Probabilistic surrogate modeling by Gaussian process: A new estimation algorithm for more reliable prediction

Amandine Marrel[a,c], Bertrand Iooss[b,c,d]

[a] *CEA, DES, IRESNE, DER, Cadarache, 13108 Saint-Paul-Lez-Durance, France*
[b] *EDF R&D, 6 quai Watier, 78400 Chatou, France*
[c] *Institut de Mathématiques de Toulouse, Toulouse, France*
[d] *Corresponding Author*

Abstract

In reliability engineering studies, computer codes are increasingly used to model physical phenomena which, in many cases, can be very time-consuming to run. A widely accepted approach consists in approximating the CPU-time expensive computer model by a surrogate model. One of the most popular surrogate model is the Gaussian Process regression, as it provides, additionally to a prediction at an unobserved point, an uncertainty around this prediction (a predictive distribution). However, in practice, the quality of this metamodel depends on several choices, as the estimation and validation algorithms. The present work aims at proposing a new algorithm, based on constrained optimization multi-objective techniques, to estimate the Gaussian process hyperparameters in order to ensure robust and reliable predictive distribution of the Gaussian process. An intensive numerical benchmark on various analytical functions, with different input dimensions and learning sample sizes, shows its good performance in comparison with standard estimation algorithms. The new algorithm is also applied to a real test case modeling an aquatic ecosystem. It is compared with a recent robust and sophisticated Bayesian method; it proves to be as efficient while being less sensitive to the specification of the Gaussian process model.

Keywords: Computer experiments, Kriging, Machine learning, Optimization, Uncertainty, Validation criteria

1. Introduction

In several engineering fields, numerical models are used to simulate physical phenomena of industrial or environmental processes in order to improve its understandings, optimize some performances or perform a risk analysis related to safety criteria. In this context, one key issue is that the numerical model under study can be very time-consuming to run, which can drastically limit the number of possible simulations. To solve this cost issue, a widely accepted approach consists in approximating the CPU-time expensive computer model by a CPU-time inexpensive mathematical function called "surrogate model" (or "metamodel", term that is used in the following). Fit from a set of inputs and outputs of computer code simulations, these metamodels can come from any machine learning technique, as the simplest like the as polynomial regression, to more complex as the random forests and neural networks, see the reviews of Villa-Vialaneix et al. [1] in the environmental domain and Afshari et al. [2] in structural reliability.

In a first paper [3], companion of the present one, the value of Gaussian process (GP) regression (kriging) metamodel for emulating costly computational codes has been emphasized. Thanks to its great flexibility, this non-parametric regression tool has proved highly effective in modelling numerical simulators, in a wide range of application fields [4]. Moreover, it yields an analytical predictive distribution for the code output at each prediction point. Having a such probabilistic

metamodel, in the sense that it provides a predictive law for each new evaluation point, is of great added value, particularly for safety, reliability or risk assessment studies. It also enables the deployment of sophisticated GP-based approaches for active learning, robust optimization, reliability assessment, etc, as reviewed for example in Fuhg et al. [5], Moustapha et al. [6]. In this context, it is essential to guarantee confidence in the GP metamodel predictive law, and not just in its mean value (i.e. the GP predictor).

This confidence first calls for a reliable estimation of the GP metamodel and, more precisely, of its parameters, referred to as hyperparameters. Secondly, a rigorous validation of the entire GP predictive distribution is required, as extensively outlined by Demay et al. [7] and Petit et al. [8]. With regard to the first requirement, the companion paper [3] has reviewed recent works dealing with the estimation of GP hyperparameters, from a theoretical and empirical point of view. It appears that the usual methods, based mainly on likelihood maximization or on minimizing an error estimated by cross-validation, sometimes lead to poor-quality and provide non-robust estimates. In such cases, the validation step (the second requirement) must detect any unreliability in the GP predictive distribution. For this, it must be based on different criteria [7, 8] evaluating the GP's predictive capabilities, as well as the reliability of the associated prediction intervals. Furthermore, particular care and informed consideration must also be taken when jointly analyzing these criteria.

To ensure a more robust estimation of hyperparameters and a more reliable prediction distribution, some recent alternatives to standard estimation approaches have been proposed. Among them, Bayesian approaches are theoretically very attractive, as they offer a kind of likelihood regularization. However, their deployment in practice is limited by their complexity, not only in terms of computational cost especially in large dimension (large number of inputs), but also in terms of the expertise required to define so-called robust priors and analyze the resulting posteriors. Other alternative approaches based on ad-hoc corrections of the quantiles of the GP predictive distribution have also been proposed, as in Acharki et al. [9], in order to ensure reliable prediction intervals for a given level. However, these approaches do not necessarily seem relevant for obtaining a metamodel for multi-objective use.

The presented work aims at proposing a new algorithm to estimate the GP hyperparameters in order to ensure robust and reliable GP predictive distribution. This algorithm is based on a preliminary thorough analysis of estimation and validation criteria, both through the recent literature on the subject (reviewed in Marrel and Iooss [3]) and through an empirical exploration of their links on a large benchmark of analytic functions. On this basis, the new algorithm proposes to jointly maximize the likelihood of observations and the accuracy of GP prediction intervals, under the constraint of not degrading the GP predictivity.

The rest of the document is organized as follows. First, a brief reminder of the formalism of GP regression, estimation methods and validation criteria are first proposed. The interested reader is invited to refer to the companion paper [3] for more details on these subjects. Then, an empirical study of the connections and links between estimation and validation criteria is presented in Section 3. From this, the new estimation algorithm is proposed and detailed in Section 4. An intensive numerical benchmark is performed in Section 5 on various analytical functions from the literature on emulation and prediction of computer experiments. This benchmark makes it possible to evaluate, for different input dimensions and learning sample sizes, the additional value brought by the new algorithm (compared to standard algorithms). Finally, in Section 6, the algorithm is applied to a real test case modeling an aquatic ecosystem and more precisely a prey-predator chain. The last section gives some conclusions and prospects of this work.

2. Reminders on Gaussian process regression, estimation and validation

All the notations introduced in our companion paper [3] are retained, and those necessary for the present paper are recalled in what follows. The numerical model (computer code or simulator) is represented by the following input-output relationship:

$$\mathcal{M} : \left| \begin{array}{ccc} \mathcal{X} & \longrightarrow & \mathcal{Y} \\ \mathbf{X} & \longmapsto & Y = \mathcal{M}(\mathbf{X}) \end{array} \right. \tag{1}$$

where $\mathbf{X} = (X_1, \ldots, X_d)^\top$ are the d uncertain input parameters belonging to some measurable space $\mathcal{X} \subset \mathbb{R}^d$. In the context of given-data and data-driven GP metamodeling, we only have a n-size learning sample of inputs and associated outputs denoted by $(\mathbf{X_s}, \mathbf{Y}(\mathbf{X_s}))$ where $\mathbf{X_s} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$ with $\mathbf{x}^{(i)} = \left(x_1^{(i)}, \ldots, x_d^{(i)}\right)$ denotes the matrix of n-size sample locations, and $\mathbf{Y_s} = \{y^{(1)}, \ldots, y^{(n)}\}$ the corresponding outputs observations with $y^{(i)} = \mathcal{M}(\mathbf{x}^{(i)})$.

2.1. Gaussian process metamodel and associated parametric choices

In the GP regression, Y is considered as a realization of a Gaussian stochastic process:

$$Y(\cdot) \sim \mathcal{N}\left(m(\cdot), k(\cdot,\cdot)\right), \tag{2}$$

where $m(\cdot)$ is the mean function and $k(\cdot,\cdot)$ is the covariance kernel function. The predictive GP distribution is then the GP conditioned by the learning sample $(\mathbf{X_s}, \mathbf{Y_s})$

$$Y(\cdot) \mid \mathbf{Y_s} \sim \mathcal{N}\left(\hat{y}(\cdot), \hat{c}(\cdot,\cdot)\right), \tag{3}$$

with

$$\hat{y}\left(\mathbf{x}\right) = \mathbb{E}[Y(\mathbf{x})|\mathbf{Y_s}] = m(\mathbf{x}) + \mathbf{k}(\mathbf{x}, \mathbf{X_S})^T \mathbf{K}^{-1}\Big(\mathbf{Y_s} - \mathbf{m}(\mathbf{X_s})\Big), \tag{4}$$

and

$$\hat{c}(\mathbf{x}, \tilde{\mathbf{x}}) = \mathbb{COV}[Y(\mathbf{x}), Y(\tilde{\mathbf{x}})|\mathbf{Y_s}] = k(\mathbf{x}, \tilde{\mathbf{x}}) - \mathbf{k}(\mathbf{x}, \mathbf{X_s})^T \mathbf{K}^{-1} \mathbf{k}(\tilde{\mathbf{x}}, \mathbf{X_s}), \tag{5}$$

where $\mathbf{m}(\mathbf{X_s}) = (m(\mathbf{x}^{(i)}))_{1 \le i \le n} \in \mathbb{R}^n$, $\mathbf{K} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}))_{1 \le i,j \le n} \in \mathbb{R}^{n \times n}$ and $\mathbf{k}(\mathbf{x}, \mathbf{X_s}) = (k(\mathbf{x}, \mathbf{x}^{(i)}))_{1 \le i \le n} \in \mathbb{R}^n$.

The efficiency of GP metamodeling strongly depends on the specifications of its regression and covariance functions. In practice, this consists of making parametric choices for $m(\cdot)$ and $k(\mathbf{x}, \tilde{\mathbf{x}})$ among a panel of usual functions. First, in the absence of prior knowledge, a constant $m(\mathbf{x}) = \beta_0$ or a one-degree polynomial trend $m(\mathbf{x}) = \beta_0 + \sum_i \beta_i x_i$ is usually considered.

Secondly, concerning $k(\mathbf{x}, \tilde{\mathbf{x}})$, as the isotropy hypothesis is often too restrictive to emulate complicated models with inputs of very different kinds and influence, an anisotropic covariance is generally assumed. More precisely, the following tensorized covariance is considered:

$$k_{\sigma,\boldsymbol{\theta}}(\mathbf{x}, \tilde{\mathbf{x}}) = \sigma^2 \prod_{i=1}^{d} k_{\theta_i}(x_i - \tilde{x}_i). \tag{6}$$

where σ^2 is the variance parameter and $\theta_i \in \mathbb{R}^+$ correlation hyperparameter (also called correlation length or length-scale) associated to the ith input. As discussed in a our companion paper [3], the d 1-D covariance functions can be of different natures. But in practice, given the large number of inputs and without any prior knowledge, the usual practice is to use the same function for all variables. Moreover, the most popular choice for this correlation function is the Matérn function defined in one dimension by:

$$k_{\nu,\theta}(x, \tilde{x}) = \frac{2^{1-\nu}}{\Gamma(\nu)} \Big(\frac{\sqrt{2\nu}|x - \tilde{x}|}{\theta}\Big)^{\nu} K_\nu \Big(\frac{\sqrt{2\nu}|x - \tilde{x}|}{\theta}\Big), \tag{7}$$

where K_ν is a modified Bessel function of second kind with parameter $\nu \in \mathbb{R}^+$, and Γ is the Euler Gamma function. Parameter ν controls the smoothness of the GP (cf. Table 1 of Marrel and Iooss [3]). Usual choices in machine learning are $\nu = 1/2$ (exponential covariance), $\nu = 3/2$ and $\nu = 5/2$ (referred to as 3/2-Matérn and 5/2-Matérn covariances), and the Gaussian covariance which can be viewed as the the limiting case of Matérn function when $\nu \to \infty$. Finally, an additional nugget effect can be considered. It consists in assuming an additive white noise effect of variance σ_ϵ^2 which results in an additional term $\sigma_\epsilon^2 \delta_{\mathbf{x}\tilde{\mathbf{x}}}$ in the covariance function. In practice, it is characterized by the parameter $\lambda = \left(\frac{\sigma_\epsilon}{\sigma}\right)^2 \in \mathbb{R}^+$.

2.2. Main estimation methods

Assuming the aforementioned parametric choices, estimating the GP from the learning sample therefore boils down to estimating its parameters $(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta})$ and eventually λ if a nugget effect is considered. As described in our companion paper [3, Section 3], three main estimation procedures are used: minimization of the squared prediction error calculated by cross-validation (CV), maximization of likelihood (denoted MLE for maximum likelihood estimation), and Bayesian approach.

