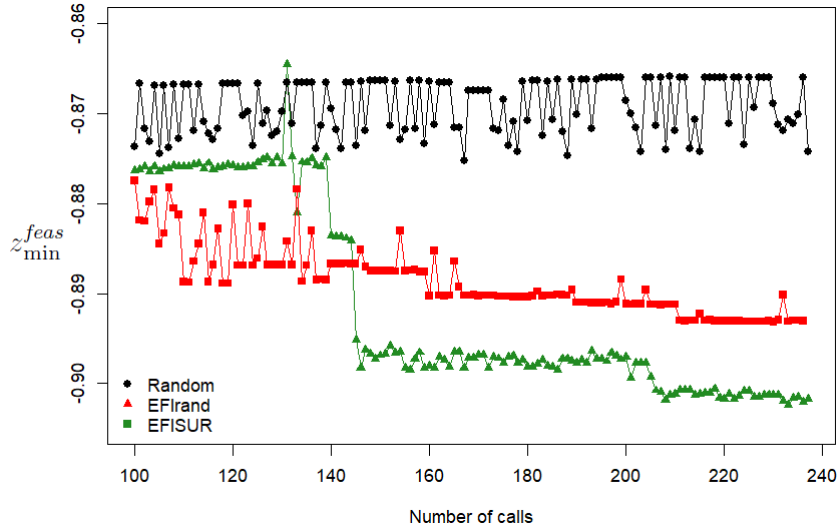

FIGURE 5.6. Convergence history of the average objective function of the current feasible minimum, $z_{\min}^{\text{feas}}$.

Figure 5.7 shows two distributions of the objective: first, in grey, all the estimated mean objective functions $m_Z^{(T)}(\mathbf{x})$ of the points evaluated during, both, the initial DoE and the optimization; second, in blue, the estimated mean objective of the best feasible point, $z_{\min}^{\text{feas}}$, during the optimization iterations. Note that the best feasible objective has shifted from -0.876 at the end of the DoE to -0.90174 after the optimization, representing 51.2% of the objective's range. All mean objectives shown on the Figure are evaluated with the last, most up-to-date, GPs.

Figure 5.8 shows, at the estimated feasible optimum, the relative value of each design variable with respect to its lower and upper bounds in polar coordinates. The largest radii are attributed to the variables which are close to their maximum allowable values, and vice versa.

Because it is generated by `EFISUR`, the above result assumes that the final GPs are accurate enough to correctly predict the probability of constraints satisfaction. In order to validate the surrogate model accuracy in the vicinity of the limit-state surface, the calculation of the probability of being feasible (Equation (5.5)) is repeated 500 times with $M = 1000$ $\mathbf{u}$ samples and $N = 1000$ trajectories of the GPs. Figure 5.9 provides the statistics of the probability of constraints satisfaction through a boxplot. Accounting for the bootstrap standard deviation, the targeted probability (0.95) remains below the confidence interval of the estimated probabilities. Thus the final GP models of the constraints are deemed accurate enough. It is worth emphasizing that only 237 calculations of f and the g_i's have been necessary to solve this 27-dimensional optimization problem.

