



# A sampling criterion for constrained Bayesian optimization with uncertainties

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

## ► To cite this version:

Reda El Amri, Rodolphe Le Riche, Céline Helbert, Christophette Blanchet-Scalliet, Sébastien da Veiga. A sampling criterion for constrained Bayesian optimization with uncertainties. SMAI Journal of Computational Mathematics, 2023, 9, pp.285-309. 10.5802/smai-jcm.102 . emse-03167452v1

**HAL Id: emse-03167452**

**<https://hal-emse.ccsd.cnrs.fr/emse-03167452v1>**

Submitted on 25 Nov 2021 (v1), last revised 12 Dec 2023 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Sampling Criterion for Constrained Bayesian Optimization with Uncertainties

Reda El Amri<sup>\*</sup>, Rodolphe Le Riche<sup>†</sup>, Céline Helbert<sup>‡</sup>,  
Christophette Blanchet-Scalliet<sup>‡</sup>, Sébastien Da Veiga<sup>§</sup>

March 24, 2021

## 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 controlled-uncontrolled 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 controlled and uncontrolled 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. As a side result, an expression for the variance of the improvement is given.

## 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

---

<sup>\*</sup>Ecole Centrale de Lyon, now with CEA DER/SESI/LEMS, France

<sup>†</sup>CNRS LIMOS (Mines St-Etienne and UCA), France

<sup>‡</sup>Université de Lyon - CNRS, UMR 5208, Institut Camille Jordan - Ecole Centrale de Lyon, France

<sup>§</sup>Safran Tech, Modelling & Simulation, Magny-Les-Hameaux, France

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 [40]. 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. BO has rapidly been extended to encompass constraints [38, 32].

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

Uncertainties may originate from random environments such as the 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 any optimization problem.

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, 18]. 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 the first class of problems, the uncertainties are instantiated within the objective or constraint functions and cannot be chosen. Such uncertainties are an endured 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, accommodate the noise in the observations and still converge to problem solutions: this has given rise in the early 50's to stochastic descent algorithms [21, 3] that have since then experienced great developments [42, 2] often in relation to machine learning [22]; well-performing versions of (stochastic) evolutionary algorithms for noisy functions have been identified [11, 25] thanks to competitions on reference benchmarks [4].

In the second class of problems, the uncertainties perturb parameters that are distinct from the design variables and can be chosen during the simulations. The separation between controlled variables and uncertain parameters was already underlying Taguchi’s design of experiments in the 80’s [24]. Because of this separation and providing a probability of occurrence of the uncertainties exists, a statistical modelling in the joint *controlled*  $\times$  *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 [20] or one of its (super-)quantile [43, 44]. In Robust Optimization, the uncertainties are handled in terms of specific compromises between the average performance and its dispersion [30, 9] or by investigating all such Pareto optimal compromises through a multi-objective formulation [36].

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 [28] 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 [43, 23, 35]. 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. Many paths have been proposed to circumvent the double loop issue, some approximating the probability of feasibility by reliability indices [45], others improving the efficiency of the probability calculations (e.g., stratified sampling in [46]), others decoupling the reliability estimation from the optimization. [41] 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, 27]. In [20] and [1], the GP not only helps for the optimization (or for the inversion) of the controlled 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 [20], an optimization

and sampling Bayesian procedure is proposed that accounts for probabilistic 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 summed up in Section 5, and its performance assessed on an analytical and an industrial test case. An expression for the variance of the improvement, whose expectation is at the heart of the popular EGO algorithm [40], can be found in Appendix.

## 2 Problem formulation

### 2.1 Starting problem

Let  $f(\mathbf{x}, \mathbf{u})$  be the scalar output of an expensive computer simulation and let  $g_i(\mathbf{x}, \mathbf{u})$ ,  $i = 1, \dots, l$  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 controlled 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 [31, 37]).

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  $g_i(\mathbf{x}, \mathbf{U})$ ,  $i = 1, \dots, l$  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  $g_i(\mathbf{x}, \mathbf{U})$ ,  $i = 1, \dots, l$  [43, 23].

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  $g_i(\mathbf{x}, \mathbf{U}) \leq 0$ ,  $i = 1, \dots, l$  are satisfied with a high probability:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} \mathbb{E}_{\mathbf{U}}[f(\mathbf{x}, \mathbf{U})] \text{ s.t. } \mathbb{P}(g_i(\mathbf{x}, \mathbf{U}) \leq 0, i = 1, \dots, l) \geq 1 - \alpha \quad (1)$$

where  $\mathbf{U} \sim \rho_{\mathbf{U}}$  with support  $\mathcal{S}_{\mathcal{U}}$ .

$\alpha$  is a reliability parameter representing the allowed constraint violation level

( $0 < \alpha < 1$ ).