For the first method, and considering the *leave-one-out* (LOO) procedure (specific case of CV), the GP parameters are computed by minimizing the LOO-MSE:

$$\text{LOO-MSE}\left(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}\right) := \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{y}_{-i}\right)^2,$$

where $\hat{y}_{-i}$ denotes the mean of GP predictive distribution in $\mathbf{x}^{(i)}$ when $(\mathbf{x}^{(i)}, y^{(i)})$ is removed from the learning sample (this comes down to consider the GP conditioned by $\mathbf{Y}_{\mathbf{s},-\mathbf{i}}$). Note that the calculation of the LOO predictive distribution is greatly facilitated by the use of CV Dubrule [10]'s formulas, which provide analytical formulation for LOO mean $\hat{y}_{-i}$ and LOO variance $\hat{s}_{-i}^2 = \hat{c}_{-i}(\mathbf{x_i}, \mathbf{x_i})$.

The MLE method consists in identifying the values of the parameters that minimize the negative log-likelihood (NLL) of the learning sample. Provided that $\boldsymbol{\theta}$ is know, analytical solutions are available for $(\boldsymbol{\beta}, \sigma^2)$ and MLE boils down to minimize the profile NLL involving just $\boldsymbol{\theta}$. But, as for LOO-MSE, there is no closed-form expressions for the optimal values of parameters $\hat{\boldsymbol{\theta}}$ and numerical methods are thus required to estimate them.

The last approach, the Bayesian one is close to the MLE as it considers the likelihood of the learning sample but it also incorporates a prior distribution for the GP parameters. Bayesian estimation thus relies on maximizing the marginal posterior distribution of the parameters with regard to this prior distribution.

As discussed in Marrel and Iooss [3, Section 3.5], even if no consensus really emerges from theoretical analysis and empirical comparison of estimation methods, the recent empirical results of Petit [11] tend to argue that MLE is often preferable to its competitors. But MLE (as LOO-MSE) can be theoretically an ill-posed problem [12, 13] and flatness of ML (or LOO-based criteria) around the optimal parameter value can lead to poor performance of optimization algorithms. The full-Bayesian approach under the assumption of suitable priors, offer better properties but suffer from poor tractability. However, the recent RobustGaSP Bayesian method proposed by Gu et al. [13] and detailed in Marrel and Iooss [3, Section 5.2] with its specific robust priors and approximations, could overcome these limitations. Finally, considering a nugget effect jointly estimated with the other hyperparameters can facilitates the estimation by MLE by regularizing the likelihood function, improving the conditioning of the correlation matrix and numerical convergence of algorithms. It is of much less interest in the Bayesian approach, where the prior already plays a regularizing role and the additional estimation of the nugget can increase the identifiability problems.

Beyond the problem of obtaining a reliable estimate of the hyperparameters $\boldsymbol{\theta}$, it also appears that the choice of regularity of covariance function (ν in Matérn class) might be a key element to ensure GP's predictive capabilities.

2.3. Validation criteria

Whatever the parametric choices and the estimation method chosen, it is essential to have criteria for assessing the reliability of the obtained GP predictive distribution. These validation criteria can be used both to select the most appropriate covariance function and to check that the estimated hyperparameters lead to a reliable predictive distribution. They can be estimated from a test sample (different from the learning sample), or, as is often the case in our small-data application context (i.e. limited number of simulations), by cross-validation (or LOO, as a special case of cross validation). Hence, as described in Demay et al. [7], different validation criteria have been proposed to assess the whole GP conditional distribution:

- predictivity coefficient Q^2 to assess the accuracy of the GP predictor $\hat{y}(\mathbf{x})$:

$$Q^2 = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_{-i}\right)^2}{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \frac{1}{n}\sum_{i=1}^{n} y_i\right)^2} = 1 - \frac{\text{LOO-MSE}}{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \frac{1}{n}\sum_{i=1}^{n} y_i\right)^2}.$$

  The closer to one the Q^2, the better the accuracy of the metamodel predictor.

- predictive variance adequacy (PVA) factor to check if the conditional GP variance is of the right order of magnitude: $\text{PVA} = \left|\log\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_{-i})^2/\hat{s}_{-i}^2\right)\right|$. The smaller the PVA, the more reliable the prediction intervals.

- IAEα criterion to assess the reliability of GP prediction intervals: $\text{IAE}\alpha = \int_0^1 \left|\hat{\Delta}(\alpha) - \alpha\right| \mathrm{d}\alpha$ where $\hat{\Delta}(\alpha)$ is the empirical coverage function [14]. IAEα quantitatively summarizes the α-PI plot of Demay et al. [7]. It lies in $[0, 1]$ and the closer to zero the IAEα, better the PI in average.

To tackle the problems of estimating hyperparameters, and given the importance of having a reliable GP predictive distribution in the context of uncertainty quantification, we are convinced of the benefits of considering these criteria, rather reserved for validation, directly in the estimation process, in addition to NLL. And to find out how best to use and combine them, the links between them are firstly investigated, according to the values of the hyperparameters.

3. Empirical study of connections between likelihood and validation criteria

To design the new estimation algorithm, a preliminary study considers the links between some of the estimation and validation criteria, according to the values of the hyperparameters. Based on the review of recent works on GP estimation [3], the MLE is considered as the main estimation criterion. The interest is then to assess how this criterion can really control the quality of the predictive distribution (under the assumption that the optimization of this criterion has converged well). For this, the three following validation criteria (to be minimized) are considered: the MSE ($\propto 1 - Q^2$), the PVA and the IAEα.

A wide range of analytical functions, commonly used in metamodeling benchmarks, are then considered. For each function, random learning samples of different size n are simulated to build the conditional GP (Eqs 4,5). The domains of variation for the hyperparameters are discretized finely (on a grid for example if $d = 2$). And for each set of values of the hyperparameters (i.e. point of the grid if $d = 2$), the corresponding GP conditioned by the learning sample is built and different criteria are computed. More precisely, are computed:

- on the one hand, the negative log-likelihood (NLL) of the learning sample (criterion that one would seek to minimize in practice in the case of MLE),
- on the other hand, three validation criteria of the conditional GP computed on a large test sample of 1000 points: $(1 - Q^2)$, PVA and IAEα. Ideally, these three criteria should be as low as possible to ensure GP accuracy and predictivity (as detailed in Marrel and Iooss [3]).

In summary, for all the aforementioned criteria, the objective is to have the minimum value. Different sample sizes n are considered, well chosen w.r.t. the analytical functions tested in order to encompass cases of underlearning (n too small w.r.t. the complexity of the function) and overlearning (n much larger than necessary). For each test case, all the procedure is repeated 100 times for different i.i.d. random learning samples.

3.1. Illustration on an infinitely differentiable toy function

This section provides some results representative of the general trends observed over the 100 repetitions. First, the "re-scaled" Branin function [15], illustrated in Marrel and Iooss [3, Section 4.2 and Figure 2], is considered. The results obtained for the four quantities of interest aforementioned are given for the exponential, 3/2-Matérn, 5/2-Matérn and Gaussian covariances, by Figure 1 for $n = 30$. Similar plots are given for $n = 50$ by Figure A.11 in Appendix A. Note that NLL is plotted in logarithmic scale (hence the legend *logNLL*). First of all, the NLL and especially its optimal values are closely correlated to those of the predictivity coefficient Q^2, whatever the covariance considered. Minimizing the NLL for the estimation of the GP hyperparameters is therefore a good way to control the quality of the metamodel predictor and to ensure a good predictivity. On the other hand, the optimal points of the NLL do not correspond to the optimal points of the two other criteria controlling the quality of the whole predictive distribution, namely PVA and IAEα. Their behaviors w.r.t. the hyperparameters is relatively similar and more irregular than NLL and Q^2, with notably different zones with local minima. Moreover, the optimal points of PVA and IAEα do not correspond to those of NLL and Q^2. It would therefore be unwise to consider only PVA or IAEα in the optimization of hyperparameters, without first ensuring that the NLL is close to its optimal values.

This example is representative of the close links often observed between NLL and Q^2, and between PVA and IAEα. It also illustrates that these same two groups can be antagonistic, particularly in the case of a rough covariance (exponential for instance), while for smoother covariances, some interesting compromises between the two groups can be found. For instance, for Gaussian covariance, without degrading Q^2 too much (by staying on the same level line for it), the second group of criteria can be improved. This concordance between the criteria, which is observed all the more when the covariance is regular, makes sense here, given the high regularity of the Branin function. In the case of a well-specified covariance, the NLL optimization could thus be relevantly completed by an additional optimization of PVA or IAEα (under the constraint, for example, of not degrading the Q^2).

3.2. Illustration on a non-differentiable function

This section considers a more irregular function (non-differentiable and non-monotonic relationships), namely the G-Sobol function, defined in dimension d for uniform independent inputs on $[0, 1]$ by:

$$\mathcal{M}_{\text{Sobol}}(\mathbf{X}) = \prod_{k=1}^{d} g_k(X_k) \text{ where } g_k(X_k) = \frac{|4X_k - 2| + a_k}{1 + a_k} \text{ and } a_k \geq 0. \tag{8}$$

The same procedure as for Branin function is applied, with $d = 2$, $a_k = k$ for $k = 1, 2$ and $n = 50$. An illustration representative of the results obtained is given by Figure 2.

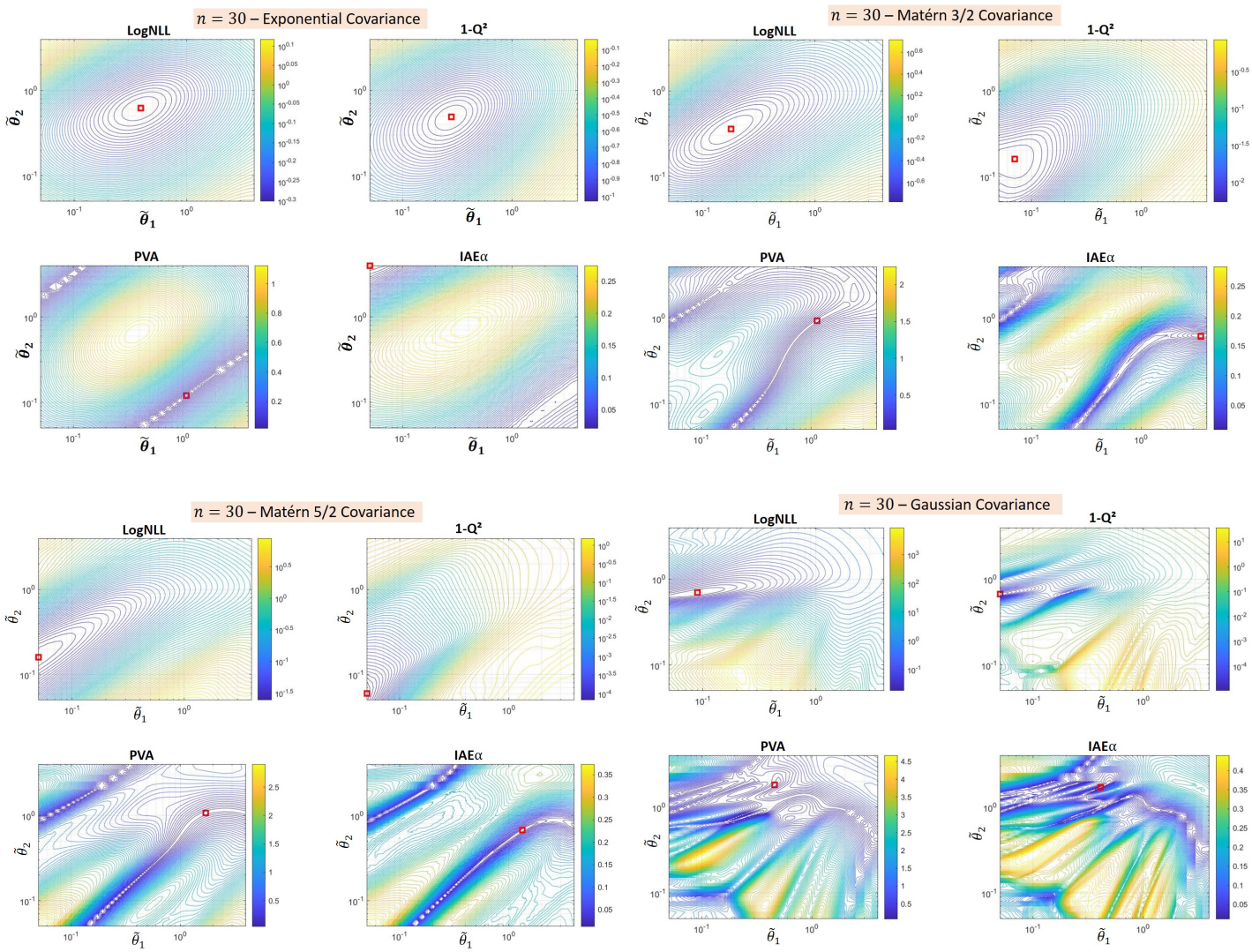

Figure 1: $\tilde{\mathcal{M}}_{\text{Branin}}$ Function – Comparison of NLL (computed on the learning sample) and validation criteria (computed on a test sample), for a GP built on a Monte Carlo learning sample of size $n = 30$. The optimal value for each quantity is indicated by a red square.

