Gaussian process optimization with simulation failures

F Bachoc, Céline Helbert, V. Picheny

Gaussian process optimization with simulation failures

F. Bachoc[1], C. Helbert[2], V. Picheny[3]
[1] Institut de Mathématiques de Toulouse, Université Paul Sabatier
[2] Univ. de Lyon, Ecole Centrale de Lyon, CNRS UMR 5208,
Institut Camille Jordan, 36 av. G. de Collongue F-69134 Ecully cedex, FRANCE
[3] PROWLER.io, Cambridge, UK

April 16, 2019

Abstract

We address the optimization of a computer model, where each simulation either fails or returns a valid output performance. We suggest a joint Gaussian process model for classification of the inputs (computation failure or success) and for regression of the performance function. We discuss the maximum likelihood estimation of the covariance parameters, with a stochastic approximation of the gradient. We then extend the celebrated expected improvement criterion to our setting of joint classification and regression, thus obtaining a global optimization algorithm. We prove the convergence of this algorithm. We also study its practical performances, on simulated data, and on a real computer model in the context of automotive fan design.

1 Introduction

Bayesian optimization (BO) is now well-established as an efficient tool to solve optimization problems with non-linear expensive-to-evaluate objectives. A wide range of applications have been tackled, from the hyperparameter tuning of machine learning algorithms [30] to wing shape design [15]. In the simplest BO setting, the aim is to find the maximum of a fixed unknown function $f : \mathcal{D} \to \mathbb{R}$, where $\mathcal{D}$ is a box of dimension d. Under that configuration, the celebrated *Efficient Global Optimization* (EGO) and its underlying acquisition function *Expected Improvement* (EI) are still considered as state-of-the-art.

Several authors have adapted BO to the constrained optimization framework, i.e. when the acceptable design space $\mathcal{A} \subset \mathcal{D}$ is defined by a set of non-linear, expensive-to-compute equations c:

$$\mathcal{A} = \{x \in \mathcal{D} \text{ s.t. } c(x) \leq 0\},$$

either by considering the EI function [29, 28, 5, 10, 24] or by proposing alternative acquisition functions [23, 11].

We consider here the problem of *crash constraints*, where the objective f is typically evaluated using a computer code that fails to provide simulation results $f(x)$ for some input conditions x. We write $\mathcal{A}$ of the form

$$\mathcal{A} = \{x \in \mathcal{D}; s(x) = 1\}$$

where $s : \mathcal{D} \to \{0, 1\}$ is a fixed unknown function.

We assume that, for each $x \in \mathcal{D}$, a single computation provides the pair $(s(x), \mathbf{1}_{s(x)=1} f(x))$. Hence, it is as costly to see if a simulation at x fails as to observe the simulation result $f(x)$ when there is no failure. A first typical example of failure can be a computational fluid dynamics (CFD) solver that does not converge. This convergence failure can be caused by an overly large time step yielding an instability of the numerical scheme and a divergence, or also by an inadapted mesh close to the boundary of the domain (see also for instance the discussions in [27]). A second typical example of failure is when $f(x)$ provides the numerical performance (e.g. the empirical risk) of a complex machine learning model (e.g. a deep neural network) depending on architecture parameters in x [14]. The computation of $f(x)$ then relies on a gradient or stochastic gradient descent, using for instance retro-propagation in the case of deep learning. In this case, a failure occurs when the gradient descent does not converge, so that there is no observable value of $f(x)$ at convergence. In these two examples, we remark that, indeed, it is no less costly to observe a failure of the form $s(x_1) = 0$, than to successfully observe $f(x_2)$ with $s(x_2) = 1$.

This optimization problem with failures was considered first by [9], where a Gaussian process classifier [GPC, 21] was used together with a spatialized EI. [16] also proposed the use of a GPC with EI, modified using an asymmetric entropy to limit as much as possible the computational resources spent on crashed simulations. However, both approaches rely on expensive Monte-Carlo simulations, which make them impractical in some cases, and do not provide any convergence guarantee.

The contribution of this paper is two-fold. First, a new GPC model is proposed, where a latent GP is simply conditioned on the signs of the observations instead of their values. Its likelihood function maximization is studied, as well as its use to predict the feasibility probability (i.e. crash likeliness) of a new design x. Second, leveraging recent results on sequential strategies [2], we propose an algorithm in the form of EGO with guaranteed convergence.

The outline of this paper is as follows: First, we introduce our GPC model (Section 2) and its use in a Bayesian optimization algorithm (Section 3). Section 4 states our main consistency result. Finally, our algorithm is first illustrated on several simulated toy problems (Section 5), then applied to an industrial case study (Section 6). A conclusion is given in Section 7. All the proofs are postponed to the appendix.

2 A Classification model for crash constraints

This section presents our classification model used to characterize the feasible space $\mathcal{A}$. It takes the classical form of a GPC with a latent GP, but conditioned solely on pointwise observations of its sign.

2.1 Conditioning GPs on observation signs

Let Z be a Gaussian process on $\mathcal{D}$, with constant mean function with value $\mu^Z \in \mathbb{R}$ and stationary covariance function k^Z. Given a set of points $x_1, \ldots, x_n \in \mathcal{D}$ and corresponding observations $Z_n = (Z(x_1), \ldots, Z(x_n))^\top$, GP regression typically amounts to using the posterior mean $m_n^Z(x, z_n) = \mathbb{E}(Z(x)|Z_n = z_n)$ and covariance $k_n^Z(x) = \mathrm{Var}(Z(x)|Z_n = z_n)$, for $z_n \in \mathbb{R}^n$.

Now, in the classification setting, Z is a latent process and z_n is not available. We propose here to predict $\mathbf{1}_{Z(x)>0}$ given the sign of Z_n, that is, we consider the conditional non-failure

probability

$$\mathrm{P}_{\mathrm{nf}}(x) = \mathbb{P}\left(Z(x) > 0 | \operatorname{sign}(Z_n) = s_n\right),$$

where $s_n = (i_1, \ldots, i_n)^\top$ with $i_1, \ldots, i_n \in \{0, 1\}$ and $\operatorname{sign}(v) = (\mathbf{1}_{v_1>0}, \ldots, \mathbf{1}_{v_n>0})^\top$ for $v = (v_1, \ldots, v_n) \in \mathbb{R}^n$.

To our knowledge, there is no exact integral-free expression of $\mathrm{P}_{\mathrm{nf}}(x)$. The following lemma provides an expression of $\mathrm{P}_{\mathrm{nf}}(x)$ that is more amenable to numerical approximation.

Lemma 1. *For $s_n \in \{0,1\}^n$, let $\phi_{s_n}^{Z_n}$ be the conditional p.d.f. of Z_n given $\operatorname{sign}(Z_n) = s_n$. Let us define, for $a \in \mathbb{R}$, $b \geq 0$,*

$$\bar{\Phi}\left(\frac{a}{b}\right) = \begin{cases} 1 - \Phi\left(\frac{a}{b}\right) & \text{if } b \neq 0 \\ \mathbf{1}_{-a>0} & \text{if } b = 0 \end{cases},$$

where Φ is the standard Gaussian c.d.f. Then we have

$$\mathrm{P}_{\mathrm{nf}}(x) = \int_{\mathbb{R}^n} \phi_{s_n}^{Z_n}(z_n) \bar{\Phi}\left(\frac{-m_n^Z(x, z_n)}{\sqrt{k_n^Z(x)}}\right) dz_n.$$

Proof. The proof is deferred to Appendix A. □ □

Because of Lemma 1, we suggest the following algorithm to approximate $\mathrm{P}_{\mathrm{nf}}(x)$.

Algorithm 1.

1. *Sample $z_n^{(1)}, \ldots, z_n^{(N)} \in \mathbb{R}^n$ from the p.d.f. $\phi_{s_n}^{Z_n}$.*
2. *For any $x \in \mathcal{D}$, approximate $\mathrm{P}_{\mathrm{nf}}(x)$ by*

$$\widehat{\mathrm{P}_{\mathrm{nf}}}(x) = \frac{1}{N} \sum_{i=1}^{N} \bar{\Phi}\left(\frac{-m_n^Z(x, z_n^{(i)})}{\sqrt{k_n^Z(x)}}\right).$$

The benefit of Algorithm 1 is that Step 1, which is the most costly, has to be performed only once (independently of $x \in \mathcal{D}$). In this step, $z_n^{(1)}, \ldots, z_n^{(N)}$ can be sampled by a basic rejection method (sampling Z_n from its Gaussian p.d.f. ϕ^{Z_n} until the signs of Z_n match $i_1, \ldots, i_n$), by a more advanced rejection method called Rejection Sampling from the Mode (RSM) [18], or by more involved Markov Chain Monte Carlo (MCMC) methods [3, 32, 22], see also their presentations in [17]. Step 2 is not costly and can be repeated for many inputs x.

2.2 Likelihood computation and optimization

Let $\{k_\theta^Z; \theta \in \Theta\}$ be a set of stationary covariance functions on $\mathcal{D}$ with $\Theta \subset \mathbb{R}^p$. Typically, θ consists of an amplitude term and one or several lengthscale terms [25, 26]. We aim at selecting a constant mean function for Z with value $\mu \in \mathbb{R}$ and a covariance parameter θ. Let us first consider two pairs $(\theta_1, \mu_1), (\theta_2, \mu_2) \in \Theta \times \mathbb{R}$ for which $k_{\theta_1}^Z / k_{\theta_1}^Z(0) = k_{\theta_2}^Z / k_{\theta_2}^Z(0)$ and $\mu_1 / (k_{\theta_1}^Z(0))^{1/2} = \mu_2 / (k_{\theta_2}^Z(0))^{1/2}$. Then, one can check that the distribution of the sign process $\{1_{Z(x)>0}; x \in \mathcal{D}\}$ is the same when Z has mean and covariance function μ_1 and k_{θ_1} or μ_2 and

k_{θ_2}. Hence, it is sufficient to let $\{k_\theta^Z; \theta \in \Theta\}$ be a set of stationary correlation function and to let $\mu \in \mathbb{R}$ be unrestricted.

For $s_n \in \{0,1\}^n$, let $\mathbb{P}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n)$ be the probability that $\mathrm{sign}(Z_n) = s_n$, calculated when Z has mean function μ and covariance function k_θ. Then, the maximum likelihood estimators for μ and θ are

$$(\hat{\mu}, \hat{\theta}) \in \underset{(\mu,\theta)\in\mathbb{R}\times\Theta}{\mathrm{argmax}}\ \mathbb{P}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n). \tag{1}$$

The likelihood criterion to optimize is the probability of an orthant of $\mathbb{R}^n$, evaluated under a multidimensional Gaussian distribution. Several advanced Monte Carlo methods exist to approximate this probability [3, 6, 1]. In addition, stochastic approximations of the gradient of $\mathbb{P}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n)$ with respect to (μ, θ) can be obtained from conditional realizations of Z_n given $\mathrm{sign}(Z_n) = s_n$. Calculations are provided in Appendix B.

2.3 Comparison with classical GPC

The model in Sections 2.1 and 2.2 can be written as

$$I_i = \mathbf{1}_{Z(x_i)>0} \ \text{ for } i = 1, \ldots, n \text{ and } \ I = \mathbf{1}_{Z(x)>0}, \tag{2}$$

where $I_1, \ldots, I_n \in \{0,1\}$ are observed and $I \in \{0,1\}$ is to be predicted. In the model (2), the parameters to estimate are the constant mean $\mu \in \mathbb{R}$ and the correlation parameter θ for Z.

Another widely used Gaussian process-based classification model is the one given in [25, 21]. In this model, there is again a Gaussian process Z and, conditionally on $Z(x_1), \ldots, Z(x_n), Z(x)$, the variables $I_1, \ldots, I_n, I$ are independent and take values 0 or 1. Furthermore, with again $Z_n = (Z(x_1), \ldots, Z(x_n))^\top$,

$$P(I_i = 1|Z_n, Z(x)) = \mathrm{sig}(\sigma_f Z(x_i)) \ \text{ for } i = 1, \ldots, n \text{ and } \ P(I = 1|Z_n, Z(x)) = \mathrm{sig}(\sigma_f Z(x)), \tag{3}$$

where $\mathrm{sig} : \mathbb{R} \to (0,1)$ is a continuous strictly increasing function satisfying $\mathrm{sig}(t) \to 0$ as $t \to -\infty$ and $\mathrm{sig}(t) \to 1$ as $t \to +\infty$ and with $\sigma_f > 0$. For instance, a classical choice in [25, 21] is the logit function defined by $\mathrm{sig}(x) = e^x/(1+e^x)$.

In the model (3), it is assumed in [25, 21] that the mean function of Z is zero[1]. The parameter to estimate for the covariance function of Z is θ, from the set of stationary covariance functions $\{k_\theta; \theta \in \Theta\}$. The parameter σ_f also has to be estimated. Since the mean function of Z is assumed to be zero, one can see that pairs $(\theta_1, \sigma_{f,1})$ and $(\theta_2, \sigma_{f,2})$ for which $\sigma_{f,1}^2 k_{\theta_1} = \sigma_{f,2}^2 k_{\theta_2}$ give the same distribution of $I_1, \ldots, I_n, I$ in (3). Thus, for the model (3), we let $\{k_\theta; \theta \in \Theta\}$ be a set of correlation functions, and $\sigma_f \geq 0$ has to be estimated as well.

We now compare our introduced model (2) with (3). The framework (2) corresponds to the limit of the model in (3), as $\sigma_f \to +\infty$. Indeed, let $\mathrm{sgn}(t) = 0$ if $t < 0$, $\mathrm{sgn}(t) = 1/2$ if $t = 0$ and $\mathrm{sgn}(t) = 1$ if $t > 0$. Then, as observed in [21], when $\sigma_f = +\infty$, we have $P(I = 1|Z_n, Z(x)) = \mathrm{sgn}(Z(x))$ and $P(I_i = 1|Z_n, Z(x)) = \mathrm{sgn}(Z(x_i))$, for $i = 1, \ldots, n$. Since the components of Z_n take values 0 with zero probability, (2) and (3) indeed give identical distributions of $(I_1, \ldots, I_n, I)$ when $\sigma_f = +\infty$.

In the framework described in Section 1, repeated calls to the code function, for the same input x, either all crash or all successfully return an output value. Furthermore, in

[1] A constant mean function could be incorporated and estimated with no additional complexity.

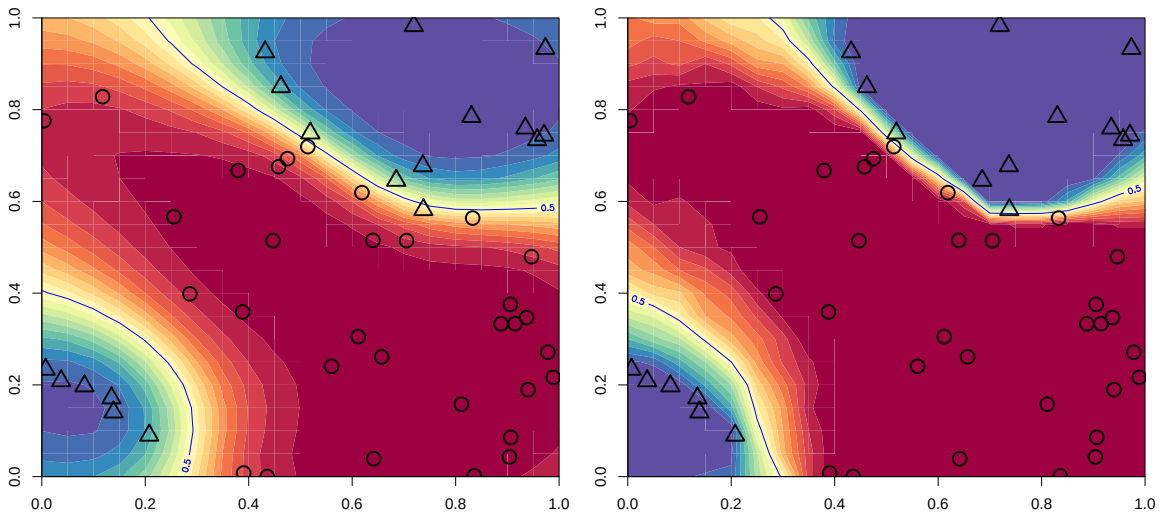


Figure 1: GPC model (3) based on EP and the logit function (left) and our GPC model (2) (right).

the setting of the industrial case study in Section 6, the set of inputs x for which the code function crashes (or returns an output value) has regularity properties. Hence, the model (2) is more appropriate than the model (3) (especially with small values of σ_f). Thus, it is beneficial to use the model (2) directly, rather than using the model (3), which entails additional cost and error risk with the estimation of σ_f. Figure 1 shows the two models built on a 50-point design of experiments on a 2D toy problem. While the model (3), based on expectation-propagation (EP) [21], returns a function with smooth transitions, our model (2) returns a much sharper function, which is more appropriate for our framework of deterministic failures. In addition, the model (1) returns conditional crash probabilities that are not equal to exactly zero or one for input points points with observed binary outputs. In contrasts, the conditional probabilities returned by our model are exactly zero or one for these input points with observed outputs. Again this is more appropriate for deterministic failures.