In the formulation of Equation (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 (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 [20]. Formulations such as Equation (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 [23, 35] but they remain an open challenge for costly problems. In the current work, Equation (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 (1) becomes

$$\begin{aligned} \mathbb{P}(g_i(\mathbf{x}, \mathbf{U}) \leq 0, i = 1, \dots, l) &\geq 1 - \alpha \\ \Leftrightarrow 1 - \alpha - \mathbb{E}_{\mathbf{U}}[\mathbb{1}_{\{g_i(\mathbf{x}, \mathbf{U}) \leq 0, i=1, \dots, l\}}] &\leq 0. \end{aligned}$$

From the last expression, the problem (1) is equivalent to

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} z(\mathbf{x}) \text{ s.t. } c(\mathbf{x}) \leq 0 \quad (2)$$

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

## 2.2 Gaussian Process regression framework

In the context of expensive computer simulation, the Problem (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

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

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  $D^{(t)} = \{(\mathbf{x}_k, \mathbf{u}_k) \mid k = 1, \dots, t\}$ . Since the expectation is a linear operator applied to  $f$ , it follows that  $Z^{(t)}(\mathbf{x}) = \mathbb{E}_{\mathbf{U}}[F^{(t)}(\mathbf{x}, \mathbf{U})]$  is still a Gaussian process with known mean  $m_Z^{(t)}$  and covariance function  $k_Z^{(t)}$  given by:

$$\begin{aligned} m_Z^{(t)}(\mathbf{x}) &= \int_{\mathbb{R}^m} m_F^{(t)}(\mathbf{x}, \mathbf{u}) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u}, \\ k_Z^{(t)}(\mathbf{x}, \mathbf{x}') &= \iint_{\mathbb{R}^m} k_F^{(t)}(\mathbf{x}, \mathbf{u}, \mathbf{x}', \mathbf{u}') \rho_{\mathbf{U}}(\mathbf{u}, \mathbf{u}') d\mathbf{u} d\mathbf{u}'. \end{aligned} \quad (3)$$

The integrals that appears in (3) can be evaluated analytically for specific choices of  $\rho_{\mathbf{U}}$  and  $k_F$  (see [20]). In the general case, a quadrature rule can be used to approximate these integrals.

We also introduce the process

$$C^{(t)}(\mathbf{x}) = 1 - \alpha - \mathbb{E}_{\mathbf{U}}[\mathbb{1}_{\cap_{i=1}^t \{G_i^{(t)}(\mathbf{x}, \mathbf{U}) \leq 0\}}]$$

which is the statistical model of the constraint  $c$  (Equation (2)). Note that the process  $C^{(t)}$  is not Gaussian. In the Bayesian GP framework, Problem (2) can be approximated as find

$$\arg \min_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} Z^{(t)}(\mathbf{x}) \text{ s.t. } C^{(t)}(\mathbf{x}) \leq 0. \quad (4)$$

To solve (4), a Bayesian optimization procedure is introduced 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 (1) within a restricted number of calls to  $f$  and the  $g_i$ 's. But 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 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:  $F^{(n_0)}, G_i^{(n_0)}, i = 1, \dots, l$  with  $D^{(n_0)}$   
**while** stopping criterion not met **do**  
     Determine a desirable, targeted,  $\mathbf{x}$  by maximizing the expectation of the progress measure,

$$\mathbf{x}_{targ} = \arg \max_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} \mathbb{E} \left( P^{(t)}(\mathbf{x}) \right) . \quad (5)$$

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

$$(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) = \arg \min_{\tilde{\mathbf{x}}, \tilde{\mathbf{u}} \in \mathcal{S}_{\mathcal{X}} \times \mathcal{S}_{\mathcal{U}}} \text{VAR} \left( P^{(t+1)}(\mathbf{x}_{targ}) \right) \quad (6)$$

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  $g_i, i = 1, \dots, l$ , 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,  $F^{(t+1)}, G_i^{(t+1)}, i = 1, \dots, l$ .

**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  $P(\mathbf{x}) = -F(\mathbf{x}) - \sum_{i=1}^l p_i \max(0, G_i(\mathbf{x}, \mathbf{u}^{\text{mod}}))$  where the  $p_i$ 's are positive penalty parameters and  $\mathbf{u}^{\text{mod}}$  the mode of  $\mathbf{U}$ , or  $P(\mathbf{x}) = \max([z_{\min}^{\text{feas}} - Z(\mathbf{x}) \mid C(\mathbf{x}) \leq 0], 0)$  where  $z_{\min}^{\text{feas}}$  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 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 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,  $\mathbf{x}_{targ}$ , 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 [40, 39] defined by

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

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

In our case,  $z_{\min}^{\text{feas}}$  must be further explained because it is not directly observed. This will be the subject of the next section. The definition of  $z_{\min}^{\text{feas}}$  and the fact that  $C^{(t)}(\mathbf{x})$  is not Gaussian is a difference between the *FI* of this article and those in [40, 39].

Following Proposition 1 and Equation (5) 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),

$$\mathbf{x}_{\text{targ}} = \arg \max_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} \text{EFI}^{(t)}(\mathbf{x}). \quad (7)$$

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

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

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

$$\text{EI}^{(t)}(\mathbf{x}) = (z_{\min}^{\text{feas}} - m_Z^{(t)}(\mathbf{x})) \Phi\left(\frac{z_{\min}^{\text{feas}} - m_Z^{(t)}(\mathbf{x})}{\sigma_Z^{(t)}(\mathbf{x})}\right) + \sigma_Z^{(t)}(\mathbf{x}) \phi\left(\frac{z_{\min}^{\text{feas}} - m_Z^{(t)}(\mathbf{x})}{\sigma_Z^{(t)}(\mathbf{x})}\right), \quad (8)$$

where  $\sigma_Z^{(t)}(\mathbf{x}) = \sqrt{k_Z^{(t)}(\mathbf{x}, \mathbf{x})}$ ,  $\Phi$  and  $\phi$  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 $z_{\min}^{\text{feas}}$