Similar results are found, with agreement between NLL and Q^2 and between PVA and IAEα, but here, there is more compatibility between the two groups in particular for covariance 3/2-Matérn. For more regular covariance, it's more difficult to optimize both groups simultaneously. However, for Gaussian covariance, the optimum area of PVA and IAEα corresponds to a local optimum zone of Q^2 and NLL, whose value is close to that of the global optimum. On this example, it could be interesting to shift slightly from the optimal point of the NLL towards more optimal values for the IAEα under the constraint of remaining in a region of Q^2 values close to the optimal Q^2.

3.3. Concluding remarks

We have carried out several other simulations on different analytical functions and for different dimensions, which are not presented here for the sake of brevity. In summary, similar observations often emerge with:

- a close behavior between NLL and Q^2 which pleads in favor of keeping NLL as the main estimation method to control the predictivity of the metamodel, which is consistent with the results obtained by Petit et al. [8];

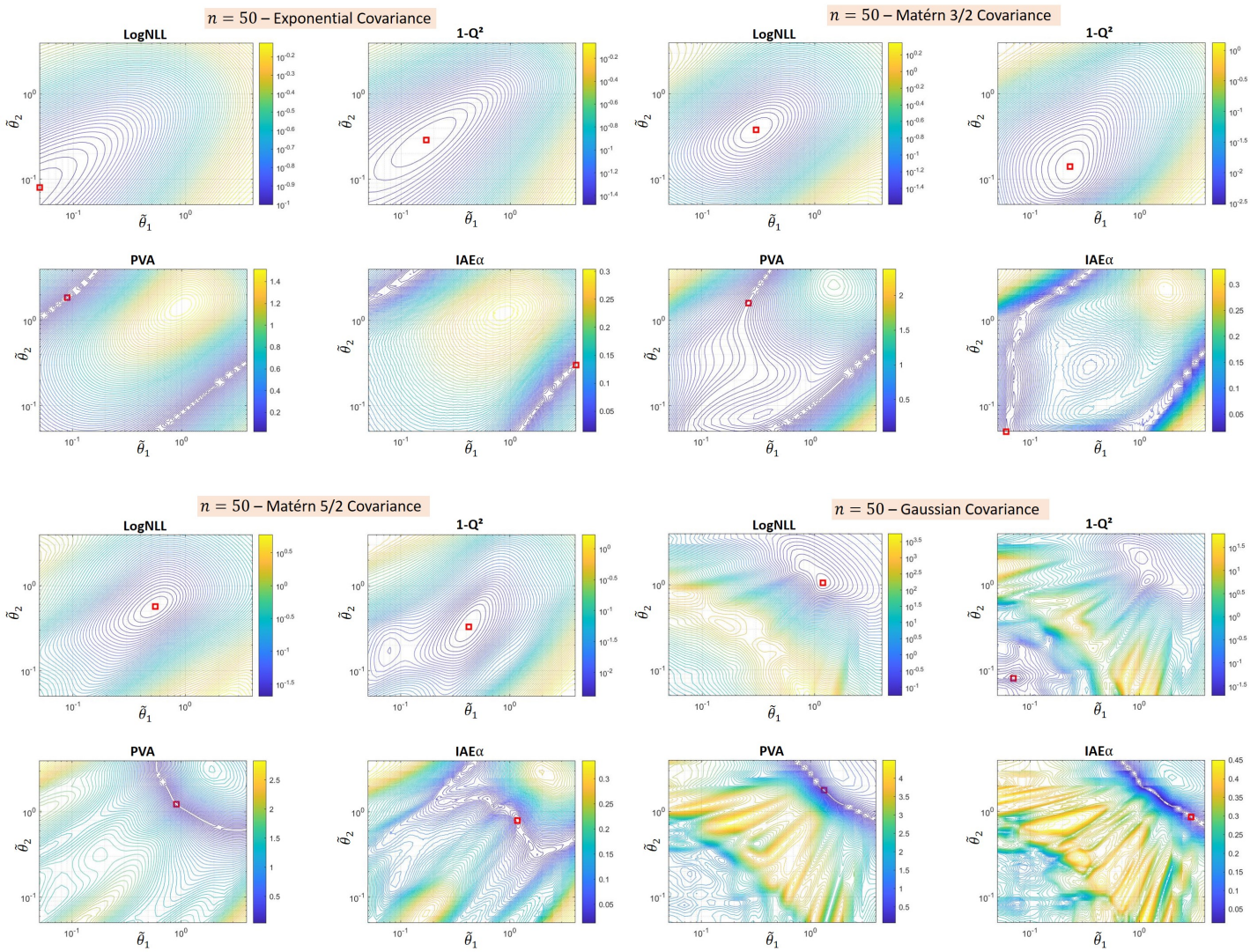

Figure 2: $\mathcal{M}_{\text{Sobol}}$ Function – Comparison of NLL (computed on the learning sample) and validation criteria (computed on a test sample), for a GP built on a Monte Carlo learning sample of size $n = 50$. The optimal value for each quantity is indicated by a red square.

- PVA and IAEα validation criteria often have a similar and more irregular behavior w.r.t. the hyperparameter values, and according to the covariance structure. They often present areas with local minima: some of them can be very compatible with rather optimal values of NLL and Q^2, while other minima correspond to very degraded values of NLL and Q^2. It would therefore be unwise to optimize only PVA or IAEα, without considering NLL or Q^2;

- In many examples, it appears possible to find in the neighborhood of the optimal NLL point, a better point w.r.t. PVA and IAEα. But this second optimization must be done while controlling the possible degradation of Q^2 value, which is the first guarantee of the good prediction capacity of the metamodel.

In addition, we also studied the shape of the Pareto fronts of the different pairs of criteria, considering their values computed either on a test basis (to have the "true" value), or by cross validation (LOO estimators) to be more representative of the results of a multi-objective optimization that would be carried out on the learning sample. In conclusion, it first emerges that the priority must be given to the minimization of the NLL to have accurate GP predictions. In addition,

Q^2 and IAEα appear as complementary criteria but without redundancy in the information they convey: the Q^2 validating the predictive mean of the GP and the IAEα validating the quality of the prediction intervals if the mean of the interval has been previously validated by the Q^2. The IAEα taken alone does not constitute a validation of the predictive distribution, it is only under condition of a good Q^2. All these considerations have guided the choice of the constrained multi-objective algorithm proposed in the next section.

4. New algorithm based on MLE and IAEα under constraint on Q^2

In line with the reparametrization considered by Gu et al. [13] and based on our expertise acquired on the optimization of GP hyperparameters, the inverse reparametrization is used in the rest of the document, namely $\tilde{\theta}_i = \dfrac{1}{\theta_i}$ for $i = 1, \ldots, d$. In the same way, the set of input variables are systematically re-scaled on $[0, 1]$ to allow an homogeneous interpretation of bounds, initial or estimated values of the hyperparameters. Hence, from the previous analysis, our new algorithm for optimizing the GP hyperparameters is based on the following three main steps:

- **Step 1: initial optimization based on NLL**. A first set of GP hyperparameters $\tilde{\boldsymbol{\theta}}^{init}_{MLE}$ is obtained by minimizing the NLL of the learning sample. For this, a multistart procedure is used combined with a BFGS algorithm, on the bounded domain of hyperparameters $\mathcal{D}_{\tilde{\boldsymbol{\theta}}}$. The LOO estimator of Q^2 corresponding to $\tilde{\boldsymbol{\theta}}^{init}_{MLE}$ is computed and denoted $\widehat{Q}^{2,init}_{LOO}$. Note that the BFGS algorithm is one of the most widely used quasi-Newton method for solving unconstrained nonlinear optimization problems.

- **Step 2: constrained multi-objective optimization based on NLL and LOO-IAEα**. A second optimization is carried out by considering two objectives: the NLL and IAEα computed by LOO on the learning sample (using Dubrule's formula [10]). For the constraint, a minimum threshold value for the Q^2 (computed by LOO), denoted c_{Q^2}, is considered. In practice, this value can be defined in absolute value $c_{Q^2} = \widehat{Q}^{2,init}_{LOO} - \gamma$ or relatively to $\widehat{Q}^{2,init}_{LOO}$ by $c_{Q^2} = \gamma \widehat{Q}^{2,init}_{LOO}$, with $\gamma \in [0, 1]$ for both cases. For instance, possible choices for the two cases are $\gamma = 0.05$ and $\gamma = 0.9$, respectively. The constraint is then expressed for any new candidate $\tilde{\boldsymbol{\theta}}^{new}$ by $Q^{2,new} \geq c_{Q^2}$ with $Q^{2,new}$ the Q^2 associated to the GP of hyperparameters $\tilde{\boldsymbol{\theta}}^{new}$. To perform this optimization, a constrained multi-objective evolutionary algorithm, namely the constrained NSGA-II (Non-dominated Sorting Genetic Algorithm), is used. Proposed by Deb et al. [16], it allows to reduce the computational complexity, and alleviate the non-elitism approach (of classic genetic algorithm) by using a modified mating and survival selection. Moreover, to ensure a search in the neighborhood of the point estimated by NLL at Step 1, the initial population is built upon $\tilde{\boldsymbol{\theta}}^{init}_{MLE}$. More precisely, half of the individuals are selected by randomly drawing a relative perturbation of +/- 10% of the values of $\tilde{\boldsymbol{\theta}}^{init}_{MLE}$. The other half is chosen via a space-filling design on $\mathcal{D}_{\tilde{\boldsymbol{\theta}}}$, namely a *maximin* Latin Hypercube Sampling (LHS), for more exploratory purposes [17, 18].

- **Step 3: Selection of the best individual in the Pareto front**. Once the set of solutions of the Pareto front has been provided by NSGA-II algorithm, a classification algorithm is first used: a k-means algorithm is carried out where the optimal number of clusters is determined by the Elbow method [19, 20]. A fundamental step for any unsupervised algorithm is to determine the optimal number of clusters into which the data may be clustered. The Elbow method is one of the most popular methods to determine this optimal value of k. Silhouette method or Gap statistic-based approach could also be used [21]. Then, several best candidates

could be chosen: the best individual in cluster 1 (cluster with minimal value of the 1st objective NLL) according to the 2nd objective (IAEα), the centroid of cluster 1, and the best individual for IAEα criterion among all the points of the Pareto Front. The first two approaches are recommended for more conservative Q^2 results.

In all the numerical experiments presented in the following, the algorithmic parameters of the different steps are set as follows:

- Step 1: $\mathcal{D}_{\tilde{\boldsymbol{\theta}}} = [0.01; 10]^d$ and a BFGS algorithm is used with a multistart approach of 10 points chosen by *maximin* LHS on $\mathcal{D}_{\tilde{\boldsymbol{\theta}}}$. An additional central initial point is considered with all the hyperparameters set at $\tilde{\theta} = 2$.

- Step 2: for the NSGA-II algorithm the crossover and mutation percentages are both set at 50%, the mutation rate and step size are respectively $\mu_{\text{NSGA}} = 0.02$ and $\sigma_{\text{NSGA}} = 0.1\Delta_{\tilde{\theta}}$ with $\Delta_{\tilde{\theta}}$ the width of the definition interval of $\tilde{\theta}$ (same values are considered for all the $(\tilde{\theta}_i)_{1\leq i\leq d}$). Moreover, the total size of the initial population of NSGA-II algorithm is 80 (40 from random perturbation of $\tilde{\boldsymbol{\theta}}_{MLE}^{init}$ an 40 from *maximin* LHS design on $\mathcal{D}_{\tilde{\boldsymbol{\theta}}}$). The maximum number of iterations is set at 50. The rest of NSGA-II parameters are set at their default values (see Table B.2 in Appendix B). Concerning the constraint, it is here defined as $c_{Q^2} = \widehat{Q}_{LOO}^{2,init} - 0.05$. This parameter is fixed just for the automation of the procedure. In practice, this choice is really to be made by the user w.r.t. the initial value $\widehat{Q}_{LOO}^{2,init}$ and the acceptable degradation of Q^2.

- Step 3: usual default values of the elbow algorithm are used. More precisely, the maximum number of clusters to try is fixed at $\lceil\sqrt{m}\rceil$ with m the number of individuals in the Pareto front. The optimal number of clusters is selected so that a fraction equal to 0.95% of the variance is explained and the k-means procedure is repeated three times, taking the best result at the end.

For all the numerical tests in the following, all the parameters of the optimization algorithm are fixed, independently of the model $\mathcal{M}$, the dimension d and the sample size n. Thus, the results obtained will be less dependent on these choices. It aims to illustrate that a use with standard values is quite possible, without real optimization of the algorithm parameters. It is obvious that in practice a more expert choice would also be possible and could further improve the results.