In terms or inference, we have discussed in Section 2.1 that, for a fixed θ, the only costly step for the model (2) is to sample realizations of the p.d.f. $\phi_{s_n}^{Z_n}$. This p.d.f. is that of a truncated Gaussian vector (restricted to an orthant of $\mathbb{R}^n$). Instead, the distribution to sample with the model (3) (the conditional distribution of Z_n given $I_1 = i_1, \ldots, I_n = i_n$) has a density on $\mathbb{R}^n$, which value at $z_1, \ldots, z_n$ is proportional to

$$\left(\prod_{j=1}^{n} \mathrm{sig}(\sigma_f z_j)^{i_j} [1 - \mathrm{sig}(\sigma_f z_j)]^{1-i_j} \right) \phi^{Z_n}(z_1, \ldots, z_n), \tag{4}$$

where ϕ^{Z_n} is the Gaussian density function of Z_n. The density in (4) is, arguably, more complicated than a truncated Gaussian density function, for which many implemented algorithms are available, as discussed above when introducing the references [3, 32, 22, 17].

In [25, 21], several approximations of the distribution in (4) by multidimensional Gaussian distributions are presented (in particular, the Laplace and EP approximations, the variational method and the Kullback-Leibler method). These approximations are usually relatively fast to obtain, from local optimization methods. Yet, they are approximations of a non-Gaussian

distribution, and do not come (to our knowledge) with theoretical guarantees. Similarly, for parameter estimation, the likelihood function of $I_1, \dots, I_n$ is approximated, and the approximation is maximized with respect to θ and σ_f. This yields a relatively fast procedure for estimating θ and σ_f, for which, again, no theoretical guarantees are available.

In contrast, with the model (2), the simulation from the truncated conditional distribution $\phi_{s_n}^{Z_n}$, with $s_n = (i_1, \dots, i_n)$ (Section 2.1) and the maximum likelihood estimation of θ and μ (Section 2.2) do not rely on approximations, and are based on Monte Carlo techniques rather than optimization. Hence, compared to the model (3), the inference in the model (2) may come with computational cost, but has more accuracy guarantees. For instance, there exists a large body of literature guaranteeing the convergence of Monte Carlo algorithms, for long runs [19].

We remark that, with the model (3) and the above discussed Gaussian approximation, once the conditional distribution of $(Z(x_1), \dots, Z(x_n))$ given $I_1 = i_1, \dots, I_n = i_n$ is approximated, it is not costly to obain the conditional distribution of I given $(I_1, \dots, I_n)$ (see [25, 21]). This is similar to Algorithm 1 for the model (2).

Finally, the constrained optimization problems addressed in the present article are of the form $\max_{x \in \mathcal{A}} f(x)$, where $\mathcal{A}$ is a fixed unknown subset. It is hence very natural to use the Bayesian prior $\{x \in \mathcal{D}; Z(x) > 0\}$ on $\mathcal{A}$, which is obtained from our classification model (2). In contrasts, the classification model (3) does not provide a fixed set of admissible inputs, since any x in $\mathcal{D}$ has non-zero probabilities to yield both categories of the binary output. As a consequence, our suggested acquisition function in (9) below, and particularly the definition of the current admissible maximum M_q there, rely on the classification model (2). Hence, also the proof of convergence in Section 4 relies on the classification model (2).

3 Bayesian optimization with crash constraints

Let us now address the case of optimization in the presence of computational failures, that is introduced in Section 1. This problem requires a model for the objective function in addition to the one for the constraint. In this section, we first consider the problem of joint modeling, then its use in a Bayesian optimization algorithm.

3.1 Joint modeling of the objective and constraint

Let us consider two independent continuous Gaussian processes Y and Z from $\mathcal{D}$ to $\mathbb{R}$. In our framework, for an input point x, we can observe the pair

$$(\text{sign}[Z(x)], \text{sign}[Z(x)]Y(x)]). \tag{5}$$

That is, we observe whether the computation fails ($Z(x) \leq 0$) or not, and in case of computation success, we observe the computation output $Y(x)$.

For Z, as in Section 2, we select a constant mean $\mu^Z \in \mathbb{R}$ and a correlation parameter $\theta_Z \in \Theta_Z$, where $\{k_{\theta_Z}^Z; \theta_Z \in \Theta_Z\}$ is a set of correlation functions with $\Theta_Z \subset \mathbb{R}^{p_Z}$. For Y, we select a constant mean $\mu^Y \in \mathbb{R}$ and a covariance parameter $\theta_Y \in \Theta_Y$, where $\{k_{\theta_Y}^Y; \theta_Y \in \Theta_Y\}$ is a set of covariance functions with $\Theta_Y \subset \mathbb{R}^{p_Y}$.

Let the pair (5) be observed for the input points $x_1, \dots, x_n \in \mathcal{D}$. For $j = 1, \dots, n$ we let $I_j = \text{sign}(Z(x_j))$ and consider the observation $(i_1, \dots, i_n, i_1 y_1, \dots, i_n y_n)$ of

$$(I_1, \dots, I_n, I_1 Y(x_1), \dots, I_n Y(x_n)). \tag{6}$$

In the next lemma, we show that a likelihood can be defined for these $2n$ observations. Since the distribution of $I_iY(x_i)$ is a mixture of continuous and discrete distributions, we add a random continuous noise in case $I_i = 0$. This random noise does not add or remove information, and is just a technicality in order to write the following lemma in terms of likelihood with respect to a simple fixed measure on $\mathbb{R}^{2n}$.

Let us introduce some notation before stating the lemma. For $s_n = (i_1, \ldots, i_n)^\top \in \{0,1\}^n$, let Y_{n,s_n} be the vector extracted from $(Y(x_1), \ldots, Y(x_n))$ by keeping only the indices $j \in \{1, \ldots, n\}$ for which $i_j = 1$. Let $\phi^Y_{\mu^Y,\theta_Y,s_n}$ be the p.d.f. of Y_{n,s_n}, calculated under the assumption that Y has a constant mean function μ^Y and covariance function $k^Y_{\theta_Y}$. For $v = (v_1, \ldots, v_n)^\top \in \mathbb{R}^n$, let v_{s_n} be the vector extracted from v by keeping only the indices $j \in \{1, \ldots, n\}$ for which $i_j = 1$.

Lemma 2. *For $j = 1, \ldots, n$, let $V_j = I_jY(x_j) + (1 - I_j)W_j$ where $W_1, \ldots, W_n$ are independent and follow the standard Gaussian distribution. Let $f_{\mu^Z,\theta_Z,\mu^Y,\theta_Y}$ be the p.d.f. of $(I_1, \ldots, I_n, V_1, \ldots, V_n)$, defined with respect to the measure $(\otimes_{i=1}^n \mu) \otimes (\otimes_{i=1}^n \lambda)$ where μ is the counting measure on $\{0,1\}$ and λ is the Lebesgue measure on $\mathbb{R}$. Then we have*

$$\begin{aligned} &f_{\mu^Z,\theta_Z,\mu^Y,\theta_Y}(i_1, \ldots, i_n, v_1, \ldots, v_n) \\ =&\mathbb{P}_{\mu^Z,\theta_Z}(I_1 = i_1, \ldots, I_n = i_n)\, \phi^Y_{\mu^Y,\theta_Y,s_n}(v_{s_n}) \left(\prod_{\substack{j=1,\ldots,n \\ i_j=0}} \phi(v_j) \right), \end{aligned}$$

where ϕ is the standard Gaussian p.d.f. and $\mathbb{P}_{\mu^Z,\theta_Z}(\cdot)$ is the probability of an event, calculated under the assumption that Z has mean and covariance functions μ^Z and $k^Z_{\theta_Z}$.

Proof. The proof is deferred to Appendix A. □ □

In view of Lemma 2, the maximum likelihood estimators of $\mu^Z, \theta_Z, \mu^Y, \theta_Y$ are

$$(\hat{\mu}^Z, \hat{\theta}_Z) \in \underset{(\mu^Z,\theta_Z)\in\mathbb{R}\times\Theta_Z}{\operatorname{argmax}} \mathbb{P}_{\mu^Z,\theta_Z}(I_1 = i_1, \ldots, I_n = i_n) \tag{7}$$

and

$$(\hat{\mu}^Y, \hat{\theta}_Y) \in \underset{(\mu^Y,\theta_Y)\in\mathbb{R}\times\Theta_Y}{\operatorname{argmax}} \phi^Y_{\mu^Y,\theta_Y,s_n}(Y_q), \tag{8}$$

with Y_q the realization of Y_{n,s_n}.

The likelihood maximization in (7) can be tackled as in Section 2. The likelihood maximization in (8) corresponds to the standard maximum likelihood in Gaussian process regression.

Once the likelihood has been optimized, it is common practice to take the optimal mean and variance as face value and neglect the uncertainty associated with their estimation ("plugin" approach). Under this assumption, we provide in the next lemma the conditional distributions of Z and Y, given the observations in (6).

Lemma 3. *Conditionally on*

$$I_1 = i_1, I_1Y(x_1) = i_1y_1, \ldots, I_n = i_n, I_nY(x_n) = i_ny_n,$$

the stochastic processes Y and Z are independent. The stochastic process Z follows the conditional distribution of Z given $I_1 = i_1, \ldots, I_n = i_n$ and the stochastic process Y follows the conditional distribution of Y given $Y_{n,s_n} = Y_q$ with Y_{n,s_n} as in Lemma 2 and with Y_q defined as after (8).

Proof. The proof is deferred to Appendix A. □ □

In other words, conditionally on the observations, Z is conditioned on its signs at $x_1, ..., x_n$, and Y is conditioned on its values at the x_i's for which $Z(x_i) > 0$. Hence, conditional inference on Z can be carried out as described in Section 2, and Y follows the standard Gaussian conditional distribution in Gaussian process regression.

3.2 Acquisition function and sequential design

Given the observations in (6), we now suggest an acquisition function to be optimized to select a new observation point $x_{n+1} \in \mathcal{D}$ given a set of existing n observations. We follow the celebrated expected improvement principle [20, 12], adapted to the partial observation setting. Thus, we choose:

$$x_{n+1} \in \operatorname*{argmax}_{x \in \mathcal{D}} \mathbb{E}\left(\mathbf{1}_{Z(x)>0} \left[Y(x) - M_q\right]^+ \middle| \mathcal{F}_n\right), \tag{9}$$

where, with $\sigma(\cdot)$ the sigma-algebra generated by a set of random variables,

$$\mathcal{F}_n = \sigma\left(I_1, I_1 Y(x_1), \ldots, I_n, I_n Y(x_n)\right) \tag{10}$$

denotes our observation event and

$$M_q = \max_{i=1,\ldots,n; Z(x_i)>0} Y(x_i)$$

with the convention $M_q = -\infty$ if $Z(x_1) \leq 0, \ldots, Z(x_n) \leq 0$. We call $\mathbb{E}\left(\mathbf{1}_{Z(x)>0} \left[Y(x) - M_q\right]^+ \middle| \mathcal{F}_n\right)$ the expected improvement with failure (EIF).

As in Lemma 3, for $s_n = (i_1, \ldots, i_n) \in \{-1, 1\}^n$, we let Y_{n,s_n} be the vector extracted from $(Y(x_1), \ldots, Y(x_n)$ by keeping only the indices $j \in \{1, \ldots, n\}$ for which $i_j = 1$. Thanks to this lemma we have

$$\begin{aligned}\mathbb{E}\left(\mathbf{1}_{Z(x)>0} \left[Y(x) - M_q\right]^+ \middle| \mathcal{F}_n\right) =& \mathbb{P}\left(Z(x) > 0 \middle| I_1 = i_1, \ldots, I_n = i_n\right) \mathbb{E}\left(\left[Y(x) - M_q\right]^+ \middle| Y_{n,s_n}\right) \\ =& \mathrm{P_{nf}}(x) \times \mathrm{EI}(x),\end{aligned}$$

say. Hence, the EIF is equal to the product of the conditional probability of non-failure $\mathrm{P_{nf}}(x)$ (conditionally on the signs of Z) and of the standard expected improvement $\mathrm{EI}(x)$ (conditionally on the observed values of Y). This criterion is similar to the one proposed in [29] and later [5] for quantifiable constraints. The criterion in [16] is slightly different in order to favor the exploration of the boundary, but at the loss of a consistent definition of *improvement*:

$$\mathrm{EI}(x)^{\alpha_1} \times \left[\frac{2\mathrm{P_{nf}}(x)\left(1 - \mathrm{P_{nf}}(x)\right)}{\mathrm{P_{nf}}(x) - 2w\mathrm{P_{nf}}(x) + w^2}\right]^{\alpha_2},$$

with α_1, α_1 and w positive parameters.

The conditional probability of non-failure $\mathrm{P_{nf}}(x)$ can be approximated by $\widehat{\mathrm{P_{nf}}}(x)$ from Algorithm 1. In this algorithm, the first step is costly but needs to be performed only once independently of x, hence outside the optimization loop (9). Then, $\widehat{\mathrm{P_{nf}}}(x)$ is a smooth function of x that is not costly to evaluate.

Turning to the expected improvement $\mathrm{EI}(x)$, let q be the length of Y_{n,s_n}. For a realization $(y_1, \ldots, y_n)$ of $Y(x_1), \ldots, Y(x_n)$, let Y_q be the vector extracted from $(y_1, \ldots, y_n)$ by keeping only the indices $j \in \{1, \ldots, n\}$ for which $i_j = 1$. Hence, Y_q is a realization of Y_{n,s_n}.

Let $x \to m_q^Y(x, Y_q)$ and $(x, y) \to k_q^Y(x, y)$ be the conditional mean and covariance functions of Y given $Y_{n,s_n} = Y_q$. Let also $k_q^Y(x) = k_q^Y(x, x)$. It is well-known (see e.g.[12]) that

$$\mathrm{EI}(x) = \left(m_q^Y(x, Y_q) - M_q\right) \Phi\left(\frac{m_q^Y(x, Y_q) - M_q}{\sqrt{k_q^Y(x)}}\right) + \sqrt{k_q^Y(x)}\phi\left(\frac{m_q^Y(x, Y_q) - M_q}{\sqrt{k_q^Y(x)}}\right), \quad (11)$$

with Φ and ϕ the c.d.f. and p.d.f., respectively, of the standard Gaussian distribution.

Solving the optimization problem in (9) is greatly facilitated by analytical gradients, which are available in our case. Calculations are provided in Appendix C.