To solve problem (7), we need to define  $z_{\min}^{\text{feas}}$ . We extend the definition of the current minimum for a non-observed process  $Z$  introduced in [20] to a problem with constraints.  $z_{\min}^{\text{feas}}$  is defined as the minimum of the mean of the process  $Z^{(t)}$  such that the constraint is satisfied in expectation

$$z_{\min}^{\text{feas}} = \arg \min_{\mathbf{x} \in \mathcal{X}_t} m_Z^{(t)}(\mathbf{x}) \text{ s.t. } \mathbb{E}[C^{(t)}(\mathbf{x})] \leq 0. \quad (9)$$

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

$$\begin{aligned}
\mathbb{E}[C^{(t)}(\mathbf{x})] &= \mathbb{E}[1 - \alpha - \mathbb{E}_{\mathbf{U}}[\mathbb{1}_{\cap_{i=1}^l \{G_i^{(t)}(\mathbf{x}, \mathbf{U}) \leq 0\}}]] \\
&= 1 - \alpha - \mathbb{E}_{\mathbf{U}}\left[\prod_{i=1}^l \mathbb{E}[\mathbb{1}_{\{G_i^{(t)}(\mathbf{x}, \mathbf{U}) \leq 0\}}]\right] \\
&= 1 - \alpha - \int_{\mathbb{R}^m} \prod_{i=1}^l \Phi\left(\frac{-m_{G_i}^{(t)}(\mathbf{x}, \mathbf{u})}{\sigma_{G_i}^{(t)}(\mathbf{x}, \mathbf{u})}\right) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u}.
\end{aligned} \tag{10}$$

If there is no solution to problem (9), 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 1, Equation (6), 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.,

$$(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) = \arg \min_{(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) \in \mathcal{S}_{\mathcal{X}} \times \mathcal{S}_{\mathcal{U}}} \text{VAR} \left( I^{(t+1)}(\mathbf{x}_{\text{targ}}) \mathbb{1}_{\{C^{(t+1)}(\mathbf{x}_{\text{targ}}) \leq 0\}} \right), \tag{11}$$

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 [20], the authors have 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 (11)) and it yields the best candidate point  $\mathbf{u}_{t+1}$ . As the one-step-ahead variance of the feasible improvement is difficult to evaluate, a proxy built with a product of variances is now proposed.

### The sampling criterion

We introduce a new sampling criterion specific to the robust optimization Problem (1), denoted  $S$  for Sampling as it is used to generate a new value for  $\mathbf{u}$ . It is defined as a proxy of the variance of the one-step-ahead feasible improvement.

**Proposition 2.** *A proxy of the variance of the one-step-ahead feasible improvement (Equation (11)) given a new observation at point  $(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$  is*

$$\begin{aligned} S(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) &= \text{VAR}(I^{t+1}(\mathbf{x}_{\text{targ}})) \int_{\mathbb{R}^m} \text{VAR}(\mathbb{1}_{\cap_{i=1}^l \{G_i^{(t+1)}(\mathbf{x}_{\text{targ}}, \mathbf{u}) \leq 0\}}) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u}, \\ &= \text{VAR}((z_{\min}^{\text{feas}} - Z^{(t+1)}(\mathbf{x}_{\text{targ}}))^+) \int_{\mathbb{R}^m} \text{VAR}(\mathbb{1}_{\cap_{i=1}^l \{G_i^{(t+1)}(\mathbf{x}_{\text{targ}}, \mathbf{u}) \leq 0\}}) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u}. \end{aligned} \quad (12)$$

In Equation (12) the first part of the expression pushes 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. But 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 (11)), is the variance of the product of the improvement times a feasibility with confidence. The proxy  $S$  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<sup>1</sup>. Furthermore, the second term in the product for  $S$  (Equation (12)) 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

$$\mathbf{u}_{t+1} = \arg \min_{\tilde{\mathbf{u}} \in S_{\mathcal{U}}} S(\mathbf{x}_{\text{targ}}, \tilde{\mathbf{u}}). \quad (13)$$

The calculation of both terms of  $S$  is now presented.

---

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

### Calculation of $\text{VAR}(I^{t+1}(\mathbf{x}_{\text{targ}}))$

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

$$\begin{aligned}\text{VAR}(I^{t+1}(\mathbf{x}_{\text{targ}})) &= \text{VAR}\left((z_{\min}^{\text{feas}} - Z^{(t+1)}(\mathbf{x}_{\text{targ}}))^+\right) \\ &= \text{VAR}\left((z_{\min}^{\text{feas}} - Z(\mathbf{x}_{\text{targ}}))^+ | F(D^{(t)}) = f^{(t)}, F(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) = f(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})\right)\end{aligned}\quad (14)$$

The expression of the expected improvement in terms of PDF and CDF of Gaussian distribution is well-known. The following gives the formula for the variance (see A.1 for the proof).