6. Concluding remarks

We have proposed a robust Bayesian optimization algorithm to solve computationally intensive chance constrained problems. The algorithm, called `EFISUR`, carefully models all available information with
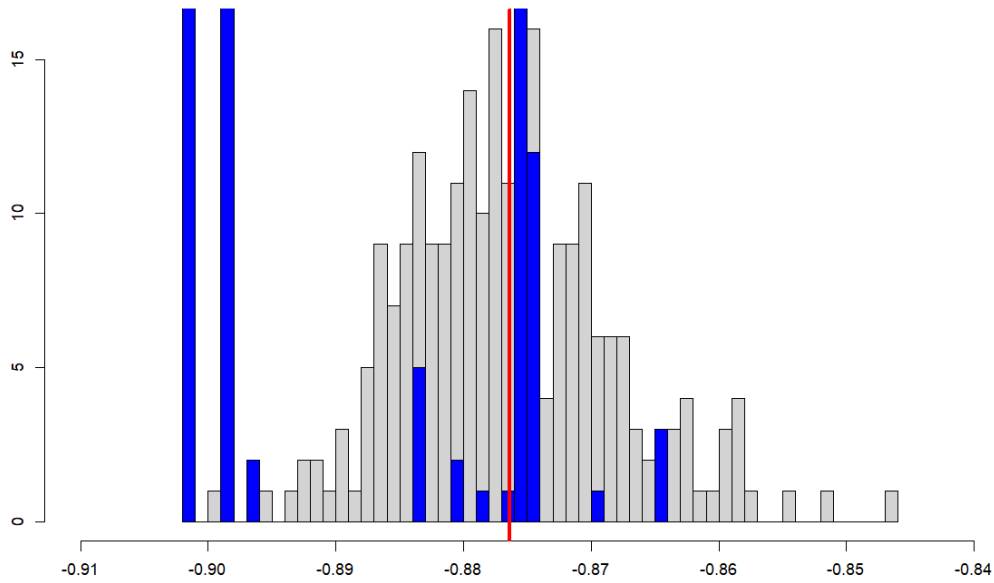
FIGURE 5.7. Histograms of *i)* in grey, the 237 estimated mean objectives $(m_Z^{(T)}(\mathbf{x}))$ of the initial DoE followed by the optimization, *ii)* in blue, the 137 mean objectives of the best feasible point generated during the optimization. The red line corresponds to the mean objective of the best feasible point at the first iteration.
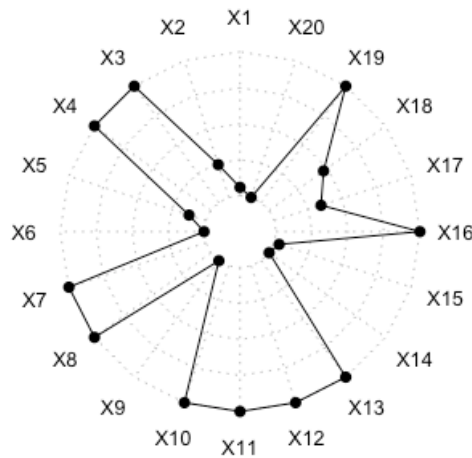

FIGURE 5.8. Relative coordinates of the optimal design with respect to their respective lower and upper bounds.

Gaussian processes built in the augmented space of the design variables and uncertain parameters. New calls to the objective and constraint functions are based on extrema of an acquisition criterion followed by a sampling criterion. The acquisition criterion is an expected feasible improvement that accounts for both the average improvement in objective function and the constraint reliability. The associated sampling criterion is inspired by the one-step-ahead variance reduction in feasible improvement and computationally tractable.

The article has detailed the analytical expressions of the acquisition and sampling criteria.
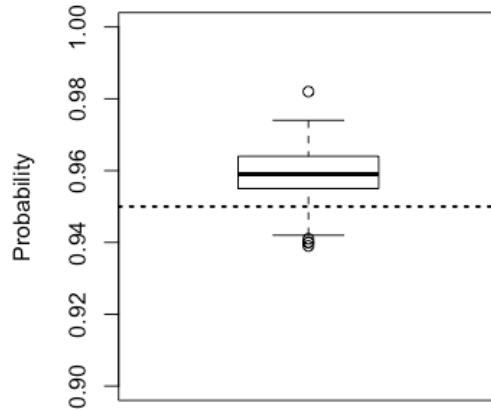
FIGURE 5.9. Distribution of constraints satisfaction probabilities at the optimal design computed with the final GPs. The boxplot summarizes 500 replications. The dashed line is the lower bound on constraint satisfaction (0.95).