4 Convergence

In this section, we prove the convergence of the sequential choice of observation points given by (9), with the slight difference that (9) is replaced by

$$x_{n+1} \in \operatorname*{argmax}_{x \in \mathcal{D}} \mathbb{E}\left(\max_{\substack{u \in \mathcal{D} \\ \mathbb{P}(Z(u)>0|\mathcal{F}_{n,x})=1 \\ \mathrm{var}(Y(u)|\mathcal{F}_{n,x})=0}} Y(u) - \tilde{M}_q \,\middle|\, \mathcal{F}_n\right), \quad (12)$$

with

$$\tilde{M}_q = \max_{\substack{x \in \mathcal{D} \\ \mathbb{P}(Z(x)>0|\mathcal{F}_n)=1 \\ k_q^Y(x)=0}} Y(x) \quad (13)$$

and where $\mathcal{F}_{n,x}$ is the sigma algebra generated by the random variables

$$I_1, I_1 Y(x_1), \ldots, I_n, I_n Y(x_n), \mathbf{1}_{Z(x)>0}, \mathbf{1}_{Z(x)>0} Y(x).$$

We remark that M_q corresponds to the maximum over the q observed values of Y, while $\tilde{M}_q$ is the maximum of Y over the input points x for which it is known (after the n first observations) that $Z(x) > 0$ and that $Y(x) = m_q^Y(x, Y_q)$.

The algorithms given by (9) and (12) coincide when Z and Y are non-degenerate, that is $(\xi(v_i))_{i=1,\ldots,r}$ has a non-degenerate distribution for any two-by-two distinct points $v_1, \ldots, v_r \in \mathcal{D}$, with $\xi = Z$ and $\xi = Y$. These two algorithms can be different when Y or Z are degenerate (which can happen, for instance, when their trajectories are known to satisfy symmetry properties, see e.g. [8]).

Hence, using (12) in the case of degenerate processes enables us to take into account that there are cases where some input points can be known to yield higher values of Y than $\max Y_q$ and to be valid. Furthermore, (12) takes into account the fact that, for $u \notin \{x_1, \ldots, x_n, x\}$,

the values $\mathbf{1}_{Z(u)>0}$ and $Y(u)$ can have zero uncertainty when $\mathbf{1}_{Z(x)>0}$ and $\mathbf{1}_{Z(x)>0}Y(x)$ are observed.

Following [33], we say that a Gaussian process ξ with continuous trajectories has the no-empty ball property (NEB) if, for any $x_0 \in \mathcal{D}$, for any $\epsilon > 0$,

$$\inf_{\substack{n\in\mathbb{N} \\ x_1,\ldots,x_n\in\mathcal{D} \\ ||x_i-x_0||\geq\epsilon,\forall i}} \operatorname{var}(\xi(x_0)|\xi(x_1),\ldots,\xi(x_n)) > 0.$$

Many standard covariance kernels provide Gaussian processes having the NEB. Indeed, a sufficient condition for the NEB is that the covariance kernel is stationary with a spectral density decreasing no faster than an inverse polynomial at infinity [33]. The most notable covariance function that does not have the NEB property is the squared exponential covariance function [34].

We are now in position to state the convergence result.

Theorem 1. *Let $\mathcal{D}$ be a compact hypercube of $\mathbb{R}^d$. Let $(X_i)_{i\in\mathbb{N}}$ be such that $X_1 = x_1$ is fixed in $\mathcal{D}$ and, for $n \geq 1$, X_{n+1} is selected by (12).*

1. *Assume that Y and Z are Gaussian processes with continuous trajectories. Then, a.s. as $n \to \infty$, $\sup_{x\in\mathcal{D}} \mathbb{P}(Z(x) > 0|\mathcal{F}_n)(m_q^Y(x, Y_{n,s_n}) - \tilde{M}_q)^+ \to 0$ and $\sup_{x\in\mathcal{D}} \mathbb{P}(Z(x) > 0|\mathcal{F}_n)k_q^Y(x) \to 0$.*

2. *If furthermore Y and Z have the NEB property, then $(X_i)_{i\in\mathbb{N}}$ is a.s. dense in $\mathcal{D}$. As a consequence $\max_{i=1,\ldots,n;Z(X_i)>0} Y(X_i) \to \max_{u\in\mathcal{D};Z(u)>0} Y(u)$ a.s. as $n \to \infty$.*

Proof. Theorem 1 is proved by combining and extending the techniques from [33, 2]. The proof is deferred to Appendix A. □ □

The first part of Theorem 1 states that, as $n \to \infty$, all the input points x provide an asymptotically negligible expected improvement (similarly, a negligible information). Indeed, they have a crash probability that are almost equal to one, or they have a conditional variance that goes to zero and a conditional mean that is no larger than the current maximum $\tilde{M}_q$.

The second part of the theorem shows that, as a consequence, the sequence of observation points is dense as $n \to \infty$ and that the observed maximum converges to the global maximum. The nature of this convergence result is similar to those given in the unconstrained case in [33, 2]. This convergence result guarantees that our suggested algorithm will not leave unexplored regions. Another formulation of Theorem 1 is that our suggested algorithm will not be trapped in local maxima of Y.

5 Simulations on 2D-gaussian processes

In this section the behavior of our optimization algorithm with crash constraints, that we now call Expected Feasible Improvement with Gaussian Process Classification with signs (*EFI GPC sign*), is studied on simulated 2D-gaussian processes. We compare this algorithm with the optimization procedure defined as in Section 3.2, but where the probabilities of satisfying the constraints are obtained from the classical Gaussian process classifier of [25, 21], based on Expectation Propagation, see Section 2.3. This second algorithm is called *EFI GPC EP*.

	$\theta_Z = 0.1$	$\theta_Z = 0.3$
$\theta_Y = 0.1$	case 1	case 3
$\theta_Y = 0.3$	case 2	case 4

Table 1: Studied ranges for the simulations.

5.1 Simulations setting

The two algorithms are run on a function $f : [0,1]^2 \to \mathbb{R}$ taken as a realization of a 2D Gaussian process Y. The correlation kernel is a tensorized *Matern5_2* kernel with the same correlation length parameter θ_Y in each direction [26]. Observation of f is conditioned by a function $s : [0,1]^2 \to \{0,1\}$ such that s is a realization of $\mathbf{1}_{Z>0}$ where Z is a 2D Gaussian process independent of Y. Z is also chosen with a tensorized *Matern5_2* kernel with the same parameter θ_Z in each direction.

Two levels of ranges for θ_Y and θ_Z are considered to represent different behaviours of the functions f and s. Four cases are studied and summarized in Table 1. In our simulations, the processes Y et Z have mean $\mu^Y = \mu^Z = 0$ and variance $\sigma_Y^2 = \sigma_Z^2 = 1$.

The initial Design of Experiments (DoE) is a maximin Latin hypercube design of 9 points. Then, 41 points are sequentially added according to (14) :

$$x_{n+1} \in \text{argmax}_{x \in \mathcal{D}} \mathrm{P}_{\mathrm{nf}}(x) \times \mathrm{EI}(x). \tag{14}$$

Note that, as discussed above, $\mathrm{P}_{\mathrm{nf}}(x)$ is calculated either through our algorithm *GPC sign* or by a classical GPC, noted *GPC EP*.

5.2 Results of our method *EFI GPC sign*

In the following we define the regret at step n

$$R_n = \max_{x \in [0,1]^2, Z(x)>0} Y(x) - \max_{1 \le i \le n, Z(x_i)>0} Y(x_i).$$

It represents the gap between the global maximum and the current maximum value of the output on the current design of experiments $\{x_1, \ldots, x_n\}$. We consider 20 different realizations of Y and Z. On Figure 2 the mean of R_n is plotted along the iteration steps in the four different cases described in Table 1. It can be noticed that in each case the algorithm converges to the global maximum. The convergence speed depends on the range level. When the correlation length of the process Y is high, i.e. $\theta_Y = 0.3$, the problem appears to be much easier, independently on the correlation length of θ_Z. To a lesser extent, a high range of the process Z also helps to accelerate the convergence.
The evolution of the Number of Successes (NoS) with iteration is plotted on Figure 3. In *case 2* and *case 4* ($\theta_Y = 0.3$), the best point is rapidly found, exploration steps are then more numerous and the increase of NoS slows down.

Range parameter estimations for the processes Y and Z are given in the top table of Table 2. The bottom table gives the estimation of trend and variance parameters for both processes. It can be observed that parameter estimation for the process Z is difficult since only signs are available. For instance, μ_Z is overestimated. This reflects under-sampling of crash areas

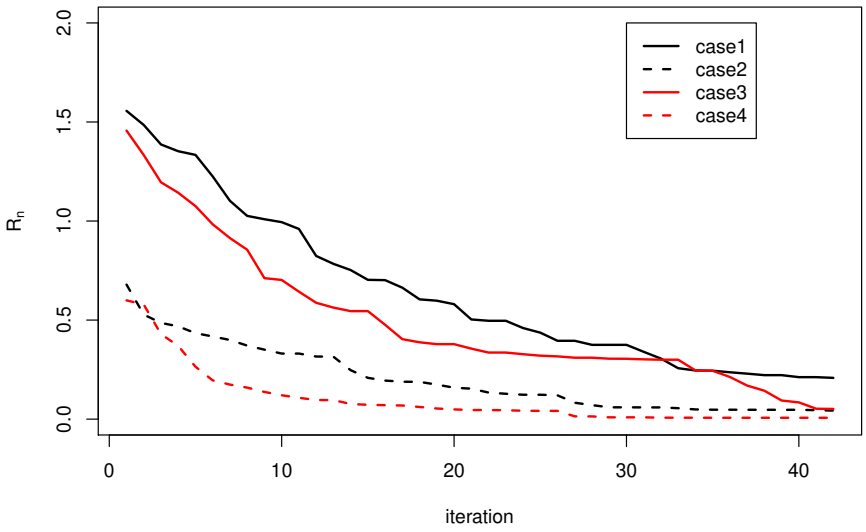


Figure 2: Evolution of R_n (in mean) along the iteration steps. Four cases of range values are considered (see Table 1). Parameters are estimated by maximum likelihood.

that provide no information on the process Y. The situation is different for the process Y. Despite failure events, available information and estimation accuracy increase with iterations.

5.3 Comparison between *EFI GPC sign* and *EFI GPC EP*

The performances of both methods (*EFI GPC sign* and *EFI GPC EP*) are compared on the same simulations as previously. It can be seen on Figure 4 that the regret of *EFI GPC sign* converges more rapidly to 0. This can be explained by the fact that the number of sucesses is more important with *EFI GPC sign* than with *EFI GPC EP*, since *EFI GPC sign* avoids crash areas more often (see Figure 5). Parameters estimations of *EFI GPC EP* are given in Table 3. It can be observed that Z-parameter estimation can hardly be compared between methods since the classification models are different. Concerning the process Y, the estimated correlation parameters tend towards the real values with more iterations. We remark that the estimated values of σ_f^2 are large so that the EP classification model is close to the sign classification model.
An example of the progression of the algorithms in *case 1* ($\theta^Z = \theta^Y = 0.1$) is given on Figure 6 for *EFI GPC sign* and 7 for *EFI GPC EP* . The evolution is quite similar but *EFI GPC sign* reaches the maximum a bit earlier than *EFI GPC EP*. Moreover the number of crashes is lower with *EFI GPC sign* than with *EFI GPC EP*. Two other simulations are compared in Section D of the Appendix. Figures 12 and 13 show an example where neither method found the global maximum. Figures 14 and 15 show a situation where the global maximum is reached at the beginning of the algorithm faster for *EFI GPC sign*.

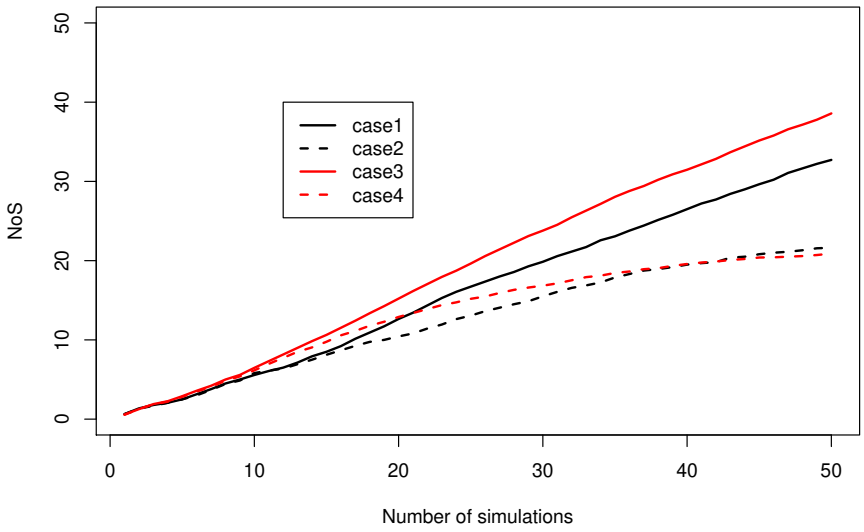


Figure 3: Evolution of NoS, Number of Succeses, (in mean) along the number of simulations. Four cases of range values are considered (see Table 1).

6 Industrial case study

The aim of this section is to find the shape of the turbomachinery of an automotive fan system such that its efficiency is the highest as possible. The geometry of the turbomachinery, more precisely the geometry of the rotor blades, is described by 15 geometric parameters: 5 chords lengths, 5 stagger angles and 5 heights of max camber (see Figure 8). A turbomachinery code has been developed by researchers of the LMFA (Laboratory of acoustics and fluid dynamics) in Ecole Centrale Lyon. It is a multi-physics 1D model based on iterative resolution of isentropic efficiency at medium radius, resolution of radial equilibrium, and deduction of blade angles through empirical correlations.

In this context we aim at selecting the geometric parameters that maximize the efficiency of the turbomachinery for a fixed input flow rate and for a fixed pressure rise. The ranges of the 15 geometric parameters are given in Table 4 (Section E of the Appendix).

The problem is that for some parameter configurations the simulation does not converge and a NA is returned. These simulation failures can be related to the empirical rules injected in the implementation of the code, that limit its validity domain. It can occur that the calculation comes out of the admissible domain, in which case the empirical correlations become inaccurate and the simulation is not valid anymore.

The issue is to find the optimal geometry considering these failures. A set of initial simulations has been run to explore failure events. We made each geometric parameter successively vary from its minimum to its maximum in Table 4 around three particular points on the diagonal of the hypercube in dimension 15: $Point1$ is close to the minimal corner of the hypercube, $Point2$ is at the center and $Point3$ is close to the maximal corner of the hypercube.

(a) Range parameters

	θ^Z	$\hat{\theta}^Z$	$\hat{\theta}^Z$	θ^Y	$\hat{\theta}^Y$	$\hat{\theta}^Y$
	true value	iteration 10	iteration 41	true value	iteration 10	iteration 41
Case 1	0.1	0.44 (0.33)	0.37 (0.30)	0.1	0.17 (0.23)	0.10 (0.02)
Case 2	0.1	0.32 (0.29)	0.24 (0.25)	0.3	0.36 (0.23)	0.32 (0.16)
Case 3	0.3	0.52 (0.35)	0.41 (0.30)	0.1	0.11 (0.08)	0.09 (0.02)
Case 4	0.3	0.49 (0.34)	0.51 (0.39)	0.3	0.34 (0.18)	0.28 (0.12)

(b) Trend and variance parameters

	$\hat{\mu}^Z$	$\hat{\mu}^Z$	$\hat{\mu}^Y$	$\hat{\mu}^Y$	$\hat{\sigma}^2_Y$	$\hat{\sigma}^2_Y$
	iteration 10	iteration 41	iteration 10	iteration 41	iteration 10	iteration 41
Case 1	0.30 (0.26)	0.64 (0.85)	-0.03 (0.57)	-0.09 (0.31)	0.68 (0.49)	0.82 (0.32)
Case 2	0.27 (0.33)	0.41 (0.39)	0.03 (0.70)	0.00 (0.53)	0.79 (0.51)	0.89 (0.70)
Case 3	0.41 (0.33)	0.51 (0.41)	-0.13 (0.36)	-0.06 (0.30)	0.66 (0.30)	0.83 (0.21)
Case 4	0.26 (0.20)	0.48 (0.41)	-0.16 (0.56)	-0.07 (0.49)	0.74 (0.58)	0.75 (0.51)

Table 2: Method *EFI GPC sign* at step 10 and 41: (a) Estimation of θ^Z and θ^Y, (b) Estimation of μ^Z (true value is 0), μ^Y (true value is 0) and σ^2_Y (true value is 1). Mean (standard deviation) over 20 simulations.