**Proposition 3** (The Variance of the Improvement).

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

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 3 to Equation (14). We use that  $F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) \sim \mathcal{N}(m_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}), k_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}}))$  and the law of total variance

$$\begin{aligned}\text{VAR}(I^{t+1}(\mathbf{x}_{\text{targ}})) &= \mathbb{E}\left[\text{VAR}\left((z_{\min}^{\text{feas}} - Z(\mathbf{x}_{\text{targ}}))^+ | F(D^{(t)}) = f^{(t)}, F(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})\right)\right] \\ &\quad + \text{VAR}\left[\mathbb{E}\left((z_{\min}^{\text{feas}} - Z(\mathbf{x}_{\text{targ}}))^+ | F(D^{(t)}) = f^{(t)}, F(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})\right)\right].\end{aligned}\quad (15)$$

To compute Equation (15), 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 3 and Equation (8) for the first and second brackets, respectively. The external  $\mathbb{E}$  and  $\text{VAR}$  only concern  $m_Z^{(t+1)}(\mathbf{x}_{\text{targ}})$  whose randomness comes from  $F(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$ ,  $\sigma_Z^{(t+1)}(\mathbf{x}_{\text{targ}})$  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

$$m_Z^{(t+1)}(\mathbf{x}_{\text{targ}}) \sim \mathcal{N}\left(m_Z^{(t)}(\mathbf{x}_{\text{targ}}), \left(\frac{\int_{\mathbb{R}^m} k_F^{(t)}(\mathbf{x}_{\text{targ}}, \mathbf{u}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}}) \rho_{\mathbf{U}}(\mathbf{u}) d\mathbf{u}}{\sqrt{k_F^{(t)}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}; \tilde{\mathbf{x}}, \tilde{\mathbf{u}})}}\right)^2\right).\quad (16)$$

Finally, the external  $\mathbb{E}$  and  $\text{VAR}$  in Equation (15) are numerically evaluated. Details are given in Section 5.2.

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

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

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

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 Sampling and Optimization through Feasible Improvement.

---

**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:

$D^{(t)} = \{(\mathbf{x}_i, \mathbf{u}_i) \mid i = 1, \dots, t\}$ , 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** EFI to define  $\mathbf{x}_{\text{targ}} = \arg \max_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} \text{EFI}^{(t)}(\mathbf{x})$  (Eq. (7))

Set  $\mathbf{x}_{t+1} = \mathbf{x}_{\text{targ}}$

**Sample** the next uncertain point by solving

$\mathbf{u}_{t+1} = \arg \min_{\tilde{\mathbf{u}} \in \mathcal{S}_{\mathcal{U}}} S(\mathbf{x}_{\text{targ}}, \tilde{\mathbf{u}})$  (Eq. (13))

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**

---

## 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. First, the **EFIr**and algorithm is identical to EFISUR to the exception of the point  $\mathbf{u}_{t+1}$  which is simply sampled from its distribution. This **EFIr**and algorithm will be useful to judge the usefulness of the sampling criterion  $S$  in EFISUR.

---

**Algorithm 2 : EFi**rand Expected Feasible Improvement with random sampling

---

Create an initial DoE of size  $t$  in the joint space and calculate simulator responses:  
 $D^{(t)} = \{(\mathbf{x}_i, \mathbf{u}_i) \mid i = 1, \dots, t\}$ , 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** EFI to define  $\mathbf{x}_{t+1} = \arg \max_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} \text{EFI}^{(t)}(\mathbf{x})$  (Eq. (7))  
    **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 which EFISUR will be compared to uses the quantile as an alternative way of measuring the failure probability:

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

The quantile of the constraint  $i = 1, \dots, l$  is approximated using the predictive mean of the GP model  $G^{(t)}$ ,

$$q_{1-\alpha}(g_i(\mathbf{x}, \mathbf{U})) \approx q_{1-\alpha}(m_{G_i}^{(t)}(\mathbf{x}, \mathbf{U})). \quad (17)$$

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:

$$\begin{aligned} \mathbf{x}_{t+1} &= \arg \max_{\mathbf{x} \in \mathcal{S}_{\mathcal{X}}} \text{EI}^{(t)}(\mathbf{x}) \\ \text{s.t. } \forall i \in \{1, \dots, l\}, \quad q_{1-\alpha}(m_{G_i}^{(t)}(\mathbf{x}, \mathbf{U})) &\leq 0. \end{aligned} \quad (18)$$

The sampling in the uncertain space is based on the *deviation number* developed by [15, 17] 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,

$$\text{DN}_i(\mathbf{u}) = \frac{|m_{G_i^{(t)}}(\mathbf{x}_{t+1}, \mathbf{u})|}{\sigma_{G_i^{(t)}}(\mathbf{x}_{t+1}, \mathbf{u})}, \quad (19)$$

The points that have a low deviation number are either close to the constraint threshold (null in Equation (19)), or they have a high GP variance. To handle multiple constraints, the constraint with the minimum value of DN is selected (as in [26]),

$$\text{DN}_c(\mathbf{u}) = \min_{i=1, \dots, l} \text{DN}_i(\mathbf{u}). \quad (20)$$

The whole algorithm is called **cEIddevNum** for Constrained EI plus Deviation Number. The **cEIddevNum** algorithm is an alternative to **EFISUR** for handling chance constraints in the joined 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 (1). The sampling step (the DN minimization) accounts only for the constraints satisfaction and not for the objective function. The **cEIddevNum** algorithm bears some resemblance to the approach described in [27]: 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 (18), can be found in other articles, e.g., [6]. However, to the authors' knowledge, **cEIddevNum** integrates these techniques within an EGO-like algorithm in an original manner.