EFISUR has been compared to two alternative algorithms which differ in the acquisition and sampling criteria. The results show a gain in favor of the expected feasible improvement and the proposed one-step-ahead variance reduction criterion. This set of criteria accounts for both the objective function and the constraints, it is opportunistic in the sense that it strives for feasible improvement at the next iteration, and both criteria (EFI and S) are coherent because they both relate to the feasible improvement. The sampling criterion follows a principle of maximal uncertainty reduction. The applicability of EFISUR to an industrial test case has also been checked.

From a methodological perspective, further work on the topic might seek to account for the correlation that typically exists between the constraint functions or between the objective and the constraint functions. This would potentially improve the overall Gaussian model of the optimization functions. It should also make it possible to assign priorities for evaluating the constraints.

Acknowledgements

We are very grateful to the three anonymous reviewers, whose comments improved the paper.

Code and data availability

The code and data generated or used during the current study are available in the first author's GitHub repository [16].

Appendix A. Expression of the aggregated variance sampling criterion

A.1. The *Variance of the Improvement*

The closed-form expression of the *Variance of the Improvement* stated in Proposition 4.2 can be obtained using the expression of generalized expected-improvement criterion introduced in [46]. For the convenience of the reader we still recall the details of the proof.

Proof.

$$VI(\mathbf{x}) = \mathbb{VAR}(I(\mathbf{x}))$$

$$= \mathbb{E}[I^2(\mathbf{x})] - \mathbb{E}[I(\mathbf{x})]^2$$

$$= \int_{-\infty}^{z_{\min}} (z_{\min} - z)^2 f_{\mathcal{N}(m_Z(\mathbf{x}), \sigma_Z^2(\mathbf{x}))}(z) dz - (\mathrm{EI}(\mathbf{x}))^2,$$

$$= \int_{-\infty}^{\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}} (z_{\min} - m_Z(\mathbf{x}) - \sigma_Z(\mathbf{x}).v)^2 \phi(v) dv - (\mathrm{EI}(\mathbf{x}))^2,$$

$$= \sigma_Z^2(\mathbf{x}) \int_{-\infty}^{\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}} v^2 \phi(v) dv + (z_{\min} - m_Z(\mathbf{x}))^2 \int_{-\infty}^{\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}} \phi(v) dv$$

$$\quad - 2\sigma_Z(\mathbf{x})(z_{\min} - m_Z(\mathbf{x})) \int_{-\infty}^{\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}} v\phi(v) dv - (\mathrm{EI}(\mathbf{x}))^2,$$

$$= \sigma_Z^2(\mathbf{x}) \int_{-\infty}^{\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}} v^2 \phi(v) dv + (z_{\min} - m_Z(\mathbf{x}))^2 \Phi\Big(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\Big)$$

$$\quad + 2\sigma_Z(\mathbf{x})(z_{\min} - m_Z(\mathbf{x}))\phi\Big(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\Big) - (\mathrm{EI}(\mathbf{x}))^2,$$

$$= \sigma_Z^2(\mathbf{x}) \Big([-v\phi(v)]_{-\infty}^{\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}} + \int_{-\infty}^{\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}} \phi(v) dv \Big)$$

$$\quad + (z_{\min} - m_Z(\mathbf{x}))^2 \Phi\Big(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\Big)$$

$$\quad + 2\sigma_Z(\mathbf{x})(z_{\min} - m_Z(\mathbf{x}))\phi\Big(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\Big) - (\mathrm{EI}(\mathbf{x}))^2,$$

$$= -\sigma_Z(\mathbf{x})(z_{\min} - m_Z(\mathbf{x}))\phi\Big(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\Big)$$

$$\quad + \sigma_Z^2(\mathbf{x})\Phi\Big(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\Big) + (z_{\min} - m_Z(\mathbf{x}))^2 \Phi\Big(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\Big)$$

$$\quad + 2\sigma_Z(\mathbf{x})(z_{\min} - m_Z(\mathbf{x}))\phi\Big(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\Big) - (\mathrm{EI}(\mathbf{x}))^2,$$

$$= \Big((z_{\min} - m_Z(\mathbf{x}))^2 + \sigma_Z^2(\mathbf{x}) \Big) \Phi\Big(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\Big)$$

$$\quad + \sigma_Z(\mathbf{x})(z_{\min} - m_Z(\mathbf{x}))\phi\Big(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\Big) - (\mathrm{EI}(\mathbf{x}))^2,$$

$$= \mathrm{EI}(\mathbf{x})\big(z_{\min} - m_Z(\mathbf{x}) - \mathrm{EI}(\mathbf{x})\big) + \sigma_Z^2(\mathbf{x})\Phi\Big(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\Big).$$