Coordinates are given in Table 5 (Section E of the Appendix). The results of these simulations in terms of NA values are represented in Figure 9. Yellow areas are failures. It is more frequent to observe a NA at the edges of the hypercube and near $Point1$. It can be observed on Figure 10 that highest efficiencies are obtained around the center of the hypercube.

Remark : To simplify the reading on all the figures of this section the points are represented in a normalized domain $[-1, 1]^{15}$.

Both methods *EFI GPC sign* and *EFI GPC EP* are applied from an initial maximin LHS composed of 75 points. Among them 18 simulations failed. The output range of converged simulations is roughly $[0.3, 0.7]$ and the highest observed efficiency is 0.70. 100 simulations are then successively chosen accordingly to (14). As it can be seen on Figure 11 the objective of improving efficiency is reached: a max value of 0.75 is observed at iteration 22 (resp. 25) for algorithm *EFI GPC sign* (resp. *EFI GPC EP*).
Several types of behavior of the algorithms can be observed along the iterations. At the beginning of the algorithms the simulations are added to locally improve efficiency : only one crash is met over the 20 first points. Then and especially above iteration 50 the algorithms explore other uncertainty areas and more failures occur. It can be noticed on Figure 11 that our algorithm *EFI GCP sign* better avoids crash areas than *EFI GCP EP* does. Only 23 failures are met over 100 iterations with *EFI GCP sign* whereas 34 crashes occur with *EFI GCP EP*.

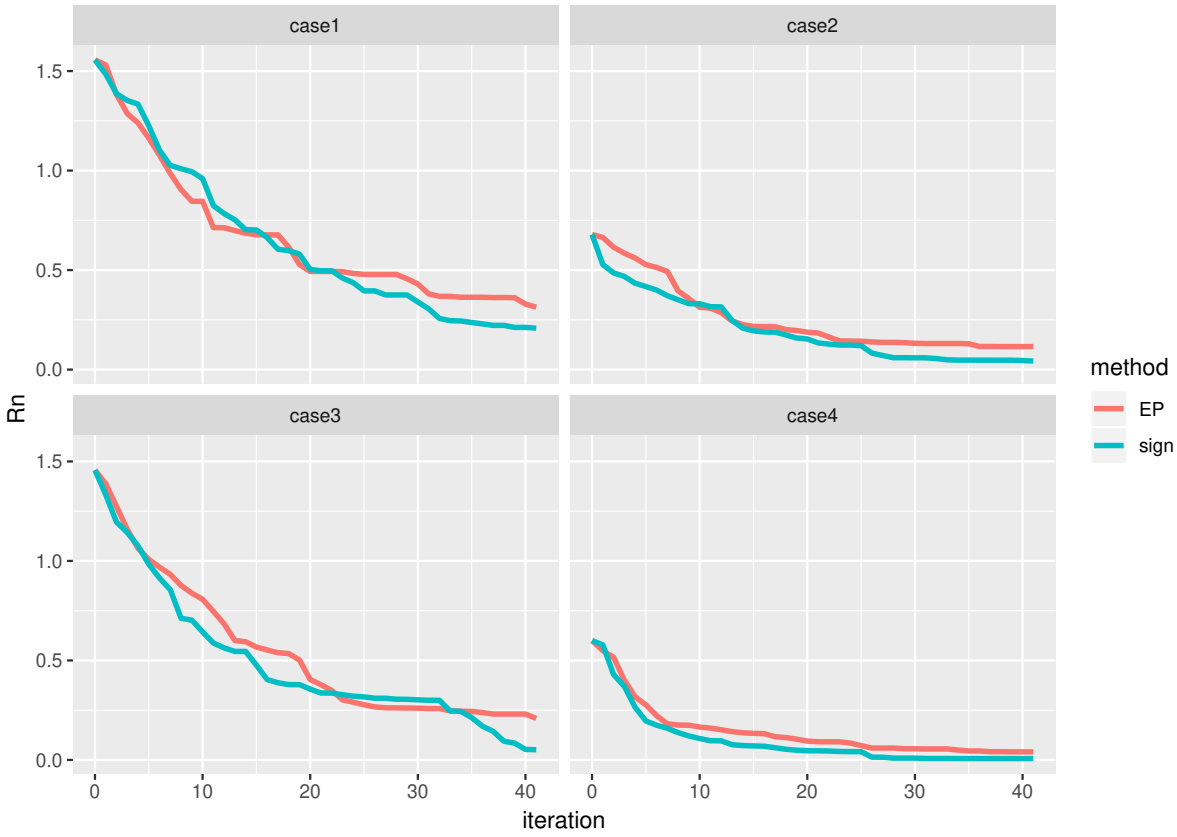


Figure 4: Evolution of R_n (in mean) along the iteration steps : *EFI GPC sign* in green and *EFI GPC EP* in red. Four cases of range values are considered (see Table 1). Parameters are estimated by maximum likelihood.

7 Conclusion

In this work, we addressed the problem of global optimization of a black-box function under "crash" constraints. To do so, we revisited Gaussian process classification with a model based on observation signs. This model exhibited sharp classification boundaries, which were appropriate in our framework, and allowed us to propose the first algorithm with guaranteed convergence for this problem. Numerical experiments showed promising results, in particular as the algorithm leads to many less simulation failures (in a sense, wasted computational resources) that the current state-of-the-art.

For simplicity, we considered in this paper the case where simulations were run one-at-a-time. A possible extension of this work is to treat the case of batch-sequential strategies, in the spirit of [7, 35]. We believe that both theoretical and practical aspects could be addressed without major difficulty. Another extension with practical importance would be to tackle problems for which either the objective function and / or the failure events are stochastic; however, a large portion of the proofs proposed here would not apply directly. Finally, convergence rates have not been considered here. Following [4, 31], future work may address this problem.

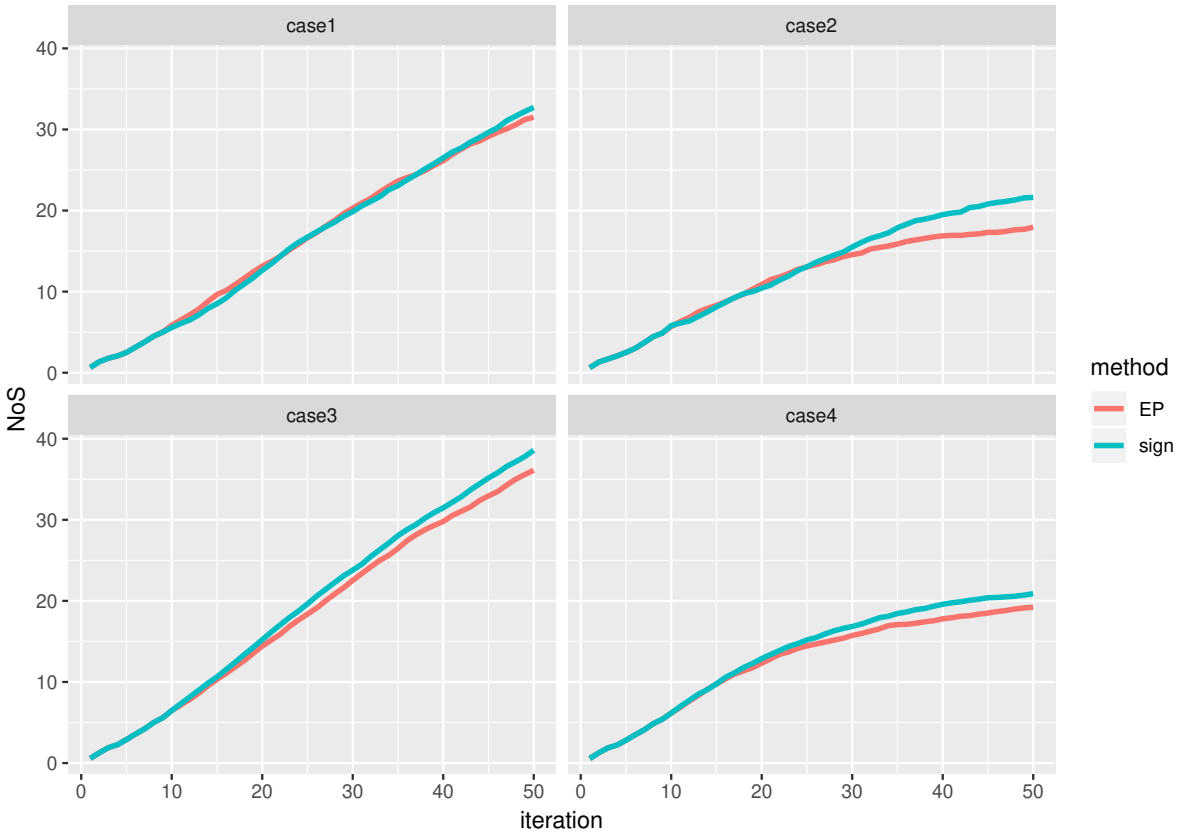


Figure 5: Evolution of NoS (Number of Successes), in mean, with the number of iterations : *EFI GPC sign* in green and *EFI GPC EP* in red. Four cases of range values are considered (see Table 1).

Acknowledgments

This work was partly funded by the ANR project PEPITO. We are grateful to Jean-Marc Azaïs and David Ginsbourger for discussions on the topic of this paper.

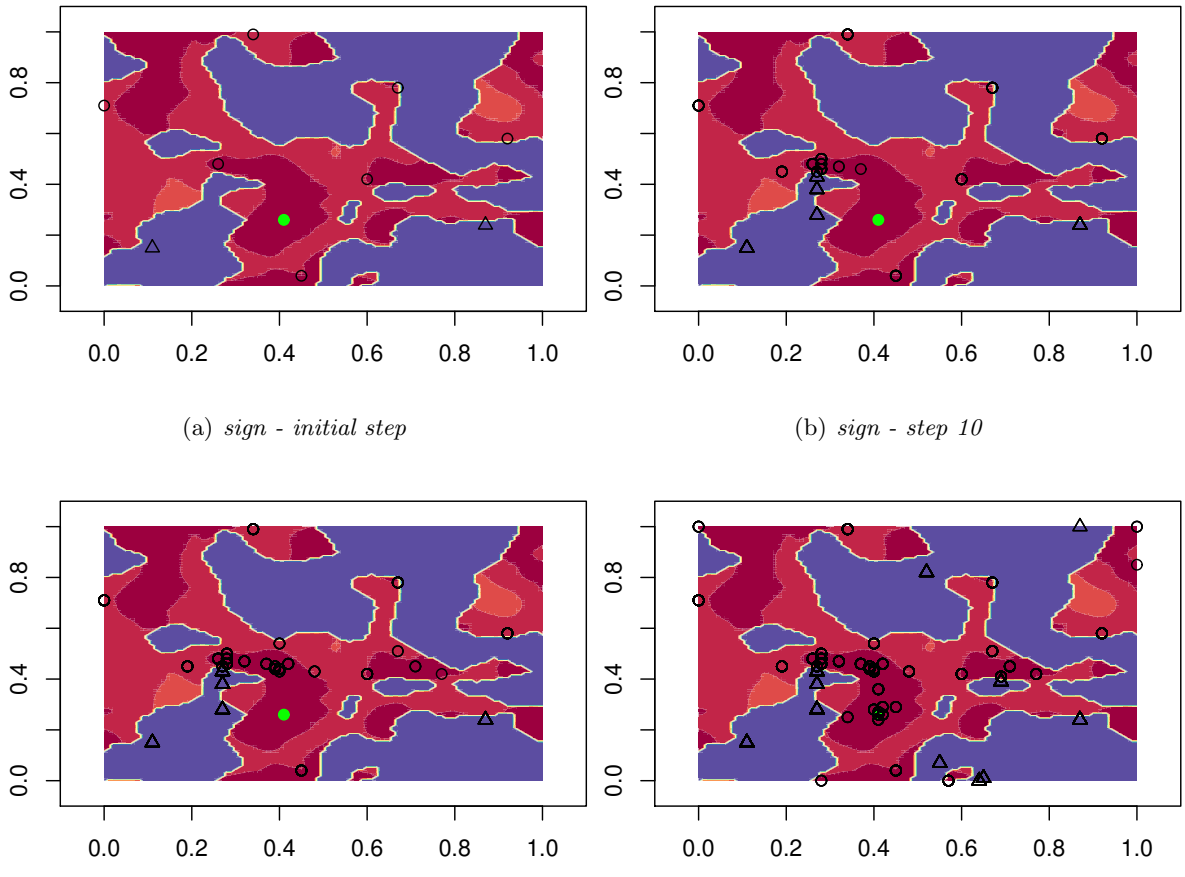
(a) Range parameters

	θ^Z	$\hat{\theta}^Z$	$\hat{\theta}^Z$	θ^Y	$\hat{\theta}^Y$	$\hat{\theta}^Y$
	true value	iteration 10	iteration 41	true value	iteration 10	iteration 41
Case 1	0.1	0.30 (0.35)	0.13 (0.10)	0.1	0.26 (0.48)	0.10 (0.03)
Case 2	0.1	0.23 (0.28)	0.18 (0.19)	0.3	0.98 (1.54)	0.36 (0.16)
Case 3	0.3	0.47 (0.37)	0.34 (0.23)	0.1	0.21 (0.45)	0.12 (0.16)
Case 4	0.3	0.44 (0.35)	0.35 (0.29)	0.3	0.48 (0.73)	0.43 (0.73)

(b) Trend and variance parameters

	$\hat{\sigma}_f^2$	$\hat{\sigma}_f^2$	$\hat{\mu}^Y$	$\hat{\mu}^Y$	$\hat{\sigma}_Y^2$	$\hat{\sigma}_Y^2$
	iteration 10	iteration 41	iteration 10	iteration 41	iteration 10	iteration 41
Case 1	8.91 (2.77)	10.00 (0.00)	-0.10 (0.55)	-0.12 (0.33)	0.70 (0.54)	0.77 (0.29)
Case 2	9.79 (0.84)	10.00 (0.00)	0.04 (0.73)	0.14 (0.53)	1.03 (0.77)	0.94 (0.73)
Case 3	9.44 (2.07)	10.00 (0.00)	-0.11 (0.37)	-0.06 (0.35)	0.62 (0.33)	0.81 (0.26)
Case 4	8.37 (3.28)	9.47 (2.29)	-0.21 (0.55)	-0.09 (0.56)	1.11 (1.81)	0.92 (0.84)

Table 3: Method *EFI GPC EP* at step 10 and 41: (a) Estimation of θ^Z and θ^Y, (b) Estimation of σ_f^2, μ^Y (true value is 0) and σ_Y^2 (true value is 1). Mean (standard deviation) over 20 simulations.