---

**Algorithm 3 : cEIdvNum**

---

Create an initial DoE of size  $t$  in the joint space and calculate simulator responses:

$D^{(t)} = \{(\mathbf{x}_i, \mathbf{u}_i) \mid i = 1, \dots, t\}$ , 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** the expected improvement under quantile constraints to determine the next iterate

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

$$\text{s.t. } \forall i \in \{1, \dots, l\}, \quad q_{1-\alpha}(m_{G_i}^{(t)}(\mathbf{x}, \mathbf{U})) \leq 0, \quad (\text{Eq. (18)})$$

**Sample** the next uncertainty by minimizing the deviation number,

$$\mathbf{u}_{t+1} = \arg \min_{\mathbf{u}} \text{DN}_c(\mathbf{u}) \quad (\text{Eq. (20)})$$

    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**

---

## 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  $\{\mathbf{u}_1, \dots, \mathbf{u}_M\}$ , obtained by a Sobol sequence, 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 Sobol sequence at different steps:

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

$$\mathbb{P}(C^{(t)}(\mathbf{x}) \leq 0) \approx \frac{1}{N} \sum_{k=1}^N \mathbb{1}_{\left(1-\alpha-\frac{1}{M} \sum_{j=1}^M \mathbb{1}_{\left(G_i^{(t)}(\mathbf{x}, \mathbf{u}_j, \omega_k) \leq 0, i=1, \dots, l\right)}\right) \leq 0} \quad (21)$$

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 **cEIddevNum** 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.

### 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 (15)). To this end, we use a quantization technique [29]. It consists in approximating the continuous distribution of  $m_Z^{(t+1)}$  (given in Equation (16)) by a discrete one. Thus, the external variance and expectation of Equation (15) 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  $\text{EFI}$  with respect to  $\mathbf{x}$  (**EFISUR** and **EFrand** algorithms) or the minimization of  $S$  or  $U_c$  with respect to  $\mathbf{u}$  (**EFISUR** and **cEIdvNum**). Such internal continuous unconstrained optimization problems are handled with the derivative-free solver **BOBYQA** [34].

The Problem (18) in the algorithm **cEIdvNum** further requires a solver that can handle constraints. It is addressed with the **COBYLA** program [33].

### 5.3 Analytical test case

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

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

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 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  $40 \times 40$  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 20 times for each method. The initial Design of Experiments of each method is a random Latin hypercube of  $4 + (d + m) = 8$  points. An additional budget of 56 iterations<sup>2</sup> 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 average distance to the solution is plotted in Figure 2. **EFISUR** and **EFrand** converge faster to the optimum than **cEIdvNum**. After 18 iterations, **EFISUR** approaches the solution more closely than **EFrand** does. A complementary view of the convergence, with dispersions, can be found in Figure 3 which shows the boxplots of the distances to the reference solution at iterations 10 (left panel), 20 (middle) and 30 (right). It

---

<sup>2</sup>An iteration encompasses one call to the objective function and one call to all the constraints.

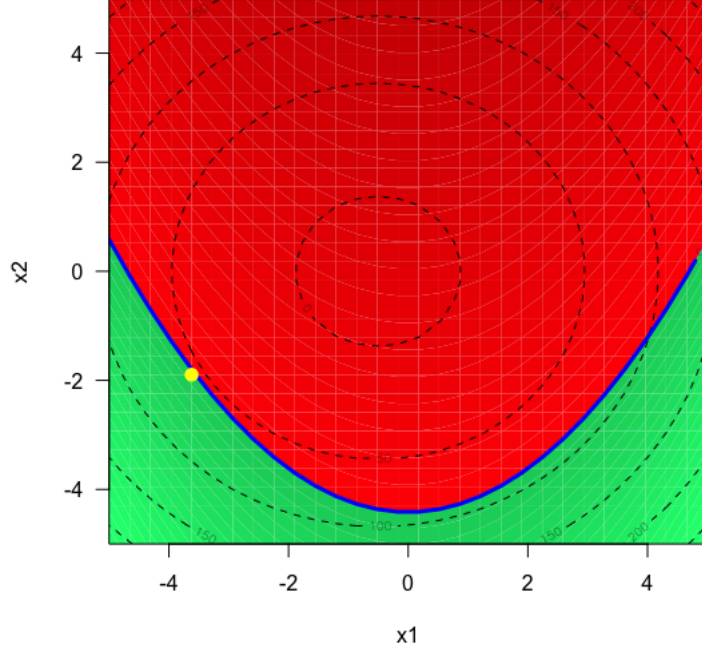


Figure 1: Contour plots of  $\mathbb{P}(g(\cdot, \mathbf{U}) \leq 0)$  and  $\mathbb{E}[f(\cdot, \mathbf{U})]$ : failure and feasible regions in red and green, respectively, the limit-state function in blue, objective function in dashed black lines. The solution is the yellow bullet.

is observed that **EFISUR** leads to an accurate solution from 20 iterations onwards with a small deviation between the runs. **EFirand** has a better start (at 10 iterations) but it is then surpassed by **EFISUR**. At all iterations, **cEIddevNum** has a larger median distance to the solution and a larger spread in results. Figure 4 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. **EFISUR** (left) and **cEIddevNum** (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, **cEIddevNum** samples 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 DN small). **EFISUR** breaks 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 “ $-x_1 u_1$ ”.