∎

Use was made of the expression of the expected improvement, EI, Equation (3.2). This shows that the *Variance of the Improvement* can be expressed in terms of PDF and CDF of $\mathcal{N}(0,1)$.

A.2. **Variance of the Improvement at step $t+1$**

Recall the expression of the variance of the improvement given in Equation (4.5) that results from the law of total variance,

$$
\begin{aligned}
\mathbb{VAR}(I^{(t+1)}(\mathbf{x}_{\text{targ}})) = {}& \mathbb{E}\left[\mathbb{VAR}\left((z_{\min}^{\text{feas}} - Z(\mathbf{x}_{\text{targ}}))^+ \,\middle|\, F(D^{(t)}) = f_t, F(\tilde{\mathbf{x}}, \tilde{u})\right)\right] \\
& + \mathbb{VAR}\left[\mathbb{E}\left((z_{\min}^{\text{feas}} - Z(\mathbf{x}_{\text{targ}}))^+ \,\middle|\, F(D^{(t)}) = f_t, F(\tilde{\mathbf{x}}, \tilde{u})\right)\right] \\
= {}& \mathbb{E}[VI^{(t+1)}(\mathbf{x}_{\text{targ}})|m_Z^{(t+1)}(\mathbf{x}_{\text{targ}})] + \mathbb{VAR}(\text{EI}^{(t+1)}(\mathbf{x}_{\text{targ}})|m_Z^{(t+1)}(\mathbf{x}_{\text{targ}})]).
\end{aligned}
$$

The last line is because the improvement variance and expectation at $t+1$ are closed-form expressions of $m_Z^{(t+1)}$ and $\sigma_Z^{(t+1)}$ and only $m_Z^{(t+1)}$ is random through $F(\tilde{\mathbf{x}}, \tilde{u})$.

It is now proved that the one-step-ahead mean of F, $m_Z^{(t+1)}$, is also Gaussian. The proof makes use of the one step update formula which is found in [12]:

$$
\begin{cases}
m_F^{(t+1)}(\mathbf{x}, \mathbf{u}) = m_F^{(t)}(\mathbf{x}, \mathbf{u}) + \dfrac{k_F^{(t)}(\mathbf{x}, \mathbf{u}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})}{k_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})}(F(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) - m_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})), \\
k_F^{(t+1)}(\mathbf{x}, \mathbf{u}, \mathbf{x}', \mathbf{u}') = k_F^{(t)}(\mathbf{x}, \mathbf{u}, \mathbf{x}', \mathbf{u}') - \dfrac{k_F^{(t)}(\mathbf{x}, \mathbf{u}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})k_F^{(t)}(\mathbf{x}', \mathbf{u}'; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})}{k_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})},
\end{cases}
$$

where $((\tilde{\mathbf{x}}, \tilde{\mathbf{u}}); F(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}))$ is the new, yet still random, observation.