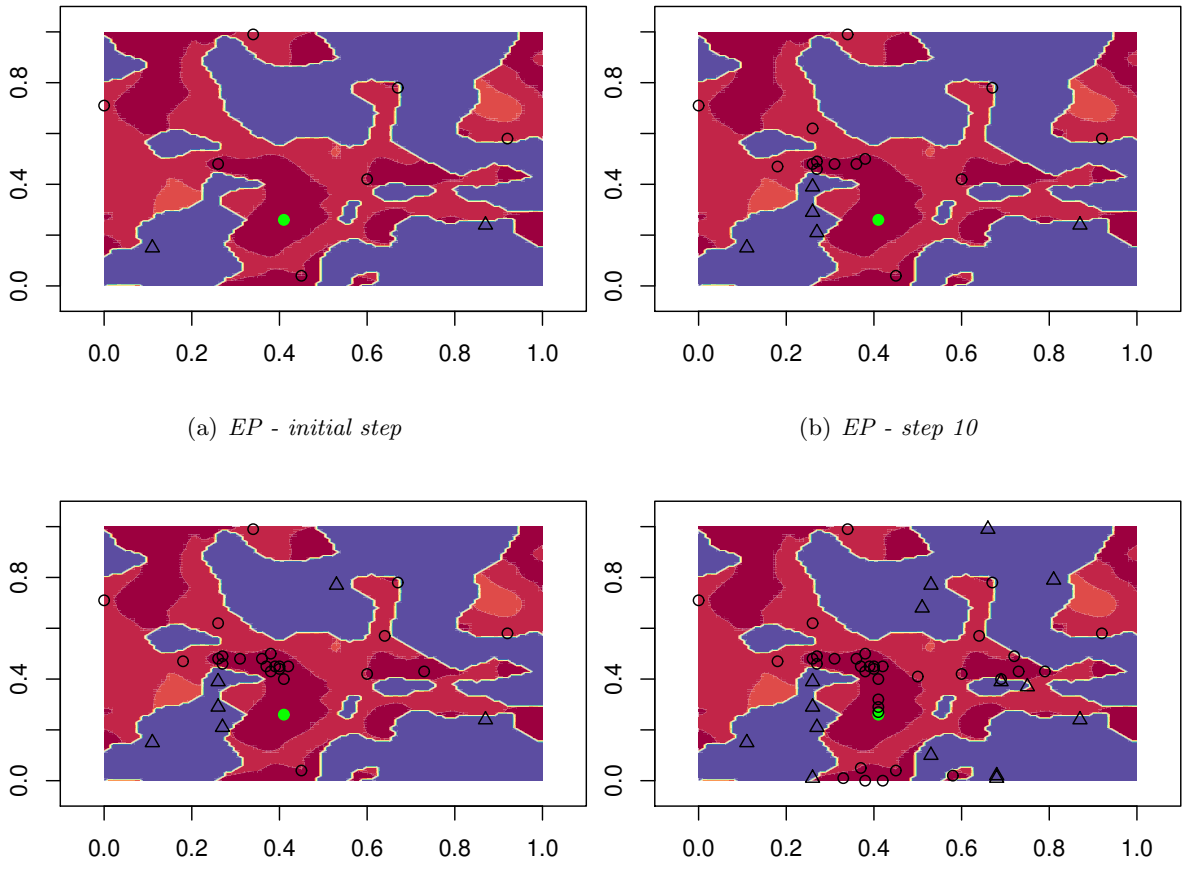
(a) *sign - initial step*

(b) *sign - step 10*

(c) *sign - step 20*

(d) *sign - final step*

Figure 6: Evolution of *EFI GCP sign* with iteration. The non admissible area is in blue. The best point is in green. NA points are plotted with triangles. Four situations are plotted : (a) initial step composed of 9 points, (b) after 10 added points, (c) after 20 added points, (d) final situation composed of 50 points.



(a) *EP - initial step*

(b) *EP - step 10*

(c) *EP - step 20*

(d) *EP - final step*

Figure 7: Evolution of *EFI GCP EP* with iteration. The non admissible area is in blue. The best point is in green. NA points are plotted with triangles. Four situations are plotted : (a) initial step composed of 9 points, (b) after 10 added points, (c) after 20 added points, (d) final situation composed of 50 points.

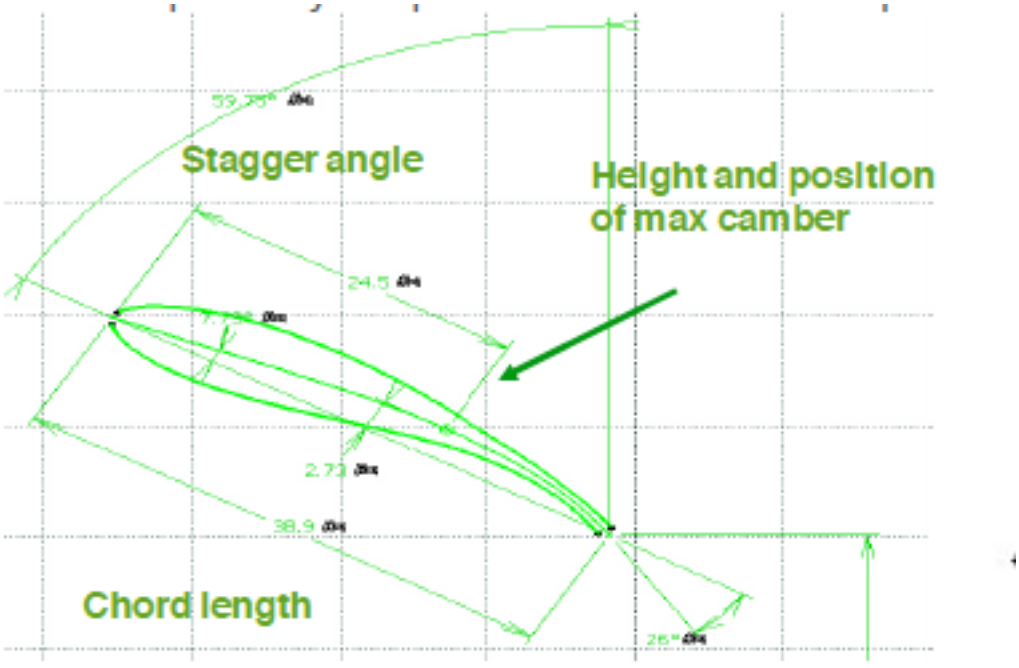


Figure 8: Section of a rotor blade for the turbomachinery of a fan system.

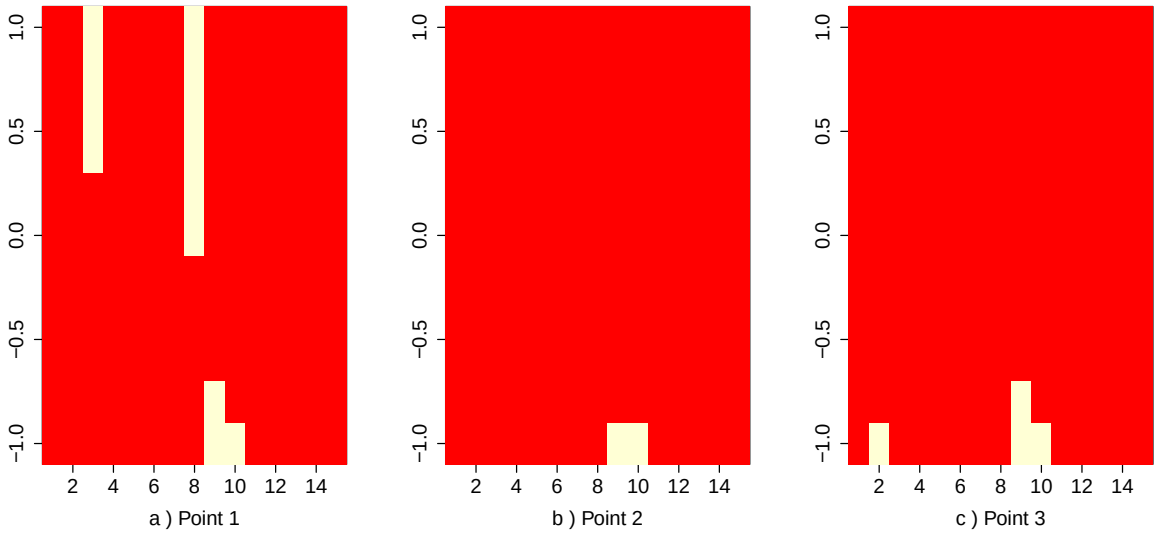


Figure 9: NA values around $Point1$, $Point2$ and $Point3$ (see Table 5). NA values are represented by yellow areas. The x-axis provides the index of the input variable to be changed. In the y-axis, -1 corresponds to the minimum for the corresponding input variable and 1 corresponds to the maximum.

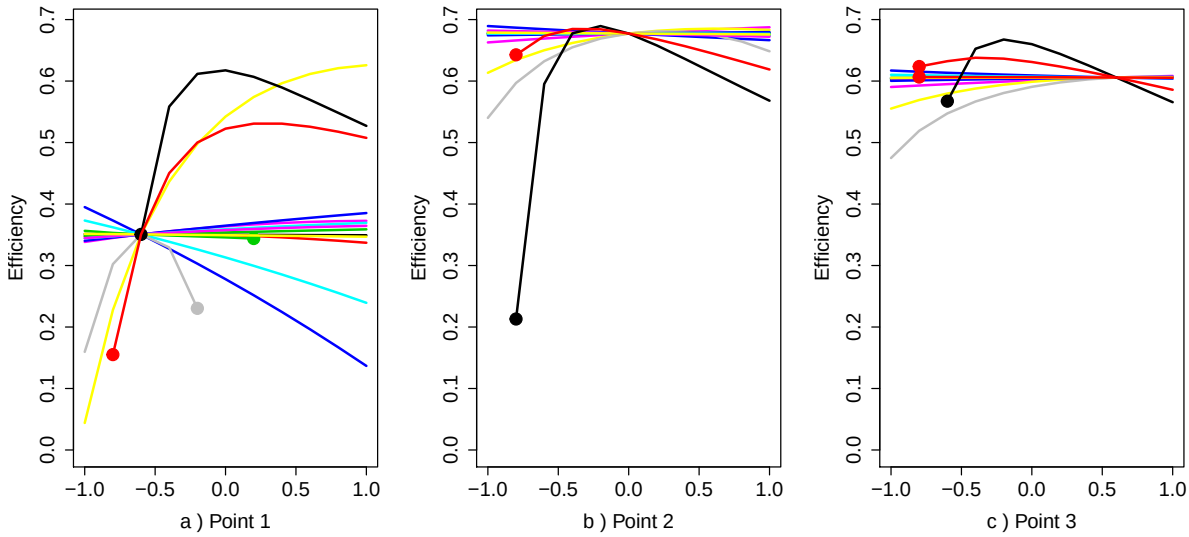
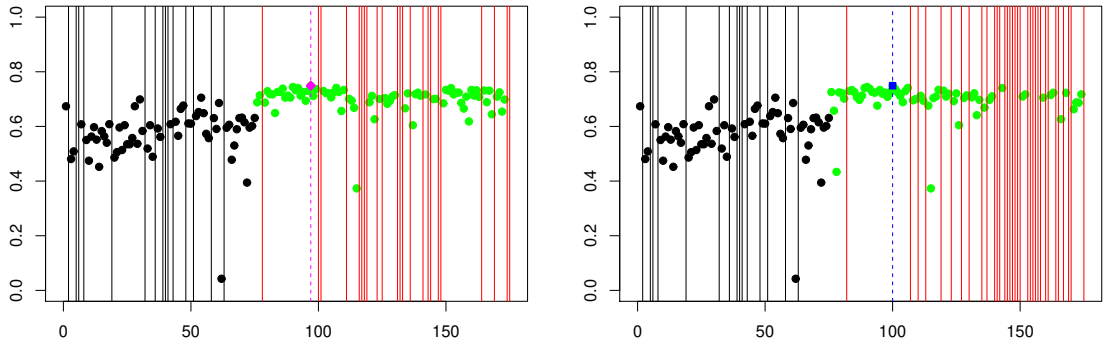


Figure 10: Evolution of the efficiency (output of the code) from min to max in each direction around $Point1$, $Point2$ and $Point3$. The colors indicate the different curves when varying the different input variables. A crash at a given value of x is indicated by the absence of the curve value. The bullets are used to highlight the beginning of the crash ranges for the input variables.



(a) *EFI PGC sign* (b) *EFI GPC EP*

Figure 11: Efficiency values for 175 geometric configurations. The first 75 points come from the initial DoE and are plotted in black. The 100 other points have been added by our optimization algorithms with failures (a) *EFI PGC sign* and (b) *EFI PGC EP*. Crashes are represented by vertical lines. On Figure (a) (resp. (b)), for *EFI PGC sign* (resp. *EFI PGC EP*), the best point is represented by a magenta diamond (resp. blue square) and is found at simulation number 97 (resp. 100).

A Proofs

Proof of Lemma 1. With ϕ^{Z_n} the p.d.f. of Z_n and with $s_n = (i_1, \dots, i_n)^\top$, we have

$$\begin{aligned}
\mathrm{P}_{\mathrm{nf}}(x) &= \mathbb{P}\left(Z(x) > 0 | \operatorname{sign}(Z_n) = s_n\right) \\
&= \frac{\mathbb{E}\left(\mathbf{1}_{Z(x)>0} \prod_{j=1}^n \mathbf{1}_{\operatorname{sign}(Z(x_j))=i_j}\right)}{\mathbb{P}\left(\operatorname{sign}(Z_n) = s_n\right)} \\
&= \frac{\int_{\mathbb{R}^n} \phi^{Z_n}(z_1, \dots, z_n) \left(\prod_{j=1}^n \mathbf{1}_{\operatorname{sign}(z_j)=i_j}\right) \mathbb{P}\left(Z(x) > 0 | Z(x_1) = z_1, \dots, Z(x_n) = z_n\right) dz_1 \dots dz_n}{\mathbb{P}\left(\operatorname{sign}(Z_n) = s_n\right)} \\
&= \int_{\mathbb{R}^n} \phi_{s_n}^{Z_n}(z_n) \bar{\Phi}\left(\frac{-m_n^Z(x, z_n)}{\sqrt{k_n^Z(x)}}\right) dz_n. \qquad (15)
\end{aligned}$$

The equation (15) is obtained by observing that

$$\frac{\phi^{Z_n}(z_1, \dots, z_n) \left(\prod_{j=1}^n \mathbf{1}_{\operatorname{sign}(z_j)=i_j}\right)}{\mathbb{P}\left(\operatorname{sign}(Z_n) = s_n\right)} = \frac{\phi^{Z_n}(z_1, \dots, z_n) \mathbf{1}_{\operatorname{sign}(Z_n)=s_n}}{\mathbb{P}\left(\operatorname{sign}(Z_n) = s_n\right)} = \phi_{s_n}^{Z_n}(z_n)$$

by definition of $\phi_{s_n}^{Z_n}(z_n)$, and that

$$\mathbb{P}\left(Z(x) > 0 | Z(x_1) = z_1, \dots, Z(x_n) = z_n\right) = \bar{\Phi}\left(\frac{-m_n^Z(x, z_n)}{\sqrt{k_n^Z(x)}}\right)$$

by Gaussian conditioning. □ □

Proof of Lemma 2. For any measurable function f, by the law of total expectation and using the independence of Y, $(W_1, \dots, W_n)$ and Z, we have

$$\begin{aligned}
&\mathbb{E}\left[f\left(I_1, \dots, I_n, V_1, \dots, V_n\right)\right] \\
&= \sum_{i_1, \dots, i_n \in \{-1,1\}} \mathbb{P}_{\mu^Z, \theta_Z}\left(I_1 = i_1, \dots, I_n = i_n\right) \\
&\quad \mathbb{E}\left[f\left(i_1, \dots, i_n, i_1 Y(x_1) + (1 - i_1) W_1, \dots, i_n Y(x_n) + (1 - i_n) W_n\right)\right] \\
&= \sum_{i_1, \dots, i_n \in \{-1,1\}} \mathbb{P}_{\mu^Z, \theta_Z}\left(I_1 = i_1, \dots, I_n = i_n\right) \\
&\qquad \int_{\mathbb{R}^n} dv \phi_{\mu^Y, \theta_Y, s_n}^Y\left(v_{s_n}\right) \left(\prod_{\substack{j=1,\dots,n \\ i_j=0}} \phi(v_j)\right) f\left(i_1, \dots, i_n, v_1, \dots, v_n\right).
\end{aligned}$$