5. Numerical benchmark on analytical functions

This section focuses on different analytical functions usually used in the literature on emulation of computer experiments, from dimension $d = 3$ to $d = 20$. All the functions have been redefined for uniform independent inputs on $[0, 1]$ and the learning sample sizes n have been chosen to obtain from medium to high GP predictivities: Q^2 around 0.7 and $Q^2 \geq 0.9$, respectively. Hence, a large number of functions have been tested, and a representative sample is given below for:

- Friedman function in dimension $d = 5$ proposed by Friedman [22]:

$$\mathcal{M}_{\text{Fried-d5}}(\mathbf{X}) = 10\sin(\pi X_1 X_2) + 20\,(X_3 - 0.5)^2 + 10X_4 + 5X_5; \tag{9}$$

- Dette & Pepelyshev function in dimension $d = 8$ defined by Dette and Pepelyshev [23]:

$$\mathcal{M}_{Dette-d8}(\mathbf{X}) = 4\left(X_1 - 2 + 8X_2 - 8X_2^2\right)^2 + (3 - 4X_2)^2 + 16\sqrt{X_3 + 1}\,(2X_3 - 1)^2 + \sum_{i=4}^{8} i \ln\left(1 + \sum_{j=3}^{i} X_j\right); \tag{10}$$

- a function in dimension $d = 20$ proposed by Marrel et al. [24] and inspired from the Friedman function:

$$\mathcal{M}_{Marrel-d20}(\mathbf{X}) = a_1 \sin\left(6\pi\ X_1^{5/2}(X_2 - 0.5)\right) + a_2(X_3 - 0.5)^2 + a_3 X_4 + a_4 X_5 + r_{X6,\ldots,X_{15}} \quad (11)$$

  where $r_{X_6,\ldots,X_{15}} = \frac{a_5}{\sqrt{(\sum_{i=6\ldots15} i^2)}} \sum_{i=6\ldots15} \sqrt{12} i (X_i - 0.5)$. More details and illustration of main effects of $\mathcal{M}_{Marrel-d20}$ are given in Appendix C.1.

We also consider the G-Sobol (introduced in the companion paper [3]), Ishigami and Becker's functions for dimension $d = 8$, $d = 3$ and $d = 9$, respectively. Details on the two last functions and associated results are given in Appendix C, for the sake of brevity.

5.1. Results without additional nugget effect

No nugget effect is considered at first and a constant mean $m(\mathbf{x}) = \beta_0$ is assumed for all the GP. The learning samples are randomly generated according to space-filling LHS and, for each configuration, all the procedure is repeated 100 times by generating independent LHS designs. Hence, for each test case, three methods for estimating the GP hyperparameters are compared:

- a simple BLFG algorithm, as a very rudimentary estimate (not to be used in practice).
- A "multistart" BFGS (denoted "*Multi-BFGS*"), as a commonly used reference method offering in general a good compromise between simplicity (theoretical and in its practical use), efficiency and execution time.
- Our constrained multi-objective algorithm denoted "*C-NSGA-II-BestC1*": the best individual (in the Pareto front) according to the IAEα is chosen in the cluster denoted C_1 with the minimal mean value of the NLL (see the description of Step 3 in Section 4).

For each configuration, validation criteria Q^2, PVA and IAEα, are computed on a random test sample of 10^4 simulations (independent from the learning sample). The results obtained are given in boxplot form by Figures 3 to 5, for the different functions ordered by increasing input dimension. For each function, different covariance functions are presented by order of increasing regularity, to facilitate interpretation. Thus, several observations can be made from these initial tests.

- ▶ Concerning the GP predictivity (Q^2), for all functions and whatever the covariance considered, the Multi-BFGS and C-NSGA-II-BestC1 algorithms are largely better in terms of predictivity than a simple BFGS which presents significantly lower and more variable Q^2 (especially when the sample size n is small). A bad estimation of the hyperparameters can thus be very penalizing for the quality of the metamodel. Then, C-NSGA-II-BestC1 does not yield a degradation of Q^2 w.r.t. Multi-BFGS. This is obviously explained by the candidate selected in the Pareto front in Step 3 of the algorithm (see 4): choosing the best candidate in C_1 being the most conservative and safe from the point of view of Q^2. However, even with a threshold of 0.05 (on the Q^2 computed by LOO on the learning sample), the observed loss of Q^2 is often much lower or even almost zero. Note that other possibilities for the choice of best candidate in Pareto front have also been tested: the centroid of cluster C_1 gives equivalent results (sometimes a little worse) while the choice of the optimal point in terms of IAEα on the whole Pareto Front causes too much degradation of the Q^2 on average.
- ▶ Regarding now the accuracy of the GP predictive distribution, C-NSGA-II-BestC1 makes it possible to improve PVA and IAEα criteria, almost all the time and sometimes in a very significant way. The improvement brought by the algorithm is even more interesting in the following cases:

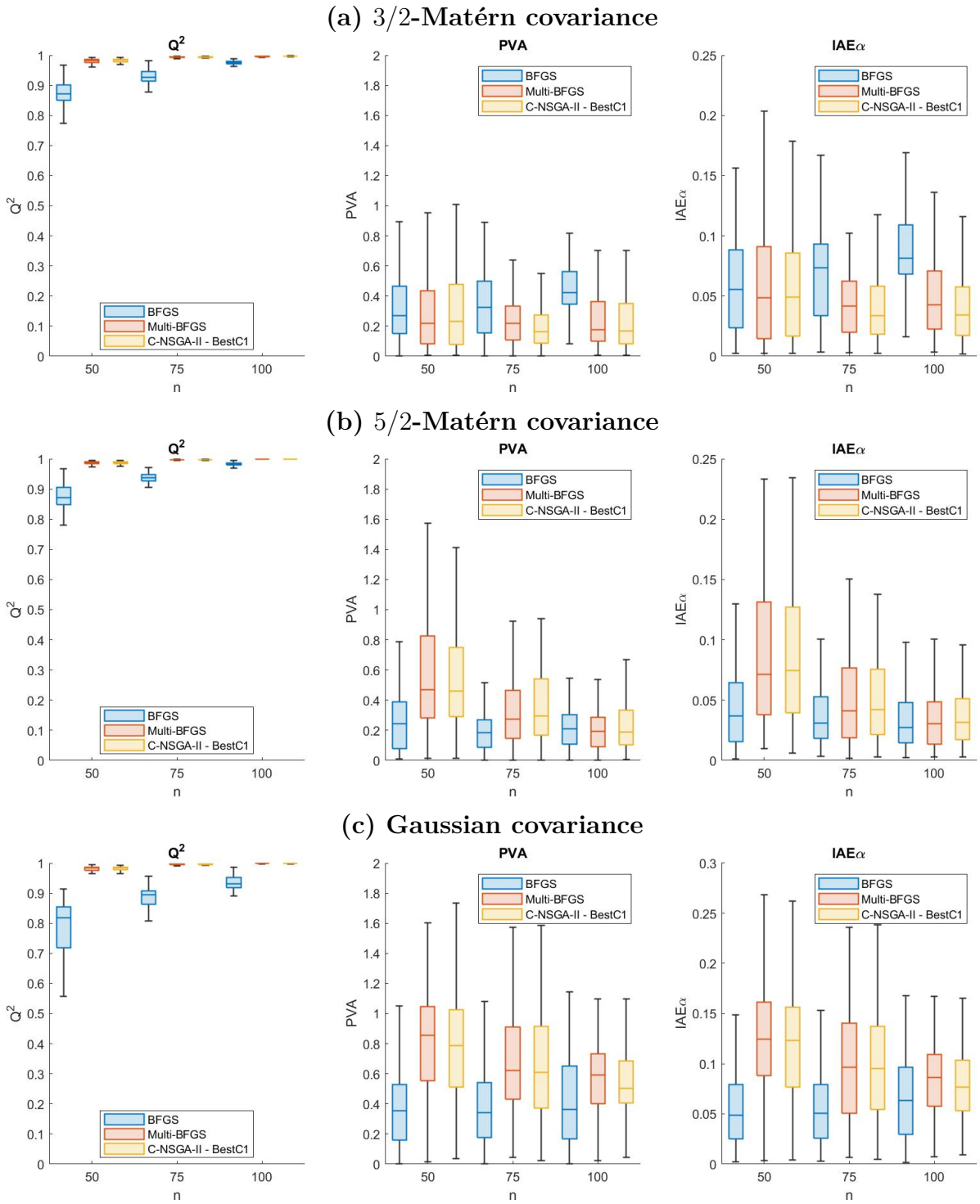

Figure 3: $\mathcal{M}_{\text{Dette-d8}}$ Function – Evolution of validation criteria, according to sample size n, for different hyperparameter estimation methods (GP with different covariances and all **without nugget effect**).

– If the Q^2 is very good and/or n is very large compared to the complexity of the function. The NLL function is probably very flat in a rather large area of the hyperparameters. As a result, its optimization is not discriminative enough, many values of the hyperparameters leading to interesting NLL (and Q^2) values. Taking into account the IAEα criterion allows to focus on more interesting hyperparameters to obtain more reliable prediction intervals. The example of $\mathcal{M}_{\text{Fried-d5}}$ with 3/2-Matérn covariance (Figure 4

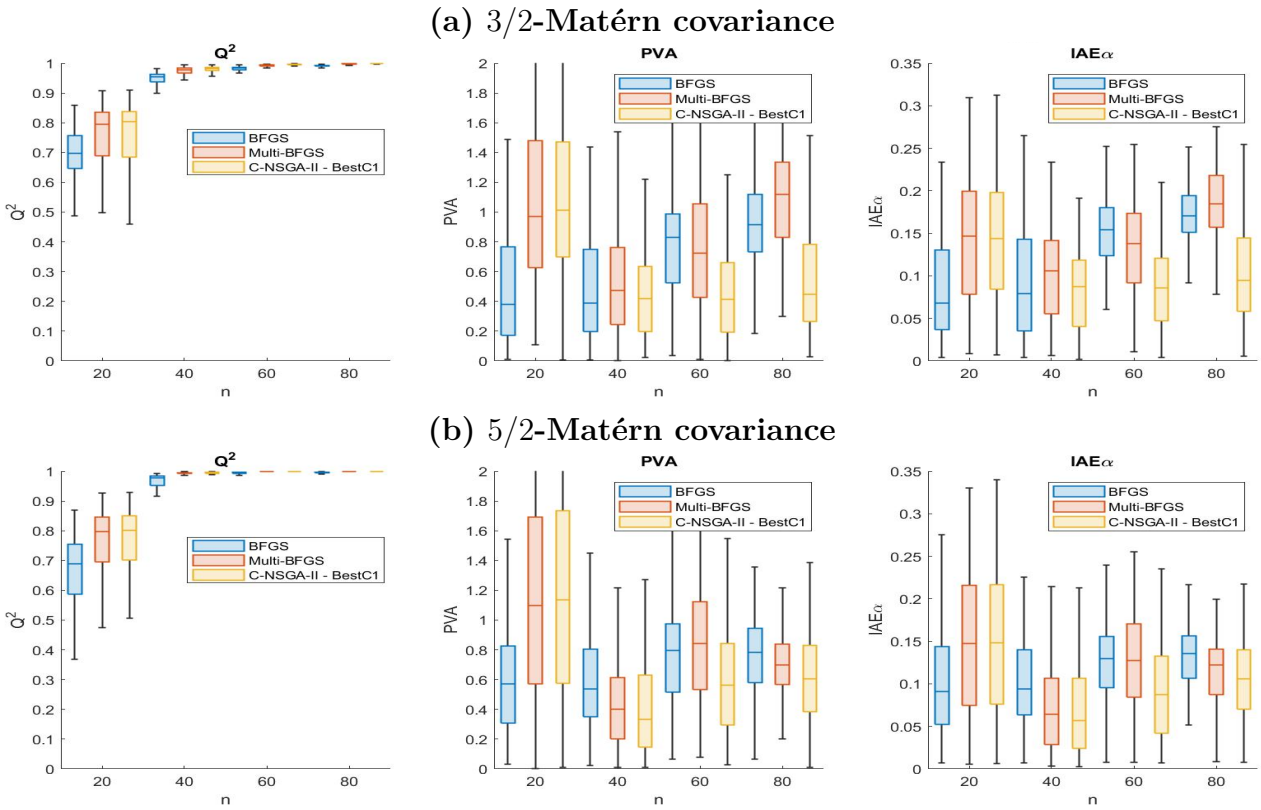

Figure 4: $\mathcal{M}_{\text{Fried-d5}}$ Function – Evolution of validation criteria, according to sample size n, for different hyperparameter estimation methods (GP with different covariances and all **without nugget effect**).