$$
\begin{aligned}
m_Z^{(t+1)}(\mathbf{x}_{\text{targ}}) ={}& \int_{\mathbb{R}^m} m_F^{(t+1)}(\mathbf{x}_{\text{targ}}, \mathbf{u})\rho_{\mathbf{U}}(\mathbf{u})d\mathbf{u}, \\
={}& \int_{\mathbb{R}^m}\left(m_F^{(t)}(\mathbf{x}_{\text{targ}}, \mathbf{u}) + \frac{k_F^{(t)}(\mathbf{x}_{\text{targ}}, \mathbf{u}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})}{k_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})}(F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) - m_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}))\right)\rho_{\mathbf{U}}(\mathbf{u})d\mathbf{u}, \\
={}& m_Z^{(t)}(\mathbf{x}_{\text{targ}}) + \frac{F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) - m_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})}{\sqrt{k_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})}}\frac{\int_{\mathbb{R}^m} k_F^{(t)}(\mathbf{x}_{\text{targ}}, \mathbf{u}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})\rho_{\mathbf{U}}(\mathbf{u})d\mathbf{u}}{\sqrt{k_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})}}, \\
\sim{}& \mathcal{N}\left(m_Z^{(t)}(\mathbf{x}_{\text{targ}}), \left(\frac{\int_{\mathbb{R}^m} k_F^{(t)}(\mathbf{x}_{\text{targ}}, \mathbf{u}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})\rho_{\mathbf{U}}(\mathbf{u})d\mathbf{u}}{\sqrt{k_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})}}\right)^2\right).
\end{aligned}
$$

Appendix B. **Time complexity calculation details**

Again, recall that t is the iteration number and the number of data points, $t \le$ budget, M is the number of $\mathbf{u}$ samples and N is the number of GP trajectories in the simulations, l the number of constraint GPs, d and m the dimensions of $\mathbf{u}$ and $\mathbf{x}$, respectively. The steps for the derivation of the time complexity of the three algorithms are detailed below. It is assumed that the cost of the various internal optimizations is linear in the search space dimension.

- Learning of the $(l+1)$ GPs: $\mathcal{O}(t^3) \times (l+1) \times (d+m)$, where the product by $(d+m)$, the input dimension size, accounts for the internal likelihood maximization.

- Prediction of a GP (mean, variance, EI) at a point, assuming the covariance matrix was decomposed and inverted before: $\mathcal{O}(t^2)$

- Sampling of a GP trajectory at M points: $\mathcal{O}(M^3)$.

- $\mathbb{P}(C^{(t)}(\mathbf{x}) \le 0)$: it costs N trajectories at M points that is $N \times \mathcal{O}(M^3)$.

- $\mathbb{E}[C^{(t)}(\mathbf{x})]$ for the calculation of $z_{\min}^{\text{feas}}$: $M \times l$ predictions, i.e., $M \times l \times \mathcal{O}(t^2)$.

- One calculation of EFI: the sum of the 2 aboves items plus the EI calculation, $N \times \mathcal{O}(M^3) + M \times l \times \mathcal{O}(t^2)$

- The maximization of EFI: the input dimension times the above complexity but $z_{\min}^{\text{feas}}$ is calculated only once, $d \times N \times \mathcal{O}(M^3) + M \times l \times \mathcal{O}(t^2)$, the dominant term is $d \times N \times \mathcal{O}(M^3)$.

- Variance of the improvement: N predictions, $N \times \mathcal{O}(t^2)$.

- Constraints variance, $\int_{\mathbb{R}^m} p(\mathbf{u})(1 - p(\mathbf{u}))\rho_{\mathbf{U}}(\mathbf{u})d\mathbf{u}$: $M \times l$ predictions, $M \times l \times \mathcal{O}(t^2)$.

- Sampling criterion: the sum of the 2 aboves items, $\max(N, M \times l) \times \mathcal{O}(t^2)$.

- Minimization of the sampling criterion: the dimension of the uncertainties times the calculation of the sampling criterion, $m \times \max(N, M \times l) \times \mathcal{O}(t^2)$.

The overall time complexity of `EFISUR` is $\max[\mathcal{O}(\text{budget}^3) \times (l + 1) \times (d + m) \ , \ d \times N \times \mathcal{O}(M^3)]$. The complexity of `EFIrand` is the same as that of `EFISUR` because the GP learning and the EFI maximization which are common to both algorithms are driving it. The cost of the sampling criterion minimization is of second order. Now, the cost of the `cEIdevNum` algorithm is split as follows.