This concludes the proof by definition of a p.d.f. □ □

Proof of Lemma 3. Consider measurable functions $f(Y)$, $g(Z)$, $h(I_1, \dots, I_n)$ and $\psi(I_1 Y(x_1), \dots,$

$I_nY(x_n))$. We have, by independence of Y and Z,

$$\begin{aligned}
&\mathbb{E}\left[f(Y)g(Z)h(I_1,\dots,I_n)\psi(I_1Y(x_1),\dots,I_nY(x_n))\right]\\
&=\sum_{i_1,\dots,i_n\in\{0,1\}}\mathbb{P}\left(I_1=i_1,\dots,I_n=i_n\right)\\
&\quad\mathbb{E}\left[f(Y)g(Z)h(i_1,\dots,i_n)\psi(i_1Y(x_1),\dots,i_nY(x_n))\middle|\, I_1=i_1,\dots,I_n=i_n\right]\\
&=\sum_{i_1,\dots,i_n\in\{0,1\}}\mathbb{P}\left(I_1=i_1,\dots,I_n=i_n\right)h(i_1,\dots,i_n)\\
&\quad\mathbb{E}\left[f(Y)\psi(i_1Y(x_1),\dots,i_nY(x_n))\right]\mathbb{E}\left[g(Z)\middle|\, I_1=i_1,\dots,I_n=i_n\right]\\
&=\sum_{i_1,\dots,i_n\in\{0,1\}}\mathbb{P}\left(I_1=i_1,\dots,I_n=i_n\right)h(i_1,\dots,i_n)\\
&\quad\mathbb{E}\left[\psi(i_1Y(x_1),\dots,i_nY(x_n))\mathbb{E}\left[f(Y)\middle|\, Y_{n,s_n}\right]\right]\mathbb{E}\left[g(Z)\middle|\, I_1=i_1,\dots,I_n=i_n\right].
\end{aligned}$$

The last display can be written as, with $\mathcal{L}_n$ the distribution of

$$I_1,\dots,I_n,I_1Y(x_1),\dots,I_nY(x_n),$$

$$\int_{\mathbb{R}^{2n}}d\mathcal{L}_n(i_1,\dots,i_n,i_1y_1,\dots,i_ny_n)h(i_1,\dots,i_n)\psi(i_1y_1,\dots,i_ny_n)$$
$$\mathbb{E}\left[f(Y)\middle|\, Y_{n,s_n}=Y_q\right]\mathbb{E}\left[g(Z)\middle|\, I_1=i_1,\dots,I_n=i_n\right],$$

where Y_q is as defined in the statement of the lemma. This concludes the proof. □ □

We now address the proof of Theorem 1. We let $(X_i)_{i\in\mathbb{N}}$ be the random observation points, such that X_i is obtained from (9) and (13) for $i\in\mathbb{N}$. The next lemma shows that conditioning on the random observation points and observed values works "as if" the observation points $X_1,\dots,X_n$ were non-random.

Lemma 4. *For any $x_1,\dots,x_k\in\mathcal{D}$, $i_1,...,i_k\in\{0,1\}^k$ and $i_1y_1,...,i_ky_k\in\mathbb{R}^k$, the conditional distribution of (Y,Z) given*

$$\begin{aligned}
&X_1=x_1,\operatorname{sign}(Z(X_1))=i_1,\operatorname{sign}(Z(X_1))Y(X_1)=i_1y_1,\dots,\\
&X_k=x_k,\operatorname{sign}(Z(X_k))=i_k,\operatorname{sign}(Z(X_k))Y(X_k)=i_ky_k
\end{aligned}$$

is the same as the conditional distribution of (Y,Z) given

$$\operatorname{sign}(Z(x_1))=i_1,\operatorname{sign}(Z(x_1))Y(x_1)=i_1y_1,\dots,\operatorname{sign}(Z(x_k))=i_k,\operatorname{sign}(Z(x_k))Y(x_k)=i_ky_k.$$

Proof. This lemma can be shown similarly as Proposition 2.6 in [2]. □ □

Proof of Theorem 1. For $k\in\mathbb{N}$, we remark that $\mathcal{F}_k$ is the sigma-algebra generated by

$$X_1,\operatorname{sign}(Z(X_1)),\operatorname{sign}(Z(X_1))Y(X_1),\dots,X_k,\operatorname{sign}(Z(X_k)),\operatorname{sign}(Z(X_k))Y(X_k).$$

We let $\mathbb{E}_k$, $\mathbb{P}_k$ and var_k denote the expectation, probability and variance conditionally on $\mathcal{F}_k$. For $x\in\mathcal{D}$, we let $\mathbb{E}_{k,x}$, $\mathbb{P}_{k,x}$ and $\operatorname{var}_{k,x}$ denote the expectation, probability and variance conditionally on

$$X_1,\operatorname{sign}(Z(X_1)),\operatorname{sign}(Z(X_1))Y(x_1),\dots,X_k,\operatorname{sign}(Z(X_k)),\operatorname{sign}(Z(X_k))Y(X_k),x,\operatorname{sign}(Z(x)),\operatorname{sign}(Z(x))Y(x).$$

We let $\sigma_k^2(u) = \mathrm{var}_k(Y(u))$, $m_k(u) = \mathbb{E}_k[Y(u)]$ and $P_k(u) = \mathbb{P}_k(Z(u) > 0)$. We also let $\sigma_{k,x}^2(u) = \mathrm{var}_{k,x}(Y(u))$, $m_{k,x}(u) = \mathbb{E}_{k,x}[Y(u)]$ and $P_{k,x}(u) = \mathbb{P}_{k,x}(Z(u) > 0)$.

With these notations, the observation points satisfy, for $k \in \mathbb{N}$,

$$X_{k+1} \in \mathrm{argmax}_{x\in\mathcal{D}}\mathbb{E}_{k,x}\left(\max_{\substack{u:P_{k,x}(u)=1\\ \sigma_{k,x}(u)=0}} Y(u) - M_k\right), \tag{16}$$

where

$$M_k = \max_{\substack{u:P_k(u)=1\\ \sigma_k(u)=0}} Y(u).$$

We first show that (16) can be defined as a stepwise uncertainty reduction (SUR) sequential design [2]. We have

$$\begin{aligned} X_{k+1} \in &\mathrm{argmax}_{x\in\mathcal{D}}\mathbb{E}_k\left(\max_{\substack{P_{k,x}(u)=1\\ \sigma_{k,x}(u)=0}} Y(u) - \max_{\substack{P_k(u)=1\\ \sigma_k(u)=0}} Y(u)\right) \\ \in &\mathrm{argmin}_{x\in\mathcal{D}}\mathbb{E}_k\left(\mathbb{E}_{k,x}\left(\max_{Z(u)>0} Y(u)\right) - \max_{\substack{P_{k,x}(u)=1\\ \sigma_{k,x}(u)=0}} Y(u)\right) \end{aligned} \tag{17}$$

since the second term in (17) does not depend on x and from the law of total expectation. We let

$$H_k = \mathbb{E}_k\left(\max_{Z(u)>0} Y(u) - \max_{\substack{P_k(u)=1\\ \sigma_k(u)=0}} Y(u)\right)$$

and

$$H_{k,x} = \mathbb{E}_{k,x}\left(\max_{Z(u)>0} Y(u) - \max_{\substack{P_{k,x}(u)=1\\ \sigma_{k,x}(u)=0}} Y(u)\right).$$

Then we have for $k \geq 1$

$$X_{k+1} \in \mathrm{argmin}_{x\in\mathcal{D}}\mathbb{E}_k\left(H_{k,x}\right).$$

We have, using the law of total expectation, and since $\mathbb{E}_{k,x}\left[\max_{P_{k,x}(u)=1,\sigma_{k,x}(u)=0} Y(u)\right] = \max_{P_{k,x}(u)=1,\sigma_{k,x}(u)=0} Y(u)$,

$$\begin{aligned} H_k - \mathbb{E}_k(H_{k+1}) =&\mathbb{E}_k\left(\max_{\substack{P_{k,X_{k+1}}(u)=1\\ \sigma_{k,X_{k+1}}(u)=0}} Y(u) - \max_{\substack{P_k(u)=1\\ \sigma_k(u)=0}} Y(u)\right) \\ \geq &0 \end{aligned}$$

since, for all $u, x \in \mathcal{D}$, $\sigma_{k,x}(u) \leq \sigma_k(u)$ and $P_k(u) = 1$ implies $P_{k,x}(u) = 1$. Hence $(H_k)_{k\in\mathbb{N}}$ is a supermartingale and of course $H_k \geq 0$ for all $k \in \mathbb{N}$. Also $|H_1| \leq 2\mathbb{E}_1\left[\max_{u\in\mathcal{D}}|Y(u)|\right]$ so that H_1 is bounded in L^1, since the mean value of the maximum of a continuous Gaussian process

on a compact set is finite. Hence, from Theorem 6.23 in [13], H_k converges a.s. as $k \to \infty$ to a finite random variable. Hence, as in the proof of Theorem 3.10 in [2], we have $H_k - \mathbb{E}_k(H_{k+1})$ goes to 0 a.s. as $k \to \infty$. Hence, by definition of X_{k+1} we obtain $\sup_{x\in\mathcal{D}}(H_k - \mathbb{E}_k(H_{k,x})) \to 0$ a.s. as $k \to \infty$. This yields, from the law of total expectation,

$$\begin{aligned} 0 \longleftarrow^{a.s.}_{n\to\infty} & \sup_{x\in\mathcal{D}} \mathbb{E}_k \left(\max_{\substack{P_{k,x}(u)=1 \\ \sigma_{k,x}(u)=0}} Y(u) - \max_{\substack{P_k(u)=1 \\ \sigma_k(u)=0}} Y(u) \right) \\ & \geq \sup_{x\in\mathcal{D}} \mathbb{E}_k \left[\mathbf{1}_{Z(x)>0} \left(Y(x) - M_k \right)^+ \right] \\ & \geq \sup_{x\in\mathcal{D}} P_k(x) \gamma(m_k(x) - M_k, \sigma_k(x)), \end{aligned} \tag{18}$$

from Lemma 3 and (11), where

$$\gamma(a,b) = a\Phi\left(\frac{a}{b}\right) + b\phi\left(\frac{a}{b}\right).$$

Recall from Section 3 in [33] that γ is continuous and satisfies $\gamma(a,b) > 0$ if $b > 0$ and $\gamma(a,b) \geq a$ if $a > 0$. We have for $k \in \mathbb{N}$, $0 \leq \sigma_k(u) \leq \max_{v\in\mathcal{D}} \sqrt{\mathrm{var}(Y(v))} < \infty$. Also, with the same proof as that of Proposition 2.9 in [2], we can show that the sequence of random functions $(m_k)_{k\in\mathbb{N}}$ converges a.s. uniformly on $\mathcal{D}$ to a continuous random function m_∞ on $\mathcal{D}$. Thus, from (18), by compacity, we have, a.s. as $k \to \infty$, $\sup_{x\in\mathcal{D}} P_k(x)(m_k(x) - M_k)^+ \to 0$ and $\sup_{x\in\mathcal{D}} P_k(x)\sigma_k(x) \to 0$. Hence, Part 1. is proved.

Let us address Part 2. For all $\tau \in \mathbb{N}$, consider fixed $v_1, \ldots, v_{N_\tau} \in \mathcal{D}$ for which $\max_{u\in\mathcal{D}} \min_{i=1,\ldots,N_\tau} ||u - v_i|| \leq 1/\tau$. Consider the event $E_\tau = \{\exists u \in \mathcal{D}; \min_{i\in\mathbb{N}} ||X_i - u|| \geq 2/\tau\}$. Then, E_τ implies the event $E_{v,\tau} = \cup_{j=1}^{N_\tau} E_{v,\tau,j}$ where $E_{v,\tau,j} = \{\min_{i\in\mathbb{N}} ||X_i - v_j|| \geq 1/\tau\}$. Let us now show that $\mathbb{P}(E_{v,\tau,j}) = 0$ for $j = 1, \ldots, N_\tau$. Assume that $E_{v,\tau,j} \cap \mathcal{C}$ holds, where $\mathcal{C}$ is the event in Part 1. of the theorem, with $\mathbb{P}(\mathcal{C}) = 1$. Since Y has the NEB property, we have $\liminf_{k\to\infty} \sigma_k(v_j) > 0$. Hence, $P_k(v_j) \to 0$ as $k \to \infty$ since $\mathcal{C}$ is assumed. We then have

$$\mathrm{var}(\mathbf{1}_{Z(v_j)>0} | \mathbf{1}_{Z(X_1)>0}, \ldots, \mathbf{1}_{Z(X_k)>0}) = P_k(v_j)(1 - P_k(v_j)) \to 0 \tag{19}$$

a.s. as $k \to \infty$. But we have

$$\begin{aligned} & \mathrm{var}(\mathbf{1}_{Z(v_j)>0} | \mathbf{1}_{Z(X_1)>0}, \ldots, \mathbf{1}_{Z(X_k)>0}) \\ & = \mathbb{E}\left[\left(\mathbf{1}_{Z(v_j)>0} - P_k(v_j) \right)^2 \middle| \mathbf{1}_{Z(X_1)>0}, \ldots, \mathbf{1}_{Z(X_k)>0} \right] \\ & = \mathbb{E}\left[\mathbb{E}\left[\left(\mathbf{1}_{Z(v_j)>0} - P_k(v_j) \right)^2 \middle| Z(x_1), \ldots, Z(x_k) \right] \middle| \mathbf{1}_{Z(X_1)>0}, \ldots, \mathbf{1}_{Z(X_k)>0} \right]. \end{aligned}$$

Since $P_k(v_j)$ is a function of $Z(x_1), \ldots, Z(x_n)$, we obtain

$$\begin{aligned} & \mathrm{var}(\mathbf{1}_{Z(v_j)>0} | \mathbf{1}_{Z(X_1)>0}, \ldots, \mathbf{1}_{Z(X_k)>0}) \\ & \geq \mathbb{E}\left[\mathrm{var}\left(\mathbf{1}_{Z(v_j)>0} | Z(x_1), \ldots, Z(x_k) \right) \middle| \mathbf{1}_{Z(X_1)>0}, \ldots, \mathbf{1}_{Z(X_k)>0} \right] \\ & = \mathbb{E}\left[g\left(\bar{\Phi}\left(\frac{-m_k(v_j)}{\sigma_k(v_j)} \right) \right) \middle| \mathbf{1}_{Z(X_1)>0}, \ldots, \mathbf{1}_{Z(X_k)>0} \right], \end{aligned}$$

with $g(t) = t(1-t)$ and with $\bar{\Phi}$ as in Lemma 1. We let $S = \sup_{k\in\mathbb{N}} |m_k(v_j)|$ and $s = \inf_{k\in\mathbb{N}} \sigma_k(v_j)$. Then, from the uniform convergence of m_k discussed below and from the NEB property of Z, we have $\mathbb{P}(E_{S,s}) = 1$ where $E_{S,s} = \{S < +\infty, s > 0\}$. Then, if $E_{v,\tau,j} \cap \mathcal{C} \cap E_{S,s}$ holds, we have

$$\begin{aligned}
&\mathrm{var}(\mathbf{1}_{Z(v_j)>0} | \mathbf{1}_{Z(X_1)>0}, \ldots, \mathbf{1}_{Z(X_k)>0}) \\
&\geq \mathbb{E}\left[g\left(\bar{\Phi}\left(\frac{S}{s}\right)\right) \middle| \mathbf{1}_{Z(X_1)>0}, \ldots, \mathbf{1}_{Z(X_k)>0} \right] \\
&\rightarrow^{a.s.}_{k\to\infty} \mathbb{E}\left[g\left(\bar{\Phi}\left(\frac{S}{s}\right)\right) \middle| \mathcal{F}_{Z,\infty} \right],
\end{aligned}$$

where $\mathcal{F}_{Z,\infty} = \sigma(\{\mathbf{1}_{Z(X_i)>0)}\}_{i\in\mathbb{N}})$ from Theorem 6.23 in [13]. Almost surely, conditionally on $\mathcal{F}_{Z,\infty}$ we have a.s. $S < \infty$ and $s > 0$. Hence we obtain that, on the event $E_{v,\tau,j} \cap \mathcal{A}$ with $\mathbb{P}(\mathcal{A}) = 1$, $\mathrm{var}(\mathbf{1}_{Z(v_j)>0} | \mathbf{1}_{Z(X_1)>0}, \ldots, \mathbf{1}_{Z(X_k)>0})$ does not go to zero. Hence, from (19), we have $\mathbb{P}(E_{v,\tau,j}) = 0$. This yields that $(X_i)_{i\in\mathbb{N}}$ is a.s. dense in $\mathcal{D}$. Hence, since $\{u; Z(u) > 0\}$ is an open set, we have $\max_{i;Z(X_i)>0} Y(X_i) \to \max_{Z(u)>0} Y(u)$ a.s. as $n \to \infty$. □ □

B Stochastic approximation of the likelihood gradient for Gaussian process based classification

In Appendixes B and C, for two matrices A and B of sizes $a \times d$ and $b \times d$, and for a function $h : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, let $h(A, B)$ be the $a \times b$ matrix $[h(a_i, b_j)]_{i=1,\ldots,a, j=1,\ldots,b}$, where a_i and b_j are the lines i and j of A and B.