Two partial conclusions can be presumed from this set of experiments. First, the algorithms that rely on the EFI aggregated criterion to choose the next set of controlled variables, i.e., **EFirand** and **EFISUR**, converge better than **cEIddevNum** and its constrained problem handled through **COBYLA**. Second, **EFISUR** provides an additional gain in efficiency thanks to its sampling

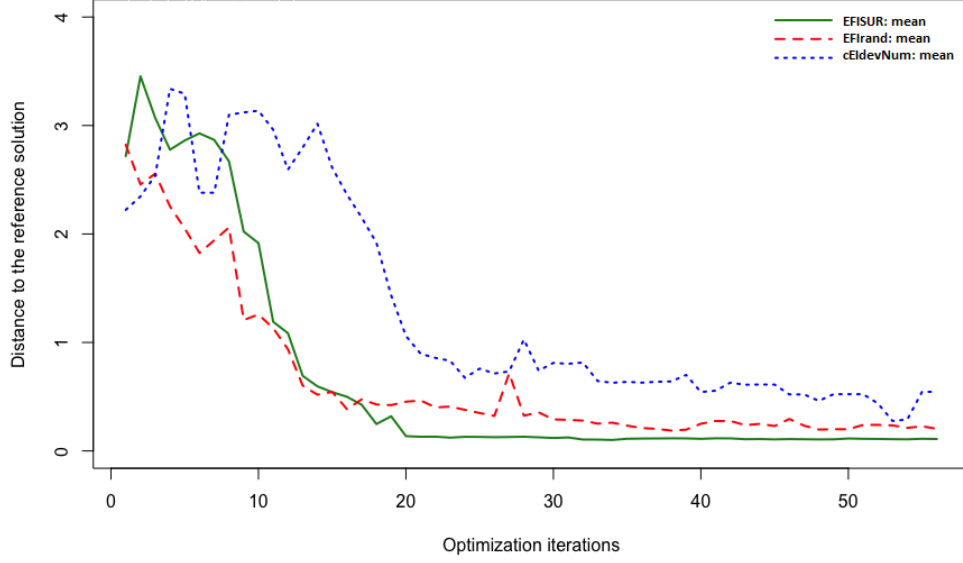


Figure 2: Mean convergence rates (Euclidean distance to the solution) on the analytical test case. The **EFISUR** method is plotted with the green solid line; **EFrand** with the red dashed line; **cEIddevNum** with the blue dotted line. The initial DoE preceded these iterations, it is not represented.

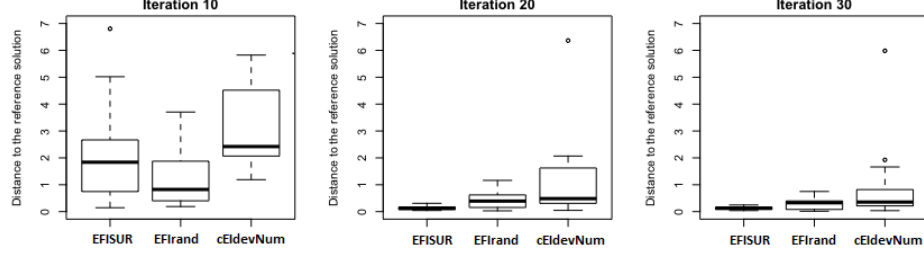


Figure 3: Distance to the reference solution at iteration 10 (left panel), 20 (middle panel) and 30 (right panel) for the three strategies. The boxplots summarize 20 replications of the runs.

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 [19]). The optimization of the NASA rotor 37 compressor blade is a

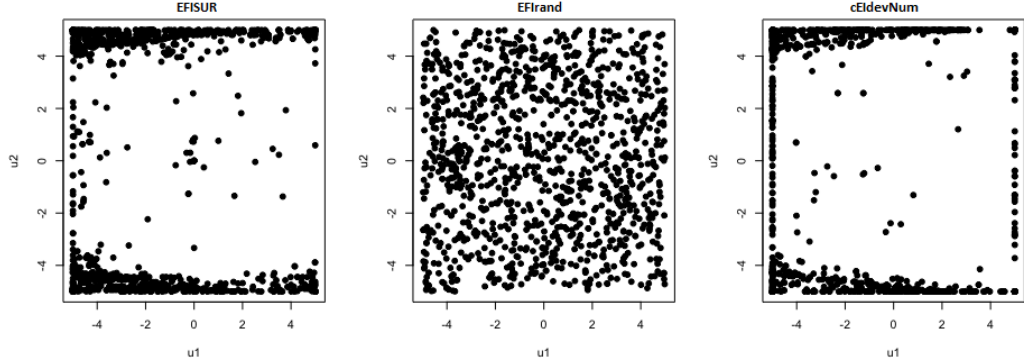


Figure 4: Enrichment in the uncertain space,  $\mathcal{S}_U$ , for the three methods.

challenging test case first of all 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:

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

Because of the dimensionality and the numerical cost, the use of surrogate models is the only path to perform such an optimization. The EFISUR method is started with a DoE of 100 points drawn from an optimal Latin Hypercube Sampling. The enrichment is then carried out by adding 137 points, for a total of 237 model evaluations. Figure 5 shows the convergence of the feasible minimum. Figure 6 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.