- Calculation of the quantiles : M predictions of l constraints, $M \times l \times \mathcal{O}(t^2)$.

- Calculation of the EI : one calculation of $z_{\min}^{\text{feas}}$ per iteration and a prediction of GP mean and variance.

- Maximization of the quantile constrained EI: $d \times (M \times l + 2) \times \mathcal{O}(t^2)$.

- Calculation of the deviation number, DN: $l \times 2 \times \mathcal{O}(t^2)$.

- Minimization of DN: $m \times l \times 2 \times \mathcal{O}(t^2)$.

The complexity of `cEIdevNum` is dominated by the cost of learning the GPs, $\mathcal{O}(\text{budget}^3) \times (l + 1) \times (d + m)$. If the number of uncertain parameters samples is larger than the budget of the GP regression, $M > \text{budget}$, then `cEIdevNum` will be faster to execute than `EFISUR` and `EFIrand`.

References

[1] Reda El Amri, Céline Helbert, Miguel Munoz Zuniga, Clémentine Prieur, and Delphine Sinoquet. Set inversion under functional uncertainties with joint meta-models. preprint available on HAL as document hal-02986558 (`https://hal-ifp.archives-ouvertes.fr/hal-02986558`), 2020.

[2] Laetitia Andrieu, Guy Cohen, and Felisa J. Vázquez-Abad. Gradient-based simulation optimization under probability constraints. *Eur. J. Oper. Res.*, 212(2):345–351, 2011.

[3] Kenneth Joseph Arrow, Hirofumi Azawa, Leonid Hurwicz, and Hirofumi Uzawa. *Studies in linear and non-linear programming*, volume 2. Stanford University Press, 1958.

[4] Anne Auger, Nikolaus Hansen, Verena Heidrich-Meisner, Olaf Mersmann, Petr Posik, and Mike Preuss. 3rd GECCO Workshop for Real-Parameter Optimization. `https://coco.gforge.inria.fr/doku.php?id=bbob-2012-results`, 2012.

[5] Mathieu Balesdent, Loïc Brevault, Jérôme Morio, and Rudy Chocat. Overview of Problem Formulations and Optimization Algorithms in the Presence of Uncertainty. In *Aerospace System Analysis and Optimization in Uncertainty*, pages 147–183. Springer, 2020.

[6] Nathalie Bartoli, Thierry Lefebvre, Sylvain Dubreuil, Romain Olivanti, Rémy Priem, Nicolas Bons, Joaquim RRA Martins, and Joseph Morlier. Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design. *Aerosp. Sci. Technol.*, 90:85–102, 2019.

[7] Julien Bect, David Ginsbourger, Ling Li, Victor Picheny, and Emmanuel Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Stat. Comput.*, 22(3):773–793, 2012.

[8] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

[9] Hans-Georg Beyer and Bernhard Sendhoff. Robust optimization–a comprehensive survey. *Comput. Methods Appl. Mech. Eng.*, 196(33-34):3190–3218, 2007.

[10] Jean-Marc Bourinet. *Reliability analysis and optimal design under uncertainty-Focus on adaptive surrogate-based approaches.* Habilitation à diriger les recherches, Univ. Clermont-Auvergne, 2018. `https://hal.archives-ouvertes.fr/tel-01737299`.

[11] Dimo Brockhoff, Anne Auger, Nikolaus Hansen, Dirk V. Arnold, and Tim Hohm. Mirrored sampling and sequential selection for evolution strategies. In *PPSN 2010: Parallel Problem Solving from Nature, PPSN XI*, pages 11–21. Springer, 2010.

[12] Clément Chevalier. *Fast uncertainty reduction strategies relying on Gaussian process models.* PhD thesis, University of Bern, 2013.

[13] Clément Chevalier, David Ginsbourger, and Xavier Emery. Corrected kriging update formulae for batch-sequential data assimilation. In *Mathematics of Planet Earth*, pages 119–122. Springer, 2014.