Let $s_n = (i_1, \ldots, i_n) \in \{0, 1\}^n$ be fixed. Assume that the likelihood $\mathbb{P}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n)$ has been evaluated by $\hat{\mathbb{P}}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n)$. Assume also that realizations $z_n^{(1)}, \ldots, z_n^{(N)}$, approximately following the conditional distribution of Z_n given $\mathrm{sign}(Z_n) = s_n$, are available.

Let $\mathcal{Z} = \{z_n \in \mathbb{R}^n : \mathrm{sign}(z_n) = s_n\}$. Treating $x_1, \ldots, x_n$ as d-dimensional line vectors, let $\mathbf{x}$ be the matrix $(x_1^\top, \ldots, x_n^\top)^\top$. Then we have

$$\mathbb{P}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n) = \int_{\mathcal{Z}} \frac{1}{(2\pi)^{n/2}} \frac{1}{\sqrt{|k_\theta^Z(\mathbf{x},\mathbf{x})|}} e^{\frac{-1}{2}(z_n - \mu 1_n)^\top k_\theta^Z(\mathbf{x},\mathbf{x})^{-1}(z_n - \mu 1_n)} dz_n,$$

where $1_n = (1, \ldots, 1)^\top \in \mathbb{R}^n$ and $|.|$ denotes the determinant.

Derivating with respect to μ yields

$$\begin{aligned}
\frac{\partial}{\partial\mu}\mathbb{P}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n) = &\int_{\mathcal{Z}} \frac{1}{(2\pi)^{n/2}} \frac{1}{\sqrt{|k_\theta^Z(\mathbf{x},\mathbf{x})|}} e^{\frac{-1}{2}(z_n - \mu 1_n)^\top k_\theta^Z(\mathbf{x},\mathbf{x})^{-1}(z_n - \mu 1_n)} \\
&(1_n^\top k_\theta^Z(\mathbf{x},\mathbf{x})^{-1}(z_n - \mu 1_n)) dz_n \\
= &\mathbb{P}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n) \mathbb{E}_{\mu,\theta}\left(1_n^\top k_\theta^Z(\mathbf{x},\mathbf{x})^{-1}(Z_n - \mu 1_n) \middle| \mathrm{sign}(Z_n) = s_n) \right),
\end{aligned}$$

where $\mathbb{E}_{\mu,\theta}$ means that the conditional expectation is calculated under the assumption that Z has constant mean function μ and covariance function k_θ. Hence we have the stochastic approximation $\hat{\nabla}_\mu$ for $\partial/\partial\mu \mathbb{P}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n)$ given by

$$\hat{\nabla}_\mu = \hat{\mathbb{P}}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n) \frac{1}{N} \sum_{i=1}^{N} 1_n^\top k_\theta^Z(\mathbf{x},\mathbf{x})^{-1}(z_n^{(i)} - \mu 1_n).$$

Derivating with respect to θ_i for $i = 1, \ldots, p$ yields, with $\mathrm{adj}(M)$ the adjugate of a matrix M,

$$
\begin{aligned}
&\frac{\partial}{\partial \theta_i} \mathbb{P}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n) = \\
&\int_{\mathcal{Z}} \left(\frac{-1}{2} |k_\theta^Z(\mathbf{x},\mathbf{x})|^{-1} \mathrm{Tr}\left(\mathrm{adj}(k_\theta^Z(\mathbf{x},\mathbf{x})) \frac{\partial k_\theta^Z(\mathbf{x},\mathbf{x})}{\partial \theta_i} \right) \right. \\
&\left. + \frac{1}{2} (z_n - \mu 1_n)^\top \frac{\partial k_\theta^Z(\mathbf{x},\mathbf{x})}{\partial \theta_i} k_\theta^Z(\mathbf{x},\mathbf{x})^{-1} \frac{\partial k_\theta^Z(\mathbf{x},\mathbf{x})}{\partial \theta_i} (z_n - \mu 1_n) \right) \\
&\frac{1}{(2\pi)^{n/2}} \frac{1}{\sqrt{|k_\theta^Z(\mathbf{x},\mathbf{x})|}} e^{\frac{-1}{2}(z_n - \mu 1_n)^\top k_\theta^Z(\mathbf{x},\mathbf{x})^{-1}(z_n - \mu 1_n)} dz_n \\
&= \mathbb{P}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n) \\
&\mathbb{E}_{\mu,\theta} \left(\frac{-1}{2} |k_\theta^Z(\mathbf{x},\mathbf{x})|^{-1} \mathrm{Tr}\left(\mathrm{adj}(k_\theta^Z(\mathbf{x},\mathbf{x})) \frac{\partial k_\theta^Z(\mathbf{x},\mathbf{x})}{\partial \theta_i} \right) \right. \\
&\left. \left. + \frac{1}{2} (Z_n - \mu 1_n)^\top \frac{\partial k_\theta^Z(\mathbf{x},\mathbf{x})}{\partial \theta_i} k_\theta^Z(\mathbf{x},\mathbf{x})^{-1} \frac{\partial k_\theta^Z(\mathbf{x},\mathbf{x})}{\partial \theta_i} (Z_n - \mu 1_n) \right| \mathrm{sign}(Z_n) = s_n \right).
\end{aligned}
$$

Hence we have the stochastic approximation $\hat{\nabla}_{\theta_i}$ for $\partial/\partial\theta_i \mathbb{P}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n)$ given by

$$
\begin{aligned}
\hat{\nabla}_{\theta_i} = \hat{\mathbb{P}}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n) \frac{1}{N} \sum_{i=1}^{N} &\left(\frac{-1}{2} |k_\theta^Z(\mathbf{x},\mathbf{x})|^{-1} \mathrm{Tr}\left(\mathrm{adj}(k_\theta^Z(\mathbf{x},\mathbf{x})) \frac{\partial k_\theta^Z(\mathbf{x},\mathbf{x})}{\partial \theta_i} \right) \right. \\
&\left. + \frac{1}{2} (z_n^{(i)} - \mu 1_n)^\top \frac{\partial k_\theta^Z(\mathbf{x},\mathbf{x})}{\partial \theta_i} k_\theta^Z(\mathbf{x},\mathbf{x})^{-1} \frac{\partial k_\theta^Z(\mathbf{x},\mathbf{x})}{\partial \theta_i} (z_n^{(i)} - \mu 1_n) \right).
\end{aligned}
$$

Remark 1. *Several implementations of algorithms are available to obtain the realizations $z_n^{(1)}, \ldots, z_n^{(N)}$, as discussed after Algorithm 1. It may also be the case that some implementations provide both the estimate $\hat{\mathbb{P}}_{\mu,\theta}(\mathrm{sign}(Z_n) = s_n)$ and the realizations $z_n^{(1)}, \ldots, z_n^{(N)}$.*

C Expressions of conditional GP mean and covariance and acquisition function gradient

Let μ^Y and k^Y be the mean and covariance functions of Y. Treating $x_1, \ldots, x_q$ as d-dimensional line vectors, let $\mathbf{x}_q$ be the matrix extracted from $(x_1^\top, \ldots, x_n^\top)^\top$ by keeping only the lines which indices j satisfy $i_j = 1$.

We first recall the classical expressions of GP conditioning:

$$
\begin{aligned}
m_q^Y(x, Y_q) &= \mu^Y + k^Y(x, \mathbf{x}_q) \left(k^Y(\mathbf{x}_q, \mathbf{x}_q) \right)^{-1} \left(Y_q - \mu^Y \right) \\
k_q^Y(x, x') &= k^Y(x, x') - k^Y(x, \mathbf{x}_q) \left(k^Y(\mathbf{x}_q, \mathbf{x}_q) \right)^{-1} k^Y(\mathbf{x}_q, x').
\end{aligned}
$$

$\nabla_x m_q^Y(x, Y_q)$ and $\nabla_x k_q^Y(x, x)$ are straightforward provided that $\nabla_x k^Y(x, y)$ is available:

$$
\begin{aligned}
\nabla_x m_q^Y(x, Y_q) &= [\nabla_x k^Y(x, \mathbf{x}_q)] \left(k^Y(\mathbf{x}_q, \mathbf{x}_q) \right)^{-1} \left(Y_q - \mu^Y \right) \\
\nabla_x k_q^Y(x, x) &= \nabla_x k^Y(x, x) - 2 k^Y(x, \mathbf{x}_q) \left(k^Y(\mathbf{x}_q, \mathbf{x}_q) \right)^{-1} \nabla_x k^Y(\mathbf{x}_q, x).
\end{aligned}
$$

Then:

$$\nabla_x EI_q(x) = \Phi\left(\frac{m_q^Y(x,Y_q) - M_q}{\sqrt{k_q^Y(x,x)}}\right)\nabla_x m_q^Y(x,Y_q) + \phi\left(\frac{M_q - m_q^Y(x,Y_q)}{\sqrt{k_q^Y(x,x)}}\right)\frac{1}{2\sqrt{k_q^Y(x,x)}}\nabla_x k_q^Y(x,x).$$

For $\mathrm{P}_{\mathrm{nf}}(x)$, using the approximation of Algorithm 1, we have:

$$\widehat{\mathrm{P}_{\mathrm{nf}}}(x) = \frac{1}{N}\sum_{i=1}^{N}\bar{\Phi}\left(\frac{-m_n^Z(x,z_n^{(i)})}{\sqrt{k_n^Z(x,x)}}\right),$$

with $k_n^Z(x,x)$ as $k_n^Y(x,x)$ and

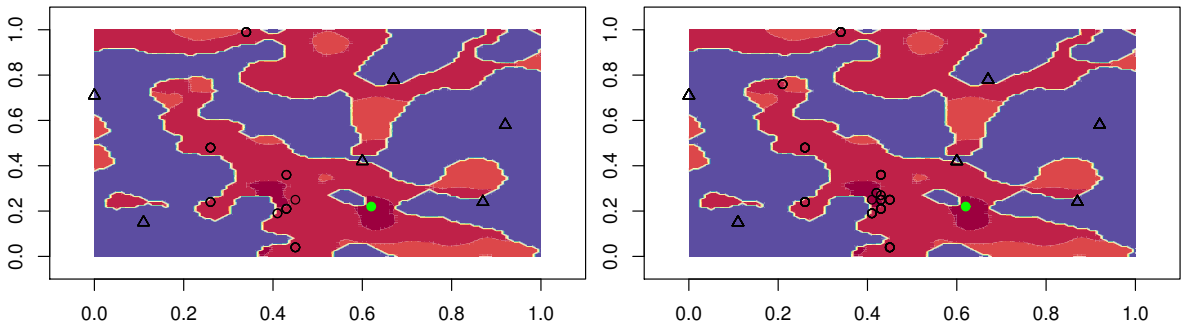
$$m_n^Z(x,z_n^{(i)}) = \mu^Z + k^Z(x,\mathbf{x})\left(k^Z(\mathbf{x},\mathbf{x})\right)^{-1}\left(z_n^{(i)} - \mu^Z\right).$$

Applying the standard differentiation rules delivers:

$$\nabla_x\widehat{\mathrm{P}_{\mathrm{nf}}}(x) \;=\; \frac{1}{N}\sum_{i=1}^{N}\phi\left(\frac{m_n^Z(x,z_n^{(i)})}{\sqrt{k_n^Z(x,x)}}\right)\left[\frac{1}{\sqrt{k_n^Z(x,x)}}\nabla m_n^Z(x,z_n^{(i)}) - \frac{m_n^Z(x,z_n^{(i)})}{2[k_n^Z(x,x)]^{3/2}}\nabla k_n^Z(x,x)\right].$$

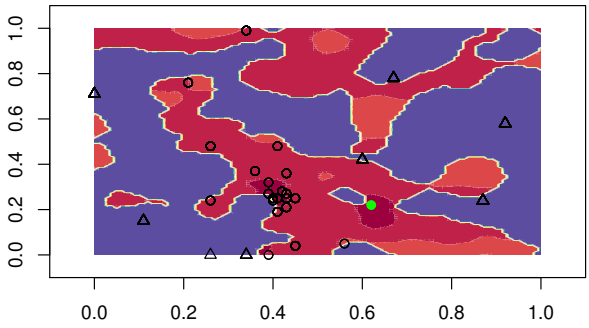
The acquisition function gradient can then be obtained using the product rule.

D 2D simulations - additional figures

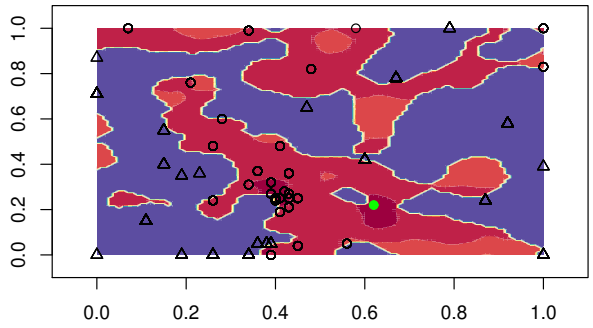


(a) *sign - initial step*

(b) *sign - step 10*

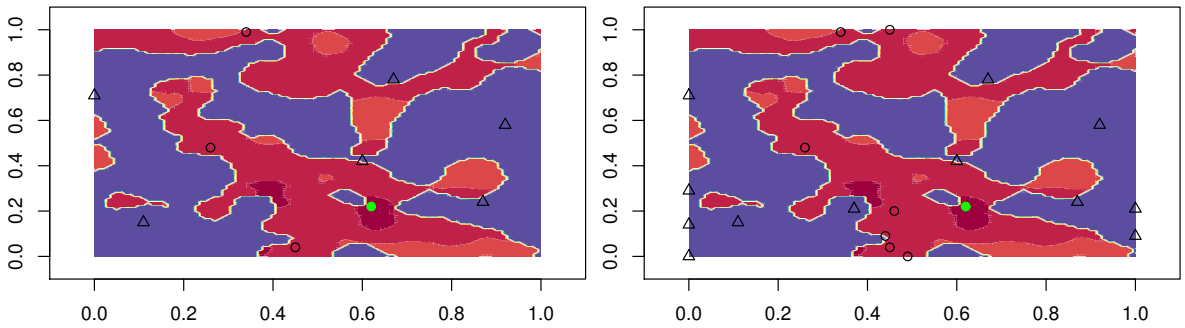


(c) *sign - step 20*



(d) *sign - final step*

Figure 12: Evolution of *EFI GCP sign* with iteration. The non admissible area is in blue. The best point is in green. NA points are plotted with triangles. Four situations are plotted : (a) initial step composed of 9 points, (b) after 10 added points, (c) after 20 added points, (d) final situation composed of 50 points.