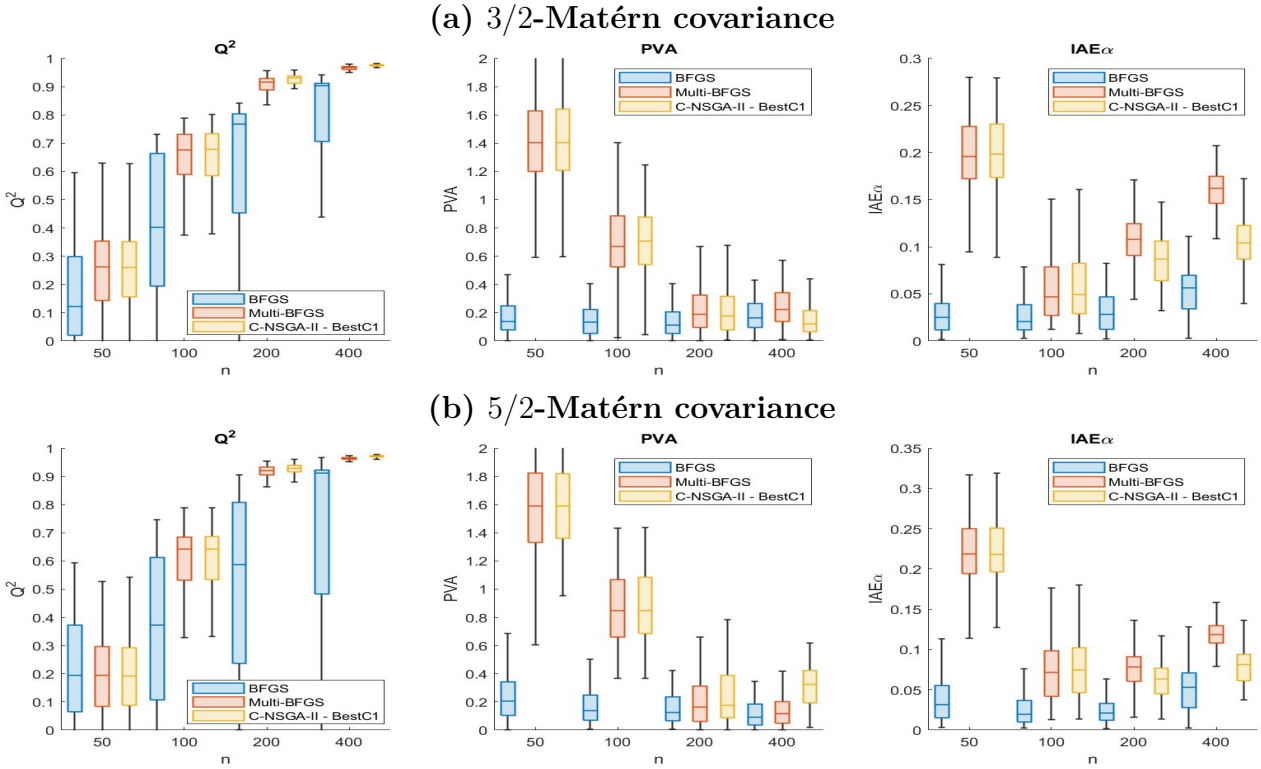

Figure 5: $\mathcal{M}_{\text{Marrel-d20}}$ Function – Evolution of validation criteria, according to sample size n, for different hyperparameter estimation methods (GP with different covariances and all **without nugget effect**).

plot(a)) is a good illustration.

- If the model is misspecified, i.e. if the covariance function does not match the regularity of the function. This is what is observed for example for $\mathcal{M}_{\text{Fried-d5}}$ with 3/2-Matérn covariance (Figure 4 plot (a)) or for the infinitely differentiable $\mathcal{M}_{\text{Ishig-d3}}$ function with a 3/2-Matérn covariance whose trajectories are only once differentiable (Figure C.13 plot (a), in Appendix C.2). In this particular case, the C-NSGA-II-BestC1 algorithm allows to divide by two the IAEα for $n = 150$.
- When the dimension of the inputs d is large (and consequently the number of hyperparameters in the case of a tensorized anisotropic stationary covariance). The C-NSGA-II-BestC1 algorithm provides significantly better and more robust results, less sensitive to sampling. This is what is observed for $\mathcal{M}_{\text{Marrel-d20}}$ in dimension $d = 20$ and $\mathcal{M}_{\text{Becker-d9}}$ in dimension $d = 9$ (see Appendix C.3).

5.2. Results with an additional estimated nugget effect

We now consider an additional homoscedastic nugget effect in the covariance in order to evaluate the robustness of the method with this additional parameter to estimate. This nugget effect is included in the proposed algorithm in the same way as the other hyperparameters: the nugget parameter λ is also estimated in the multi-objective optimization. Its variation domain is $\mathcal{D}_\lambda = [10^{-3}; 0.5]$ with an initial value set at $\lambda = 0.3$. A few results are given by Figures 6 to 8. As in the case without the nugget effect, a general improvement brought by the C-NSGA-II-BestC1 algorithm on PVA and IAEα criteria is observed, with no degradation of Q^2. In large dimension, the contribution of the constrained multi-objective algorithm is all the more significant on the reliability of the GP predictive law, provided that the metamodel is already sufficiently predictive ($Q^2 > 0.8$ e.g.), as it can be observed for $\mathcal{M}_{\text{Marrel-d20}}$ from $n = 200$ (see Figure 8). Note that on this last function, we observe a re-increase of PVA and IAEα indicators when the model becomes highly predictive ($Q^2 > 0.95$ for $n = 400$), this increase being mitigated by the use of C-NSGA-II-BestC1. In this case, the GP metamodel predicts the function almost perfectly but the width of the prediction intervals does not decrease quickly enough with n and they become too conservative.

Finally, similar benchmarks have also been performed considering simple Monte Carlo designs for the learning samples (instead of space-filling LHS designs). An extract of the results obtained is given in Appendix D. Overall, the results are close to those obtained with LHS designs, but more variable, due to the more variable sampling and, above all, less efficient for metamodeling purpose. Similar conclusions are drawn about the relative performance of the methods: the constrained multi-objective approach again gives better results (larger Q^2 and more reliable prediction intervals) and is significantly more robust to the sampling variance.

6. Application to an aquatic ecosystem model

In this section, a test case modeling a prey-predator chain in an aquatic ecosystem is studied. This so-called model MELODY (for MESocosm structure and functioning for representing LOtic DYnamic ecosystems) simulates the functioning of aquatic mesocosms and the impact of toxic substances on the dynamics of populations. Inside the model, two compartments linked within a prey-predator chain are considered: the Periphyton and the Grazers. The Periphyton-Grazers sub-model is representative of processes involved in dynamics of primary producers and primary consumers, i.e. photosynthesis, excretion, respiration, egestion, mortality, sloughing and predation. More details are available in Ciric et al. [25] and Iooss et al. [26]. In this test case, a total number of $d = 20$ uncertain and independent input parameters are considered. These parameters characterize,

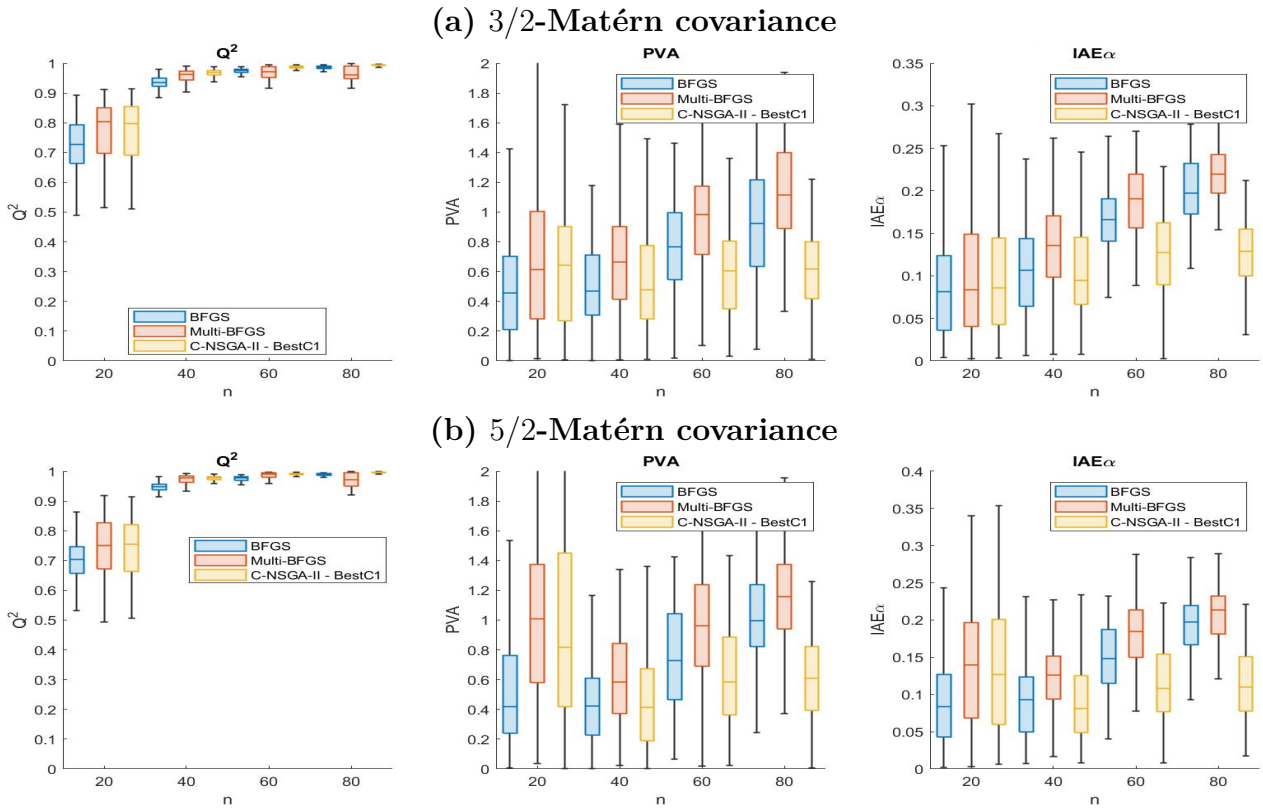

Figure 6: $\mathcal{M}_{\text{Fried-d5}}$ Function – Evolution of validation criteria, according to sample size n, for different hyperparameter estimation methods (GP with different covariances and all **with an estimated nugget effect**).

among others, the photosynthesis, consumption, respiration, mortality or excretion rates of both populations. In absence of any expert opinion or relevant information on their uncertainty, these parameters are assumed to follow uniform distributions whose variation intervals are given in Ciric et al. [25]. As output from the model, we focus on the Grazers biomass at a given reference time (day 60 of simulations), denoted Y_G.

A sample of $n = 100$ simulations of the model MELODY is available, drawn from a LHS with low discrepancy [18]. Previous sensitivity analysis studies [27] have revealed the presence of strong non-linear and interaction effects for Y_G. Considering the inputs' high dimensionality, these outputs are consequently very complex to emulate, making this test case relevant to evaluate the contribution of the multi-objective algorithm.