[14] Vincent Dubourg, Bruno Sudret, and Jean-Marc Bourinet. Reliability-based design optimization using kriging surrogates and subset simulation. *Struct. Multidiscip. Optim.*, 44(5):673–690, 2011.

[15] B. Echard, N. Gayton, and M. Lemaire. AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation. *Structural Safety*, 33(2):145–154, 2011.

[16] Reda El Amri. EFISUR, a new acquisition function. `https://github.com/elamrireda/EFISUR`, 2021.

[17] David Eriksson and Matthias Poloczek. Scalable constrained Bayesian optimization. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *PMLR*, pages 730–738. 2021.

[18] William Fauriat and Nicolas Gayton. AK-SYS: an adaptation of the AK-MCS method for system reliability. *Reliability Engineering & System Safety*, 123:137–144, 2014.

[19] Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. *Eur. J. Oper. Res.*, 235(3):471–483, 2014.

[20] Roman Garnett. *Bayesian Optimization.* Cambridge University Press, 2022.

[21] José Miguel Hernandez-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive Entropy Search for Bayesian Optimization with Unknown Constraints. In *ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, pages 1699–1707. JMLR, 2015.

[22] Charles Hirsch, Dirk Wunsch, Jacek Szumbarski, J. Pons-Prats, et al. *Uncertainty Management for Robust Industrial Design in Aeronautics.* Springer, 2019.

[23] Janis Janusevskis and Rodolphe Le Riche. Simultaneous kriging-based estimation and optimization of mean response. *J. Glob. Optim.*, 2012.

[24] Jack Kiefer, Jacob Wolfowitz, et al. Stochastic estimation of the maximum of a regression function. *Ann. Math. Stat.*, 23(3):462–466, 1952.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, 2014. `https://arxiv.org/abs/1412.6980`.

[26] Rodolphe Le Riche. Optimization under uncertainties: an overview with a focus on Gaussian processes. lecture at the CNRS French-German University school on Modeling and Numerical Methods for Uncertainty Quantification, cf. `https://hal.archives-ouvertes.fr/cel-02285533`, 2019.

[27] Nicholas Logothetis and Henry Wynn. *Quality through design: Experimental design, off-line quality control, and Taguchi's contributions.* Oxford University Press, 1989.

[28] Ilya Loshchilov, Marc Schoenauer, and Michèle Sebag. Self-adaptive surrogate-assisted covariance matrix adaptation evolution strategy. In *Proceedings of the 14th GECCO*, pages 321–328. ACM Press, 2012.

[29] Julien Marzat, Eric Walter, and Hélène Piet-Lahanier. A new expected-improvement algorithm for continuous minimax optimization. *J. Glob. Optim.*, 64:785–802, 2016.

[30] Maliki Moustapha and Bruno Sudret. Quantile-based optimization under uncertainties using bootstrap polynomial chaos expansions. In *Proceedings of the 12th International Conference on Structural Safety and Reliability (ICOSSAR 2017)*, pages 1561–1569. TU Verlag, 2017.

[31] Maliki Moustapha, Bruno Sudret, Jean-Marc Bourinet, and Benoît Guillaume. Quantile-based optimization under uncertainties using adaptive Kriging surrogate models. *Struct. Multidiscip. Optim.*, 54(6):1403–1421, 2016.

[32] Arkadi Nemirovski. On safe tractable approximations of chance constraints. *Eur. J. Oper. Res.*, 219(3):707–718, 2012.

[33] Gilles Pagès. Introduction to vector quantization and its applications for numerics. *ESAIM, Proc. Surv.*, 48:29–79, 2015.

[34] Gyung-Jin Park, Kwang-Hyeon Hwang, Tae Lee, and Kwon Hee Lee. Robust Design: An Overview. *AIAA J.*, 44:181–191, 2006.