The result given above 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 (21)) is repeated 500 times with  $M = 1000$   $\mathbf{u}$  samples and  $N = 1000$  trajectories of the GPs. Figure 7 provides the statistics of the probability of constraints

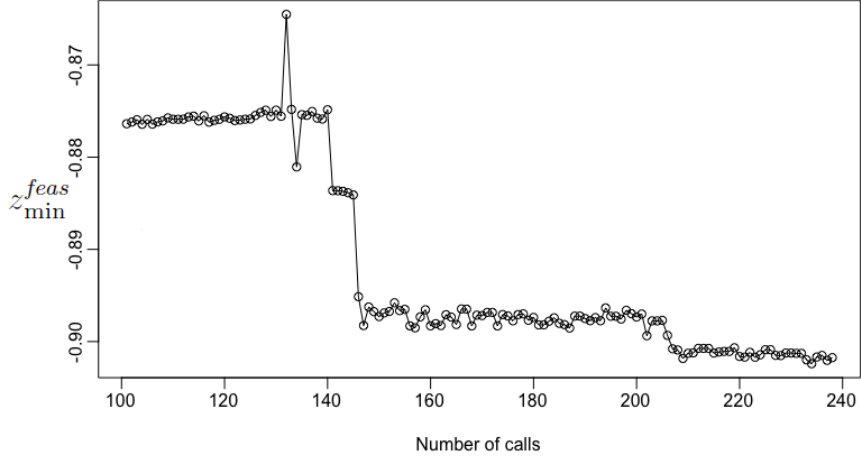


Figure 5: Convergence history of the current feasible minimum,  $z_{\min}^{\text{feas}}$ .

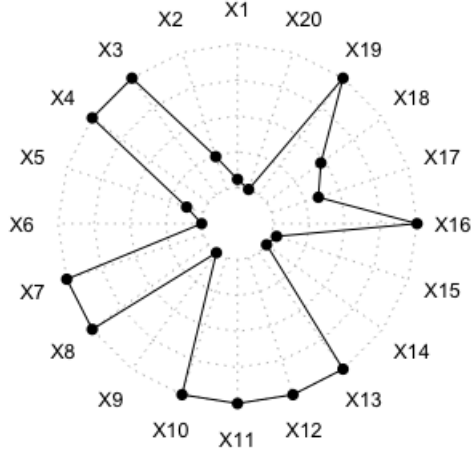


Figure 6: Relative coordinates of the optimal design with respect to their respective lower and upper bounds.

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.

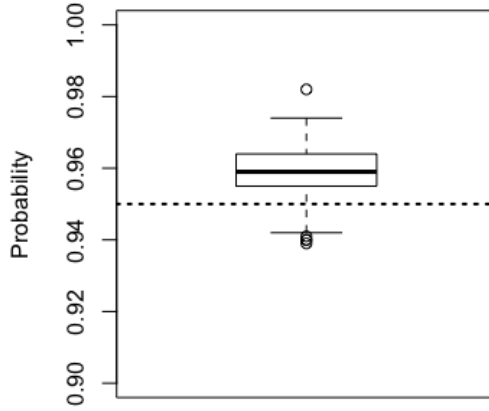


Figure 7: Distribution of constraints satisfaction probabilities at the optimal design computed with the final GPs. The boxplot summarizes 500 replications. The dashed line is the lower bound on constraint satisfaction (0.95).

## 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 Gaussian processes built in the augmented space of the controlled 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 a computationally tractable approximation to the one-step-ahead variance reduction in feasible improvement.

The article has detailed the analytical expressions of the acquisition and sampling criteria. Along the way, an expression for the variance of the improvement has been given.

**EFISUR** has been compared to 2 alternative algorithms which differ in the acquisition and sampling criteria. The results show a gain in favor of the expected feasible improvement and its one-step-ahead variance reduction. 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.

## Acknowledgement

This work was supported in part by the research chair OQUAIDO in applied mathematics.

## References

- [1] Reda El Amri, Céline Helbert, Miguel Munoz Zuniga, Clémentine Prieur, and Delphine Sinoquet. Set inversion under functional uncertainties with joint meta-models. working paper or preprint, 2020.
- [2] Laetitia Andrieu, Guy Cohen, and Felisa J Vázquez-Abad. Gradient-based simulation optimization under probability constraints. *European Journal of Operational Research*, 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. *Aerospace Science and technology*, 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. *Statistics and Computing*, 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. *Computer methods in applied mechanics and engineering*, 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*, 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. *Structural and Multidisciplinary Optimization*, 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] 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.
- [18] Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. *European journal of operational research*, 235(3):471–483, 2014.
- [19] Charles Hirsch, Dirk Wunsch, Jacek Szumbariski, J Pons-Prats, et al. Uncertainty management for robust industrial design in aeronautics. 2019.

- [20] Janis Janusevskis and Rodolphe Le Riche. Simultaneous kriging-based estimation and optimization of mean response. *Journal of Global Optimization*, 2012. DOI 10.1007/s10898-011-9836-5.
- [21] Jack Kiefer, Jacob Wolfowitz, et al. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.
- [22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.
- [23] 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/ce1-02285533>, September 2019.
- [24] Nicholas Logothetis and Henry P Wynn. *Quality through design: Experimental design, off-line quality control, and Taguchi’s contributions*. Oxford University Press, USA, 1989.
- [25] Ilya Loshchilov, Marc Schoenauer, and Michèle Sebag. Self-adaptive surrogate-assisted covariance matrix adaptation evolution strategy. In *Proc. of the 14th GECCO*, pages 321–328. ACM, 2012.
- [26] 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.
- [27] Maliki Moustapha, Bruno Sudret, Jean-Marc Bourinet, and Benoît Guillaume. Quantile-based optimization under uncertainties using adaptive kriging surrogate models. *Structural and multidisciplinary optimization*, 54(6):1403–1421, 2016.
- [28] Arkadi Nemirovski. On safe tractable approximations of chance constraints. *European Journal of Operational Research*, 219(3):707–718, 2012.
- [29] Gilles Pagès. Introduction to vector quantization and its applications for numerics. *ESAIM: proceedings and surveys*, 48:29–79, 2015.
- [30] Gyung-Jin Park, Kwang-Hyeon Hwang, Tae Lee, and Kwon Hee Lee. Robust design: An overview. *AIAA Journal*, 44:181–191, 01 2006.
- [31] Julien Pelamatti, Loïc Brevault, Mathieu Balesdent, El-Ghazali Talbi, and Yannick Guerin. Efficient global optimization of constrained mixed variable problems. *Journal of Global Optimization*, 73(3):583–613, 2019.