6.1. Description of the dataset and lognormal-kriging approach

The highly dispersed nature of the data requires to apply a preliminar logarithmic transformation to represent them. The histograms of the obtained values, denoted $Z_G = \log(Y_G)$, is given by Figure 9. A kernel density estimation plot is also added to provide a graphical illustration of the probability density function. This logarithmic transformation will also be considered for the metamodeling, considering a lognormal-kriging approach to return to the initial space of the output values [28, 29]. Note that a more general Box-Cox transformation could also be considered. Hence, a predictive Gaussian distribution $\mathcal{N}(\hat{z}_G(\mathbf{x}), \hat{s}^2_{z_G}(\mathbf{x}))$ is obtained for Z_G (see Eqs (4) and (5)). Lognormal-kriging backtransformations [29] are then used to obtain the predictor and associated variance prediction for the original data Y_G:

$$\hat{y}_G(\mathbf{x}) = e^{\left(\hat{z}_G(\mathbf{x}) + 0.5\hat{s}^2_{z_G}(\mathbf{x})\right)} \tag{12}$$

$$\hat{s}^2_Y(\mathbf{x}) = \left(e^{\hat{s}^2_{z_G}(\mathbf{x})} - 1\right) e^{\left(2\hat{z}_G(\mathbf{x}) + \hat{s}^2_{z_G}(\mathbf{x})\right)}. \tag{13}$$

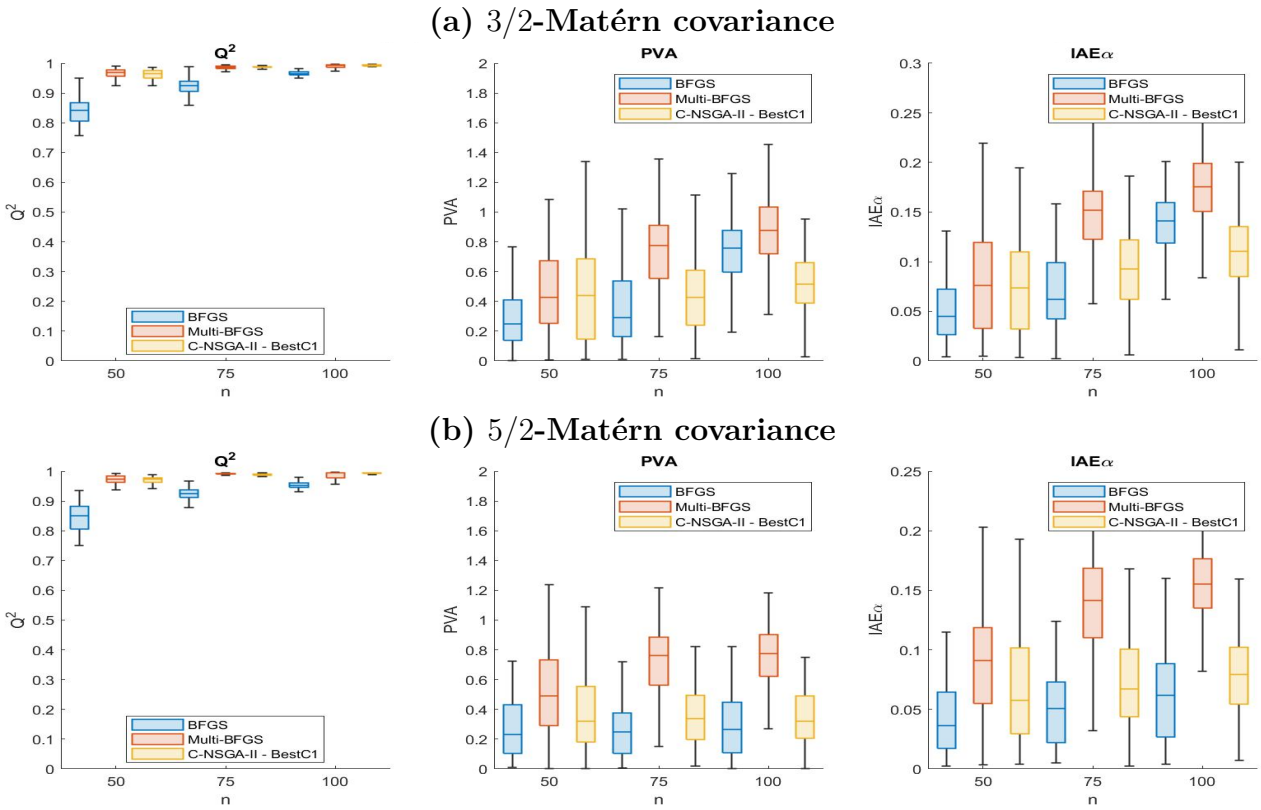

Figure 7: $\mathcal{M}_{\text{Dette-d8}}$ Function – Evolution of validation criteria, according to sample size n, for different hyperparameter estimation methods (GP with different covariances and all **with an estimated nugget effect**).

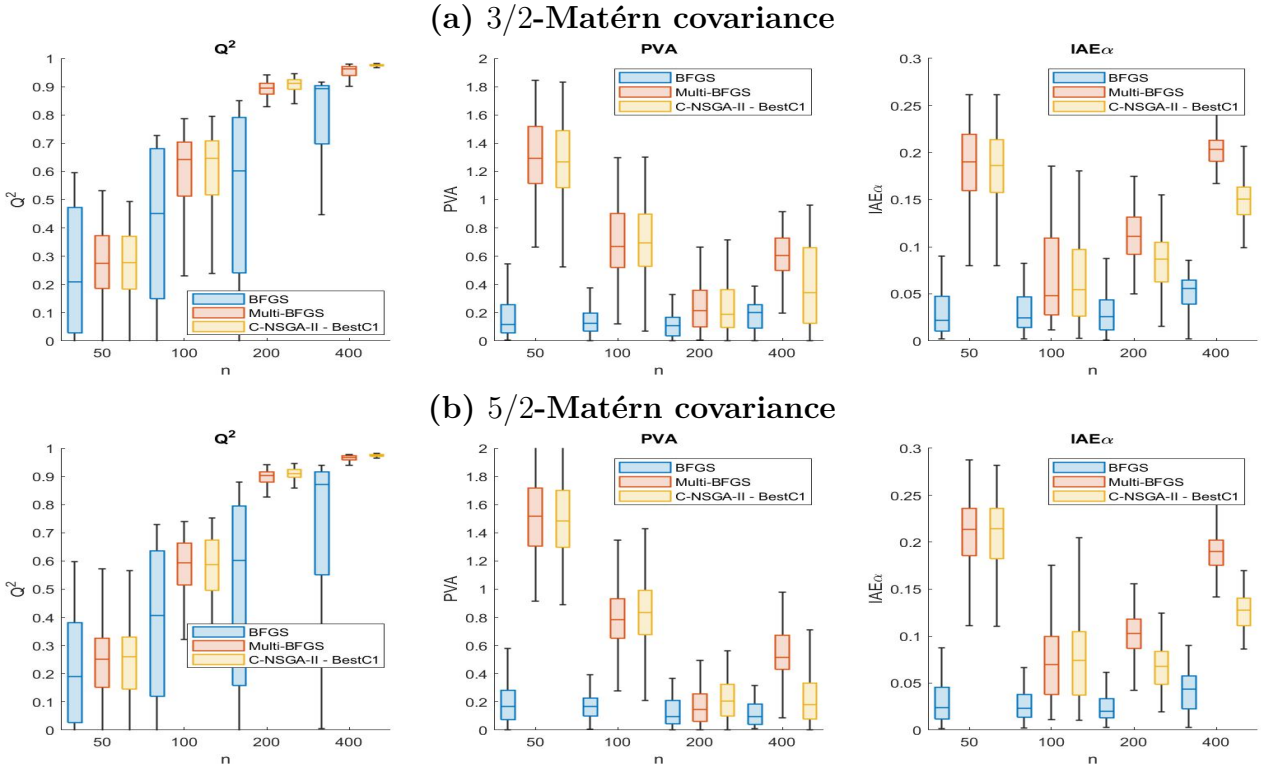

Figure 8: $\mathcal{M}_{\text{Marrel-d20}}$ Function – Evolution of validation criteria, according to sample size n, for different hyperparameter estimation methods (GP with different covariances and all **with an estimated nugget effect**).

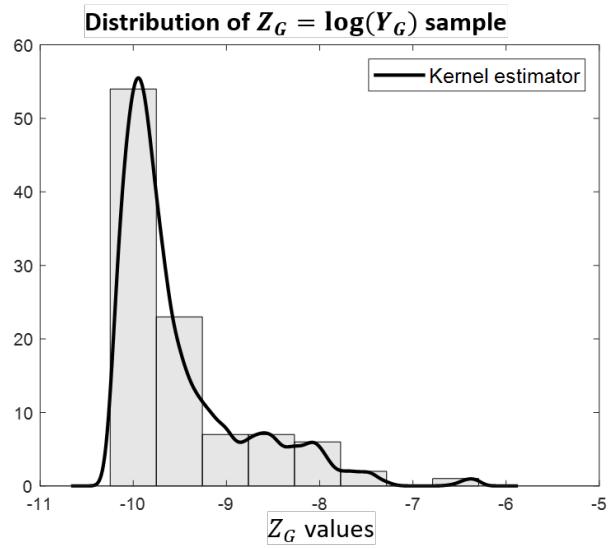

Figure 9: MELODY test case – Histogram of the output Y_G after log-transformation, for the $n = 100$-size dataset.

6.2. Results of Gaussian process metamodeling

The lognormal-kriging approach is applied to the dataset: a GP is fitted on Z_G sample with either the usual multistart-BFGS or our new C-NSGA-II multi-objective algorithm (C-NSGA-II-BestC1 algorithm). Different covariance functions are considered, with or without an estimated nugget effect, and a constant mean is considered in all cases. Validation criteria are computed by LOO on the learning sample, for both Z_G and Y_G. It should be noted that the computation of criteria by cross-validation for small data results in a double loop cross-validation structure for the C-NSGA-II algorithm, since it already includes a cross-validation loop. However this internal loop remains largely tractable for $n = 100$ and thanks to Dubrule's formulas [10]. The obtained results, with the most interesting highlighted, are given in Table 1 (columns "Multi-BFGS" and "New C-NSGA-II Algorithm" of Tables 1(a) and 1(b)).

First, for the transformed variable Z_G, the predictivity is similar for the different metamodels with nearly 90% of output variance explained. Results differ when we go back to the variables Y_G. This is explained by the expression of the log-kriging predictor (Eq. (12)) which depends on the prediction variance $\hat{s}_Z^2$. The multi-objective algorithm, which has more reliable prediction variances for Z, has consequently a much better predictivity for Y, with Q^2 on average increased by 0.04. Log-kriging is a further illustration of the need for reliable predictive variances in GP regression.

Moreover, C-NSGA-II algorithm also yields more reliable prediction intervals, with a IAEα significantly reduced, whatever the covariance considered and whether or not there is a nugget effect. Note that for the IAEα criterion, the results are identical in both spaces, the empirical coverage function in GP framework being invariant by strictly monotonic transformation of the output variable. Note that IAEα is invariant by logarithmic transformation and more generally by strictly monotonic transformations since they preserve (for increasing transformations) or reverse (for decreasing transformations) the order of data. For example, in the case of the logarithmic transformation, the quantiles of the predictive law for Y are equal to the exponential of the quantiles of the GP predictive law obtained for $Z_G = \log(Y_G)$. The values of $\hat{\Delta}(\alpha)$ computed with the set of z_G or $y_G = e^{z_G}$ are therefore equal.

Some diagnostic plots are also given by Figure 10 with 5/2-Matérn covariance, to illustrate more clearly the improvement brought by the C-NSGA-II algorithm on the accuracy of prediction intervals. As an indication of the degree of non-linearity of the output, it may be mentioned that

a linear regression with an ElasticNet-type penalization [30] leads to $Q^2 = 0.57$ for Z_G and to zero Q^2 if directly fitted on Y_G.

(a) With nugget parameter (estimated)

Data	Covariance	Predictivity Coefficient Q^2				IAEα			
		Multi-BFGS	New C-NSGA-II Algorithm	RobustGaSP	New C-NSGA-II with Robust NLL	Multi-BFGS	New C-NSGA-II Algorithm	RobustGaSP	New C-NSGA-II with Robust NLL
Z_G	Matern3/2	0,89	0,89	0,64	0,86	0,10	0,07	0,04	**0,02**
	Matern5/2	0,90	0,89	0,87	0,88	0,09	**0,02**	0,07	**0,03**
	Gaussian	0,89	0,89	0,87	0,88	0,11	**0,01**	**0,06**	**0,04**
		Multi-BFGS	New C-NSGA-II Algorithm	RobustGaSP	New C-NSGA-II with Robust NLL	Multi-BFGS	New C-NSGA-II Algorithm	RobustGaSP	New C-NSGA-II with Robust NLL
Y_G	Matern3/2	0,70	0,74	0,25	**0,77**	0,10	0,07	0,04	**0,02**
	Matern5/2	0,77	**0,82**	0,66	**0,83**	0,09	**0,02**	0,07	**0,02**
	Gaussian	0,75	**0,79**	0,66	**0,79**	0,08	**0,02**	0,06	**0,04**

(b) Without nugget parameter

Data	Covariance	Predictivity Coefficient Q^2				IAEα			
		Multi-BFGS	New C-NSGA-II Algorithm	RobustGaSP	New C-NSGA-II with Robust NLL	Multi-BFGS	New C-NSGA-II Algorithm	RobustGaSP	New C-NSGA-II with Robust NLL
Z_G	Matern3/2	0,89	0,88	0,81	0,86	0,10	0,06	0,03	0,02
	Matern5/2	0,90	0,88	0,91	0,88	0,08	**0,02**	0,07	**0,02**
	Gaussian	0,90	0,89	0,93	0,86	0,06	**0,03**	0,06	**0,01**
		Multi-BFGS	New C-NSGA-II Algorithm	RobustGaSP	New C-NSGA-II with Robust NLL	Multi-BFGS	New C-NSGA-II Algorithm	RobustGaSP	New C-NSGA-II with Robust NLL
Y_G	Matern3/2	0,70	**0,75**	0,47	0,74	0,10	0,06	0,03	0,02
	Matern5/2	0,78	**0,84**	**0,83**	**0,88**	0,08	**0,02**	0,07	**0,02**
	Gaussian	0,70	**0,72**	**0,89**	**0,74**	0,06	**0,03**	0,06	**0,01**

Table 1: MELODY Output Y_G – Validation criteria computed by cross validation for GP with different covariances and estimation methods, without and with (estimated) nugget effect. The significantly better (resp. best) results are framed in black (resp. red).