[35] Julien Pelamatti, Loïc Brevault, Mathieu Balesdent, El-Ghazali Talbi, and Yannick Guerin. Efficient global optimization of constrained mixed variable problems. *J. Glob. Optim.*, 73(3):583–613, 2019.

[36] Valerio Perrone, Iaroslav Shcherbatyi, Rodolphe Jenatton, Cedric Archambeau, and Matthias Seeger. Constrained bayesian optimization with max-value entropy search. `https://arxiv.org/abs/1910.07003`, 2019.

[37] Victor Picheny. A Stepwise uncertainty reduction approach to constrained global optimization. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 787–795. PMLR, 2014.

[38] Michael J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer, 1994.

[39] Michael J. D. Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. In *Cambridge NA Report NA2009/06*, pages 26–46. University of Cambridge, 2009.

[40] Gilles Pujol, Rodolphe Le Riche, Olivier Roustant, and Xavier Bay. L'incertitude en conception: formalisation, estimation. In Rajan Filomeno Coelho and Piotr Breitkopf, editors, *Optimisation multidisciplinaire en mécanique: Réduction de modèles, robustesse, fiabilité, réalisations logicielles*. Hermes Science Publications, 2009.

[41] Jixiang Qing, Tom Dhaene, and Ivo Couckuyt. Spectral representation of robustness measures for optimization under input uncertainty. In *ICML2022, the 39th International Conference on Machine Learning*, pages 1–26, 2022.

[42] Mélina Ribaud, Christophette Blanchet-Scalliet, Céline Helbert, and Frédéric Gillot. Robust optimization: A kriging-based multi-objective optimization approach. *Reliability Engineering & System Safety*, 200: article no. 106913, 2020.

[43] Olivier Roustant, Esperan Padonou, Yves Deville, Aloïs Clément, Guillaume Perrin, Jean Giorla, and Henri Wynn. Group kernels for Gaussian process metamodels with categorical inputs. *SIAM/ASA J. Uncertain. Quantif.*, 2019.

[44] Michael Sasena, Panos Papalambros, and Pierre Goovaerts. Exploration of metamodeling sampling criteria for constrained global optimization. *Eng. Optim.*, 34(3):263–278, 2002.

[45] Michael Sasena, Panos Papalambros, and Pierre Goovaerts. Global optimization of problems with disconnected feasible regions via surrogate modeling. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002.

[46] Matthias Schonlau, William J. Welch, and Donald R. Jones. Global versus local search in constrained optimization of computer models. In *New developments and applications in experimental design*, volume 34 of *IMS Lecture Notes Monogr. Ser.*, pages 11–25. Institute of Mathematical Statistics, 1998.

[47] Gerhart I. Schuëller and Hector A. Jensen. Computational methods in optimization considering uncertainties–an overview. *Comput. Methods Appl. Mech. Eng.*, 198(1):2–13, 2008.

[48] James C. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons, 2005.

[49] Léonard Torossian, Aurélien Garivier, and Victor Picheny. X-Armed Bandits: Optimizing Quantiles, CVaR and Other Risks. In *Asian Conference on Machine Learning*, pages 252–267. PMLR, 2019.

[50] Léonard Torossian, Victor Picheny, Robert Faivre, and Aurélien Garivier. A review on quantile regression for stochastic computer experiments. *Reliability Engineering & System Safety*, : article no. 106858, 2020.

[51] Marcos A. Valdebenito and Gerhart I. Schuëller. A survey on approaches for reliability-based optimization. *Struct. Multidiscip. Optim.*, 42(5):645–663, 2010.

[52] Yunxiang Zhang, Xiangyu Zhang, and Peter Frazier. Constrained Two-step Look-Ahead Bayesian Optimization. *Adv. Neural Inf. Process. Syst.*, 34:12563–12575, 2021.

[53] Miguel Munoz Zuniga, Josselin Garnier, Emmanuel Remy, and Etienne de Rocquigny. Analysis of adaptive directional stratification for the controlled estimation of rare event probabilities. *Stat. Comput.*, 22(3):809–821, 2012.