- [32] Victor Picheny. A Stepwise uncertainty reduction approach to constrained global optimization. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 787–795, Reykjavik, Iceland, 22–25 Apr 2014. PMLR.
- [33] Michael JD 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.
- [34] Michael JD Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06*, University of Cambridge, Cambridge, pages 26–46, 2009.
- [35] Gilles Pujol, Rodolphe Le Riche, Olivier Roustant, and Xavier Bay. *Optimisation multidisciplinaire en mécanique: Réduction de modèles, robustesse, fiabilité, réalisations logicielles*, chapter L’incertitude en conception: formalisation, estimation. Hermes Science Publications, Paris, 2009. (in French).
- [36] Mélina Ribaud, Christopette Blanchet-Scalliet, Céline Helbert, and Frédéric Gillot. Robust optimization: A kriging-based multi-objective optimization approach. *Reliability Engineering and System Safety*, 200:106913, 2020.
- [37] 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 Journal on Uncertainty Quantification*, 2019.
- [38] 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*, page 5573, 2002.
- [39] Michael J Sasena, Panos Papalambros, and Pierre Goovaerts. Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering optimization*, 34(3):263–278, 2002.
- [40] Matthias Schonlau, William J Welch, and Donald R Jones. Global versus local search in constrained optimization of computer models. *Lecture Notes-Monograph Series*, pages 11–25, 1998.
- [41] Gerhart I Schuëller and Hector A Jensen. Computational methods in optimization considering uncertainties—an overview. *Computer Methods in Applied Mechanics and Engineering*, 198(1):2–13, 2008.

- [42] James C Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons, 2005.
- [43] 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.
- [44] Léonard Torossian, Victor Picheny, Robert Faivre, and Aurélien Garivier. A review on quantile regression for stochastic computer experiments. *Reliability Engineering & System Safety*, page 106858, 2020.
- [45] Marcos A Valdebenito and Gerhart I Schuëller. A survey on approaches for reliability-based optimization. *Structural and Multidisciplinary Optimization*, 42(5):645–663, 2010.
- [46] Miguel Munoz Zuniga, Josselin Garnier, Emmanuel Remy, and Etienne de Rocquigny. Analysis of adaptive directional stratification for the controlled estimation of rare event probabilities. *Statistics and Computing*, 22(3):809–821, 2012.

## A Expression of the aggregated variance sampling criterion

### A.1 The *Variance of the Improvement*

We now prove the closed-form expression of the *Variance of the Improvement* stated in Proposition 3:

$$\text{VAR}(I(\mathbf{x})) = \text{EI}(\mathbf{x})(z_{\min} - m_Z(\mathbf{x}) - \text{EI}(\mathbf{x})) + \sigma_Z^2(\mathbf{x})\phi\left(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\right).$$

*Proof.*

$$\begin{aligned}
VI(\mathbf{x}) &= \text{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 - (\text{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}) \cdot v)^2 \phi(v) dv - (\text{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 - (\text{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\left(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\right) \\
&\quad + 2\sigma_Z(\mathbf{x})(z_{\min} - m_Z(\mathbf{x})) \phi\left(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\right) - (\text{EI}(\mathbf{x}))^2, \\
&= \sigma_Z^2(\mathbf{x}) \left( [-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 \right) \\
&\quad + (z_{\min} - m_Z(\mathbf{x}))^2 \Phi\left(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\right) \\
&\quad + 2\sigma_Z(\mathbf{x})(z_{\min} - m_Z(\mathbf{x})) \phi\left(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\right) - (\text{EI}(\mathbf{x}))^2, \\
&= -\sigma_Z(\mathbf{x})(z_{\min} - m_Z(\mathbf{x})) \phi\left(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\right) \\
&\quad + \sigma_Z^2(\mathbf{x}) \Phi\left(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\right) + (z_{\min} - m_Z(\mathbf{x}))^2 \Phi\left(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\right) \\
&\quad + 2\sigma_Z(\mathbf{x})(z_{\min} - m_Z(\mathbf{x})) \phi\left(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\right) - (\text{EI}(\mathbf{x}))^2, \\
&= \left( (z_{\min} - m_Z(\mathbf{x}))^2 + \sigma_Z^2(\mathbf{x}) \right) \Phi\left(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\right) \\
&\quad + \sigma_Z(\mathbf{x})(z_{\min} - m_Z(\mathbf{x})) \phi\left(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\right) - (\text{EI}(\mathbf{x}))^2, \\
&= \text{EI}(\mathbf{x})(z_{\min} - m_Z(\mathbf{x}) - \text{EI}(\mathbf{x})) + \sigma_Z^2(\mathbf{x}) \Phi\left(\frac{z_{\min} - m_Z(\mathbf{x})}{\sigma_Z(\mathbf{x})}\right).
\end{aligned}$$

□

Use was made of the expression of the expected improvement, EI, Equation (8). 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 (15) that results from the law of total variance,

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

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  $F(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$ .

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]:

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

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

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