6.3. Comparison with the RobustGaSP simplified Bayesian approach

To the best of our knowledge, and as detailed in our companion paper [3], the RobustGaSP approach proposed by Gu et al. [13] is the most interesting existing Bayesian approach to perform robust hyperparameter estimation, while being tractable in large dimension. To carry out the comparison with our approach, the R package RobustGaSP [31] is used. The jointly robust prior is considered to efficiently approximate the reference prior (argument `prior_choice` set at `ref_approx` in the `rgasp` function). In addition, we have also combined our multi-objective algorithm with RobustGaSP. To do so, the new algorithm is adapted by, replacing the NLL in the multi-objective procedure, by the marginal posterior of $\boldsymbol{\theta}$ proposed by Gu et al. [13] with their jointly robust priors (see Eq. (27) of companion paper [3]). The obtained results for RobustGaSP approach and C-NSGA-II combined with RobustGaSP are given by Table 1 in columns 'RobustGaSP" and "New C-NSGA-II with Robust NLL", respectively.

If a nugget effect is considered and estimated, the C-NSGA-II algorithm gives significantly better results than RobustGaSP, for both Q^2 and IAEα criteria. Best results are obtained with 5/2-Matérn covariance. If no nugget effect is considered, RobustGaSP performs much better for the smoother covariances, namely 5/2-Matérn and Gaussian ones. Compared with C-NSGA-II algorithm, it yields an equivalent Q^2 for 5/2-Matérn covariance but much better Q^2 for Gaussian covariance. But in any case, it performs less well in terms of prediction interval reliability with a IAEα two to three times higher. RobustGasp results are therefore very sensitive to GP specifications (covariance function and nugget), whereas our C-NSGA-II algorithm is more robust to these choices. If we finally look at the combination of both approaches in the so-called "New C-NSGA-II with Robust NLL" algorithm, its results are promising: it seems to be a good option since the

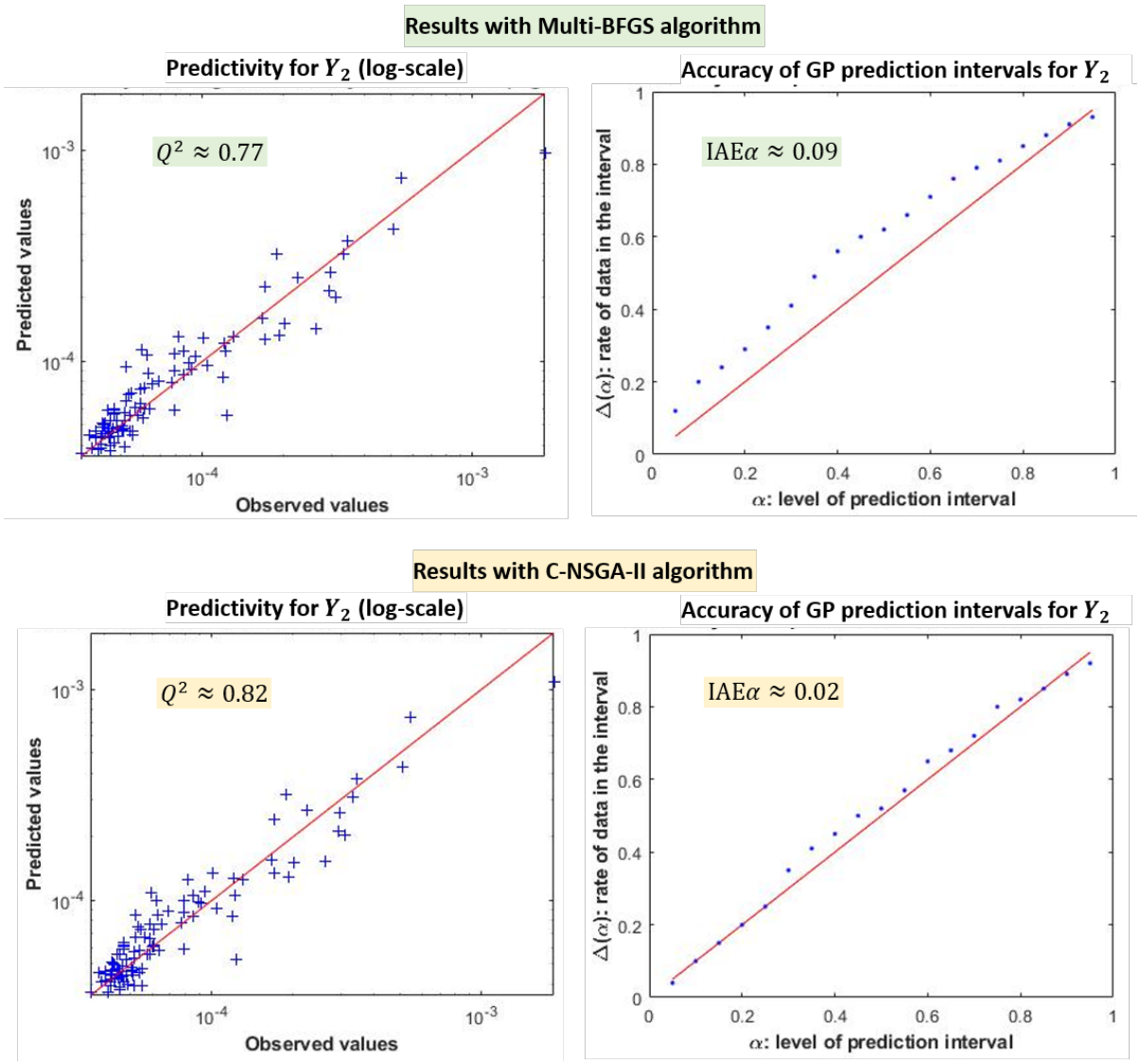

Figure 10: MELODY Output Y_G – Diagnostic plots for GP metamodeling with 5/2-Matérn covariance and nugget effect: LOO predicted values versus exact values on the left (with log-scale) and α-plot on the right.

results of C-NSGA-II algorithm are systematically improved. Moreover, it leads to the best overall result with 5/2-Matérn covariance and nugget effect ($Q^2 = 0.88$ and IAE$\alpha = 0.02$).

7. Conclusion

In the framework of emulation of numerical simulators with GP regression, this work has introduced a new algorithm for the estimation of GP covariance parameters, referred to as GP hyperparameters. Since the use of the entire GP predictive distribution is one of the major assets of this metamodel, a reliable and robust estimation of the hyperparameters is a cornerstone of successful metamodeling. For this purpose, and convinced of the value of going beyond the usual maximum likelihood approach, a new constrained multi-objective algorithm has been proposed. Based on a thorough analysis of the links between different estimation and validation criteria, this algorithm consists in jointly maximizing the likelihood of the observations as well as the empirical coverage function of GP prediction intervals computed by a leave-one-out (LOO) procedure, under the constraint of not degrading the GP predictivity. Particular care has been taken to detail and justify the parametric choices of the algorithm to facilitate its implementation and favor

reproducible results.

A large benchmark on analytical functions of variable dimension (1-D to dimension 20) has been performed, considering different designs of experiments and different covariance models. The new algorithm was compared to usual simple and multi-start Quasi-Newton algorithms (which focuses only on the likelihood optimization). The constrained multi-objective algorithm provided better results in terms of predictivity and reliability of prediction intervals, and was much more robust to the sampling variance. The improvement brought by the algorithm is all the more interesting when the covariance model is misspecified, when the number of GP hyperparameters is large or even when the size of the data is large.

The application relevance of this algorithm has been shown on a real test case modeling an aquatic ecosystem and more precisely a prey-predator chain. The GP metamodeling is used to predict the biomass of the two species at a given time. Once again, the multi-objective algorithm performs better than standard algorithms. Furthermore, the log-kriging approach notably illustrates the need for well-estimated and reliable prediction variances in GP regression. Finally, these good results have been confirmed by a comparison with the RobustGaSP method of Gu et al. [13], which is to our knowledge the only efficient and tractable Bayesian method. The multi-objective algorithm achieves results at least as good as those of RobustGaSP, while being less sensitive to GP specifications (covariance function and nugget). The combination of both approaches in a modified version of our algorithm yields promising results, taking the best of both methods.

This paves the way for other improvements and variants of the algorithm (e.g. use of gradients for the optimization of LOO-based criteria, see Petit et al. [8], focusing the multi-objective procedure only on the main influential hyperparameters to reduce the computation cost, etc.). But the aim of this work was elsewhere: to demonstrate the value of going beyond standard estimation approaches by combining several criteria, and of taking particular care to control its quality *in fine*. GP metamodel is a powerful tool but its implementation requires a certain expertise which can be a limiting factor in practice for its automation and its use on complex industrial cases. While this paper attempts to provide guidance, the fact remains that GP validation also requires careful and informed consideration of the topic, to ensure confidence in its use for predictive purposes.

Acknowledgments

This work was funded by the French ANR project SAMOURAI (ANR-20-CE46-0013).

Appendix A. Additional results on the links between NLL and validation criteria

As in Section 3, a further illustration of the behavior of NLL and validation criteria for the "re-scaled" Branin function is given below by Figure A.11, for a dataset of size $n = 50$.

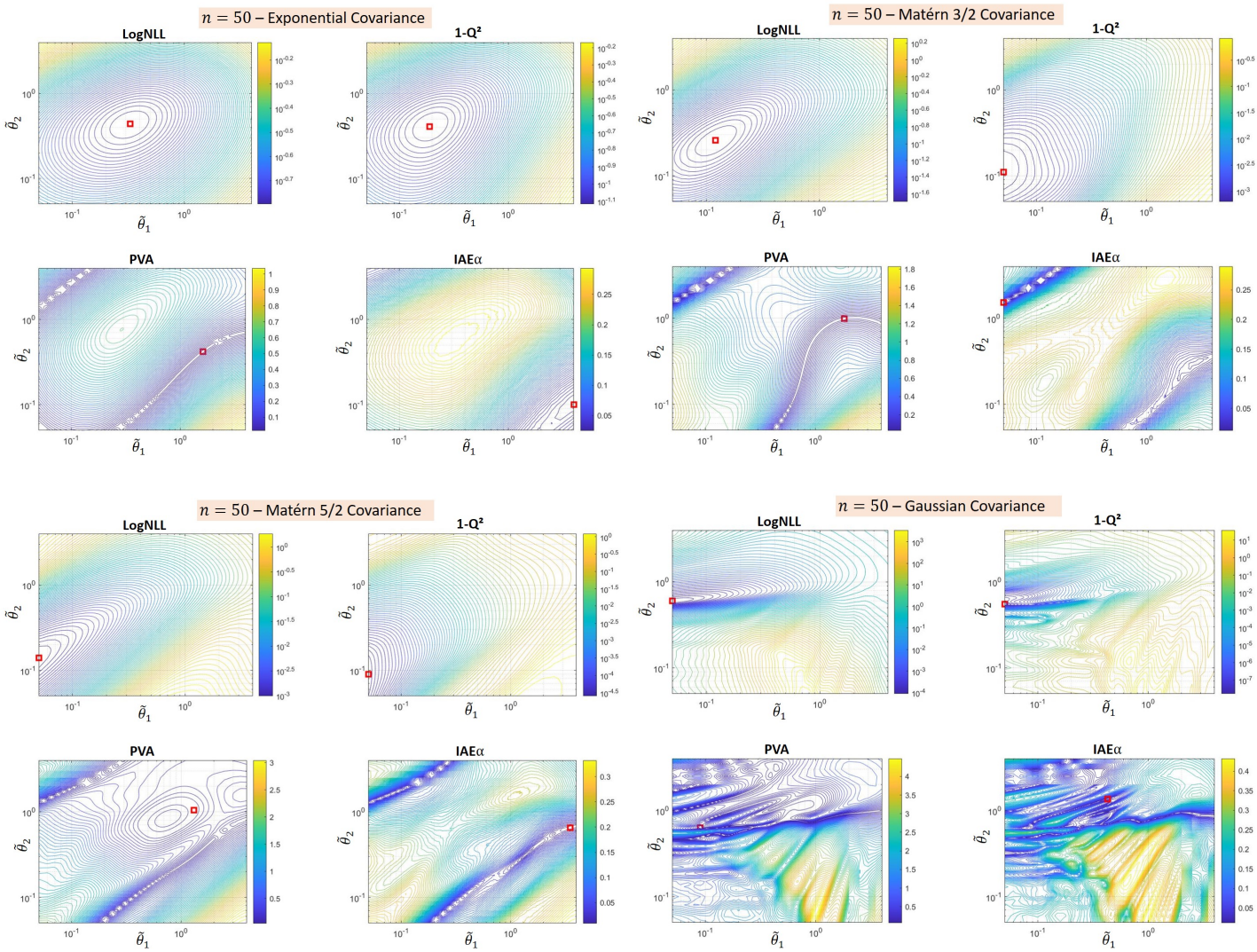

Figure A.11: $\tilde{\mathcal{M}}_{\text{Branin}}$ Function – Comparison of NLL (computed on the learning sample) and validation criteria (computed on a test sample), for a GP built on a Monte Carlo learning sample of size $n = 50$. The optimal value for each quantity is indicated by a red square.