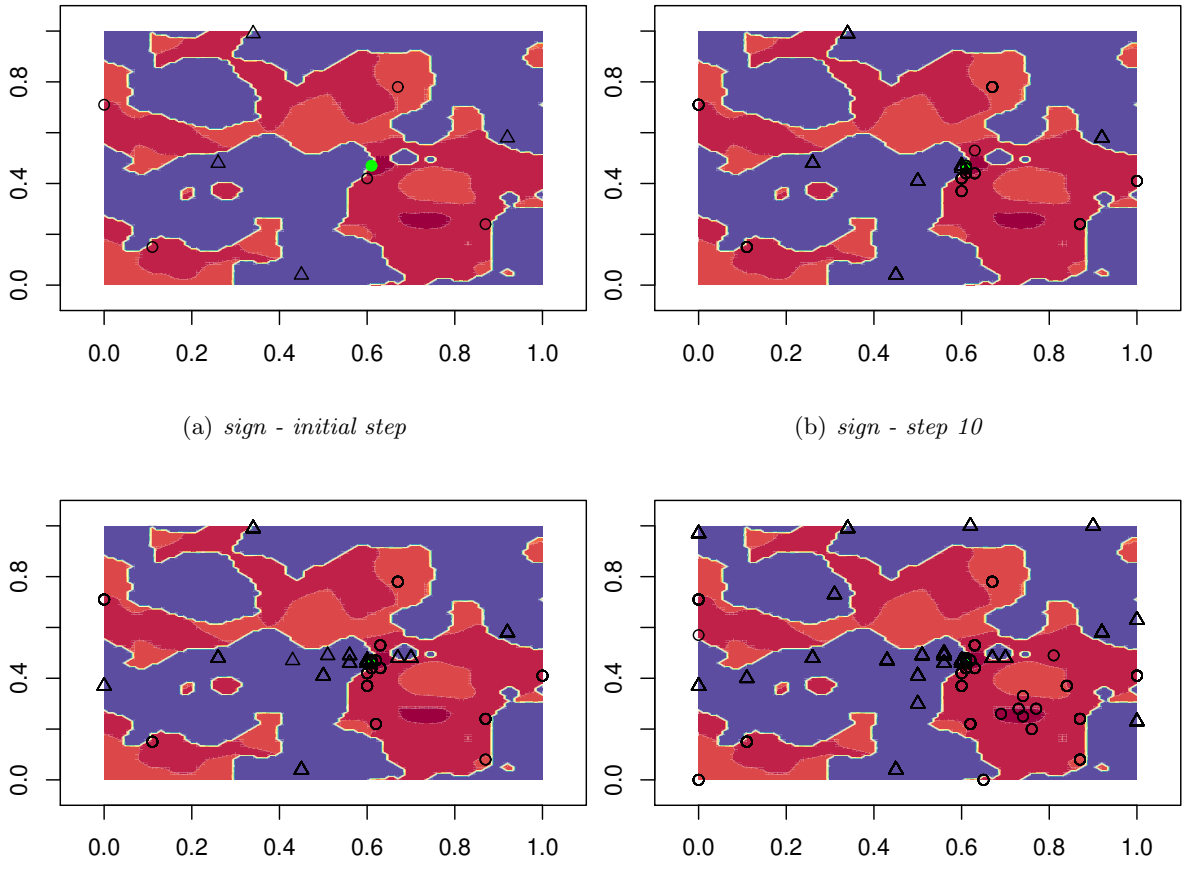
(a) *EP - initial step*

(b) *EP - step 10*

(c) *EP - step 20*

(d) *EP - final step*

Figure 13: Evolution of *EFI GCP EP* with iteration. The non admissible area is in blue. The best point is in green. NA points are plotted with triangles. Four situations are plotted : (a) initial step composed of 9 points, (b) after 10 added points, (c) after 20 added points, (d) final situation composed of 50 points.



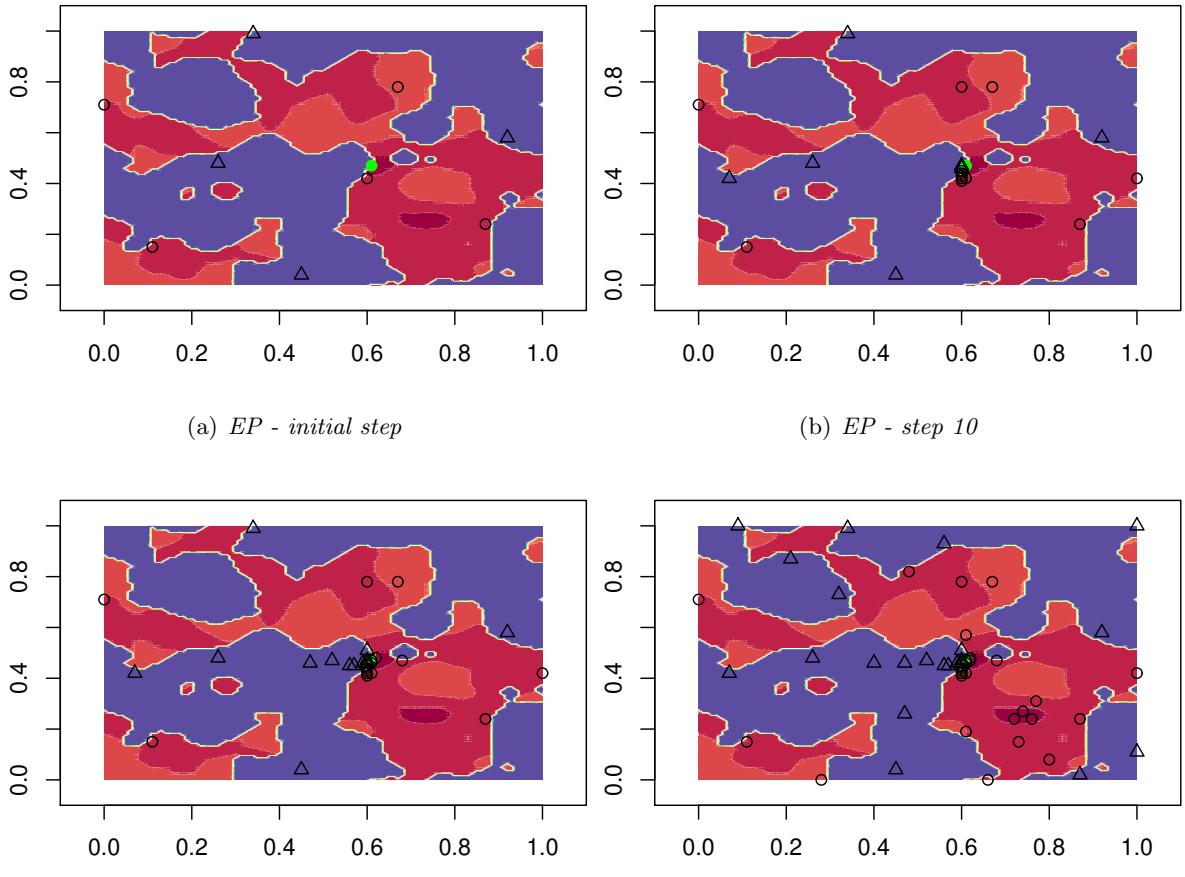
(a) *sign - initial step*

(b) *sign - step 10*

(c) *sign - step 20*

(d) *sign - final step*

Figure 14: Evolution of *EFI GCP sign* with iteration. The non admissible area is in blue. The best point is in green. NA points are plotted with triangles. Four situations are plotted : (a) initial step composed of 9 points, (b) after 10 added points, (c) after 20 added points, (d) final situation composed of 50 points.



(a) *EP - initial step*

(b) *EP - step 10*

(c) *EP - step 20*

(d) *EP - final step*

Figure 15: Evolution of *EFI GCP EP* with iteration. The non admissible area is in blue. The best point is in green. NA points are plotted with triangles. Four situations are plotted : (a) initial step composed of 9 points, (b) after 10 added points, (c) after 20 added points, (d) final situation composed of 50 points.

E Industrial case study - additional tables

	min	max
chord (m) 1	0.04138525	0.06752330
chord (m) 2	0.05650678	0.09219527
chord (m) 3	0.07162831	0.11686725
chord (m) 4	0.08674985	0.14153922
chord (m) 5	0.10187138	0.16621120
stagger (deg) 1	-53.35443064	-43.43365215
stagger (deg) 2	-62.79903182	-51.12211434
stagger (deg) 3	-69.27937009	-56.39749175
stagger (deg) 4	-73.91186646	-60.16861691
stagger (deg) 5	-77.35418265	-62.97086524
Hmax (% chord) 1	3.61896038	5.90461956
Hmax (% chord) 2	3.61896038	5.90461956
Hmax (% chord) 3	3.61896038	5.90461956
Hmax (% chord) 4	2.71422028	4.42846467
Hmax (% chord) 5	1.80948019	2.95230978

Table 4: Ranges for the 15 studied parameters

	Point 1	Point 2	Point 3
chord (m) 1	0.04661286	0.05445427	0.06229569
chord (m) 2	0.06364448	0.07435103	0.08505757
chord (m) 3	0.08067610	0.09424778	0.10781946
chord (m) 4	0.09770772	0.11414453	0.13058135
chord (m) 5	0.11473934	0.13404129	0.15334323
stagger (deg) 1	-51.37027494	-48.39404140	-45.41780785
stagger (deg) 2	-60.46364832	-56.96057308	-53.45749783
stagger (deg) 3	-66.70299442	-62.83843092	-58.97386742
stagger (deg) 4	-71.16321655	-67.04024168	-62.91726682
stagger (deg) 5	-74.47751917	-70.16252394	-65.84752872
Hmax (% chord) 1	4.07609221	4.76178997	5.44748772
Hmax (% chord) 2	4.07609221	4.76178997	5.44748772
Hmax (% chord) 3	4.07609221	4.76178997	5.44748772
Hmax (% chord) 4	3.05706916	3.57134248	4.08561579
Hmax (% chord) 5	2.03804611	2.38089498	2.72374386

Table 5: Coordinates of three reference points inside the hypercube.

References

[1] D. Azzimonti and D. Ginsbourger. Estimating orthant probabilities of high-dimensional Gaussian vectors with an application to set estimation. *Journal of Computational and Graphical Statistics*, 27(2):255–267, 2018.

[2] J. Bect, F. Bachoc, and D. Ginsbourger. A supermartingale approach to Gaussian process based sequential design of experiments. *Bernoulli, forthcoming*, 2016.

[3] Z. I. Botev. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(1):125–148, 2017.

[4] A. D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(Oct):2879–2904, 2011.

[5] M. A. Gelbart, J. Snoek, and R. P. Adams. Bayesian optimization with unknown constraints. In *UAI*, 2014.

[6] A. Genz. Numerical computation of multivariate normal probabilities. *Journal of computational and graphical statistics*, 1(2):141–149, 1992.

[7] D. Ginsbourger, R. Le Riche, and L. Carraro. Kriging is well-suited to parallelize optimization. In *Computational intelligence in expensive optimization problems*, pages 131–162. Springer, 2010.

[8] D. Ginsbourger, O. Roustant, and N. Durrande. On degeneracy and invariances of random fields paths with applications in Gaussian process modelling. *Journal of statistical planning and inference*, 170:117–128, 2016.

[9] R. Gramacy and H. Lee. Optimization under unknown constraints. *Bayesian Statistics*, 9, 2011.

[10] R. B. Gramacy, G. A. Gray, S. Le Digabel, H. K. Lee, P. Ranjan, G. Wells, and S. M. Wild. Modeling an augmented Lagrangian for blackbox constrained optimization. *Technometrics*, 58(1):1–11, 2016.

[11] J. M. Hernandez-Lobato, M. Gelbart, M. Hoffman, R. Adams, and Z. Ghahramani. Predictive entropy search for Bayesian optimization with unknown constraints. In *International Conference on Machine Learning*, pages 1699–1707, 2015.

[12] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[13] O. Kallenberg. *Foundations of Modern Probability. Second edition.* Springer-Verlag, 2002.

[14] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. P. Xing. Neural architecture search with Bayesian optimisation and optimal transport. In *Advances in Neural Information Processing Systems*, pages 2016–2025, 2018.

[15] A. Keane and P. Nair. *Computational approaches for aerospace design: the pursuit of excellence*. John Wiley & Sons, 2005.

[16] D. V. Lindberg and H. K. Lee. Optimization under constraints by applying an asymmetric entropy measure. *Journal of Computational and Graphical Statistics*, 24(2):379–393, 2015.

[17] A. F. López-Lopera, F. Bachoc, N. Durrande, and O. Roustant. Finite-dimensional Gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1224–1255, 2018.

[18] H. Maatouk and X. Bay. *A New Rejection Sampling Method for Truncated Multivariate Gaussian Random Variables Restricted to Convex Sets*, pages 521–530. Springer International Publishing, Cham, 2016.

[19] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[20] J. B. Mockus, V. Tiesis, and A. Žilinskas. The application of Bayesian methods for seeking the extremum. In L. C. W. Dixon and G. P. Szegö, editors, *Towards Global Optimization*, volume 2, pages 117–129, North Holland, New York, 1978.

[21] H. Nickisch and C. E. Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9(Oct):2035–2078, 2008.

[22] A. Pakman and L. Paninski. Exact Hamiltonian Monte Carlo for truncated multivariate Gaussians. *Journal of Computational and Graphical Statistics*, 23(2):518–542, 2014.

[23] V. Picheny. A stepwise uncertainty reduction approach to constrained global optimization. In *Artificial Intelligence and Statistics*, pages 787–795, 2014.

[24] V. Picheny, R. B. Gramacy, S. Wild, and S. Le Digabel. Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian. In *Advances in Neural Information Processing Systems*, pages 1435–1443, 2016.

[25] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, 2006.

[26] O. Roustant, D. Ginsbourger, and Y. Deville. DiceKriging, DiceOptim: Two Rpackages for the analysis of computer experiments by Kriging-based metamodeling and optimization. *Journal of statistical software*, 51(1):1–55, 2012.

[27] M. Sacher, R. Duvigneau, O. Le Maitre, M. Durand, E. Berrini, F. Hauville, and J.-A. Astolfi. A classification approach to efficient global optimization in presence of non-computable domains. *Structural and Multidisciplinary Optimization*, 58(4):1537–1557, 2018.

[28] M. J. Sasena, P. Papalambros, and P. Goovaerts. Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering optimization*, 34(3):263–278, 2002.

[29] M. Schonlau, W. J. Welch, and D. R. Jones. Global versus local search in constrained optimization of computer models. *Lecture Notes-Monograph Series*, pages 11–25, 1998.

[30] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[31] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1015–1022, 2010.

[32] J. Taylor and Y. Benjamini. RestrictedMVN: multivariate normal restricted by affine constraints. `https://cran.r-project.org/web/packages/restrictedMVN/index.html`, 2017. [Online; 02-Feb-2017].

[33] E. Vazquez and J. Bect. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and inference*, 140(11):3088–3095, 2010.

[34] E. Vazquez and J. Bect. Pointwise consistency of the kriging predictor with known mean and covariance functions. In *mODa 9–Advances in Model-Oriented Design and Analysis*, pages 221–228. Springer, 2010.

[35] J. Wu and P. Frazier. The parallel knowledge gradient method for batch Bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 3126–3134, 2016.