Appendix B. Parameters of NSGA-II algorithm

The default (and recommended) values used in all the paper for the NSGA-II algorithm are specified in Table The default (and recommended) values for the NSGA-II algorithm are specified in Table B.2. They are fixed and used for all the tests in the article, for the sake of reproducibility of results.

Maximum number of iterations	Maxlt = 50
Population size	nPop = 80
Crossover percentage	pCrossover = 0.5
Mutation percentage	pMutation = 0.5
Mutation rate	mu = 0.02
Number of parnets (offsprings)	nCrossover = 2*round(pCrossover*nPop/2)
Number of mutants	nMutation = round(pMutation*nPop)
Mutation step size	sigma = $0.1 * (\tilde{\theta}_{max} - \tilde{\theta}_{min})$
Generating reference points	nDivision = 10 Zr = GenerateReferencePoints(nObj, nDivision)

Table B.2: Default values of parameters of NSGA-II algorithm.

Appendix C. Benchmark of Section 5: details on test functions and additional results

Appendix C.1. Details on a modified version of Friedman function

The analytical model proposed by Marrel et al. [24] and inspired from the Friedman function [22] is defined in dimension $d = 20$ by:

$$\mathcal{M}_{Marrel-d20}(\mathbf{X}) = a_1 \sin\left(6\pi\ X_1^{5/2}(X_2 - 0.5)\right) + a_2(X_3 - 0.5)^2 + a_3 X_4 + a_4 X_5 + r_{X6,\ldots,X_{15}} \quad \text{(C.1)}$$

where $r_{X_6,\ldots,X_{15}} = \dfrac{a_5}{\sqrt{(\sum_{i=6\ldots15} i^2)}} \sum_{i=6\ldots15} \sqrt{12} i(X_i - 0.5)$ and $\mathbf{X} = (X_1, \ldots, X_{20})$ are independent and uniform random variables on $[0, 1]$.

The model depends only on the 15 first inputs. The first term represents a strong and non monotonic interaction between the two first inputs. The second term is a quadratic function of X_3 while the other ones are linear. The parameters for tuning the influence of the different inputs are chosen as follows: $\mathbf{a} = (5,\ 20,\ 8,\ 5,\ 1.5)^\top$. Under this parametrization, X_2 explains alone around 10% of the output variance, X_1 has no individual effect but its interaction effect with X_2 is strong (around 30% of the output variance). X_3, X_4 and X_5 only have individual effects (no interaction) and explain, respectively, around 11%, 28% and 10% of the output variance. The effects of the ten remaining inputs (X_6 to X_{15}) represent around 11.5% of the output variance. The one-dimensional mean effects of the five first (and main influential) inputs are plotted on Figure (C.12).

Appendix C.2. Results for Ishighami function in dimension $d = 3$

Following a protocol strictly similar to that described in Section 5, tests were carried out on other analytic functions, among them the Ishighami and Becker's functions. The Ishighami function introduced by Ishigami and Homma [32] is defined in dimension $d = 3$ by:

$$\mathcal{M}_{\text{Ishig-d3}}(X_1, X_2, X_3) = \sin(2\pi X_1 - \pi) + 7\ \sin^2(2\pi X_2 - \pi) + 0.1\ (2\pi X_3 - \pi)^4\ \sin(2\pi X_1 - \pi); \quad \text{(C.2)}$$

Results obtained for Ishighami function without nugget effect are given by Figure C.13.

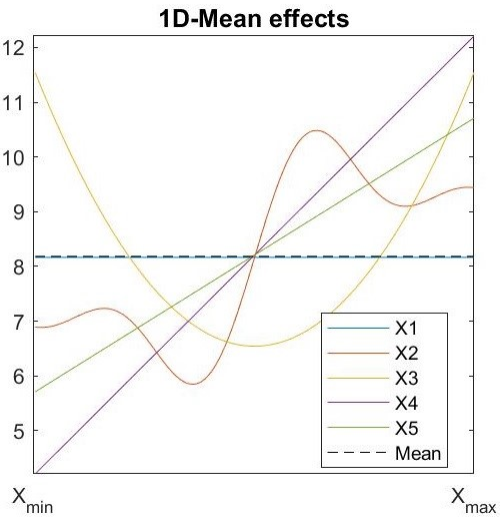

Figure C.12: $\mathcal{M}_{\text{Marrel-d20}}$ – Illustration of the one-dimensional mean effects for the first five inputs, figure extracted from Marrel et al. [24].

Appendix C.3. Results for Becker's function in dimension $d = 20$

We also considered the Becker's function in dimension $d = 20$ proposed by Becker [33]. This function relies on a list of univariate basis functions that may possibly appear, representing response features observed in physical models. This list of basis functions is the following:

$$
\begin{aligned}
&f^1(x) = x && \text{(linear)}\\
&f^2(x) = x^2 && \text{(quadratic)}\\
&f^3(x) = x^3 && \text{(cubic)}\\
&f^4(x) = \left(e^x - 1\right)/(e-1) && \text{(exponential)}\\
&f^5(x) = \frac{1}{2}\sin(2\pi x) + \frac{1}{2} && \text{(periodic)}\\
&f^6(x) = 1 \text{ if } x > \frac{1}{2} \text{ and } 0 \text{ otherwise} && \text{(discontinuity)}\\
&f^7(x) = 0 && \text{(no effect)}\\
&f^8(x) = 4\left(x - \frac{1}{2}\right)^2 && \text{(quadratic, non-monotonic)}\\
&f^9(x) = (10 - 1/1.1)^{-1}(x + 0.1)^{-1} - 0.1 && \text{(inverse with small shift)}
\end{aligned}
$$

where each basis function has been scaled so that inputs in $[0, 1]$ also map to outputs in $[0, 1]$ (see Figure of Becker [33]).

Given these inputs, the function $\mathcal{M}_{\text{Becker}}$ is built as a sum of the main effects and interactions:

$$
\begin{aligned}
\mathcal{M}_{\text{Becker}}(\boldsymbol{x}, \boldsymbol{u}, \mathbf{V}, \mathbf{w}, \Theta) = &\sum_{i=1}^{d} a_i f^{u_i}(x_i) + \sum_{i=1}^{d_2} b_i f^{u_{V_{i,1}}}\left(x_{V_{i,1}}\right) f^{u_{V_{i,2}}}\left(x_{V_{i,2}}\right)\\
&+ \sum_{i=1}^{d_3} c_i f^{u_{W_{i,1}}}\left(x_{W_{i,1}}\right) f^{u_{W_{i,2}}}\left(x_{W_{i,2}}\right) f^{u_{W_{i,3}}}\left(x_{W_{i,3}}\right),
\end{aligned}
$$

where $\Theta = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$ are weighting coefficients applied to the main effect, second and third-order interaction terms respectively. One basis function is therefore specified for each input variable as its main effect. The interaction terms are generated as mixtures of these main effect terms.

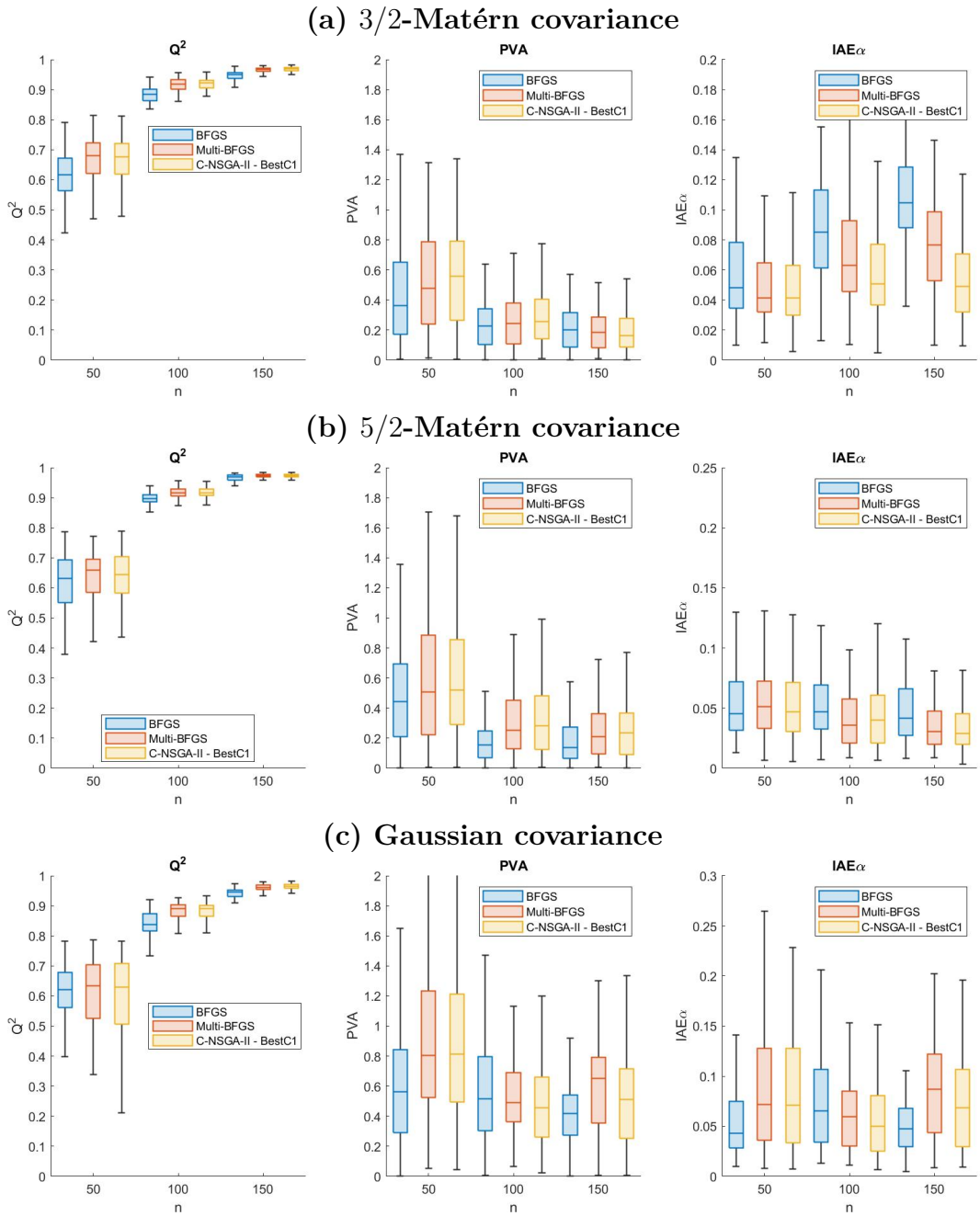

Figure C.13: $\mathcal{M}_{\text{Ishig-d3}}$ Function – Evolution of validation criteria, according to sample size n, for different hyper-parameter estimation methods (GP with different covariances and without nugget effect).

Note that this restriction on the construction of the interaction terms is done to avoid additional parameters, but it could be set otherwise.

To cover a large spectrum of models encountered in physical applications, random functions are then generated from the definition of $\mathcal{M}_{\text{Becker}}$ by sampling the parameters of $\{\boldsymbol{u}, \mathbf{V}, \mathbf{W}, \Theta\}$ from an appropriate distribution. More precisely, as suggested by Becker [33], $\boldsymbol{u}$ and $\mathbf{V}$ are sampled independently from uniform discrete distributions on $[\![1, d]\!]$. Each parameter of Θ is sampled independently from a mixture of two zero-mean Gaussian distributions: one with a low variance equals to 0.5, and another with a high variance equals to 5. The mixture parameters are set at 0.7 and 0.3, respectively for the two distributions. Note that for our numerical benchmark, we only considered functions with interaction effects of second order at most (i.e., we have imposed that

$c_i = 0 \ \ \forall\, i$). Results obtained for the Becker function in dimension $d = 9$ without nugget effect are illustrated by figure C.14.

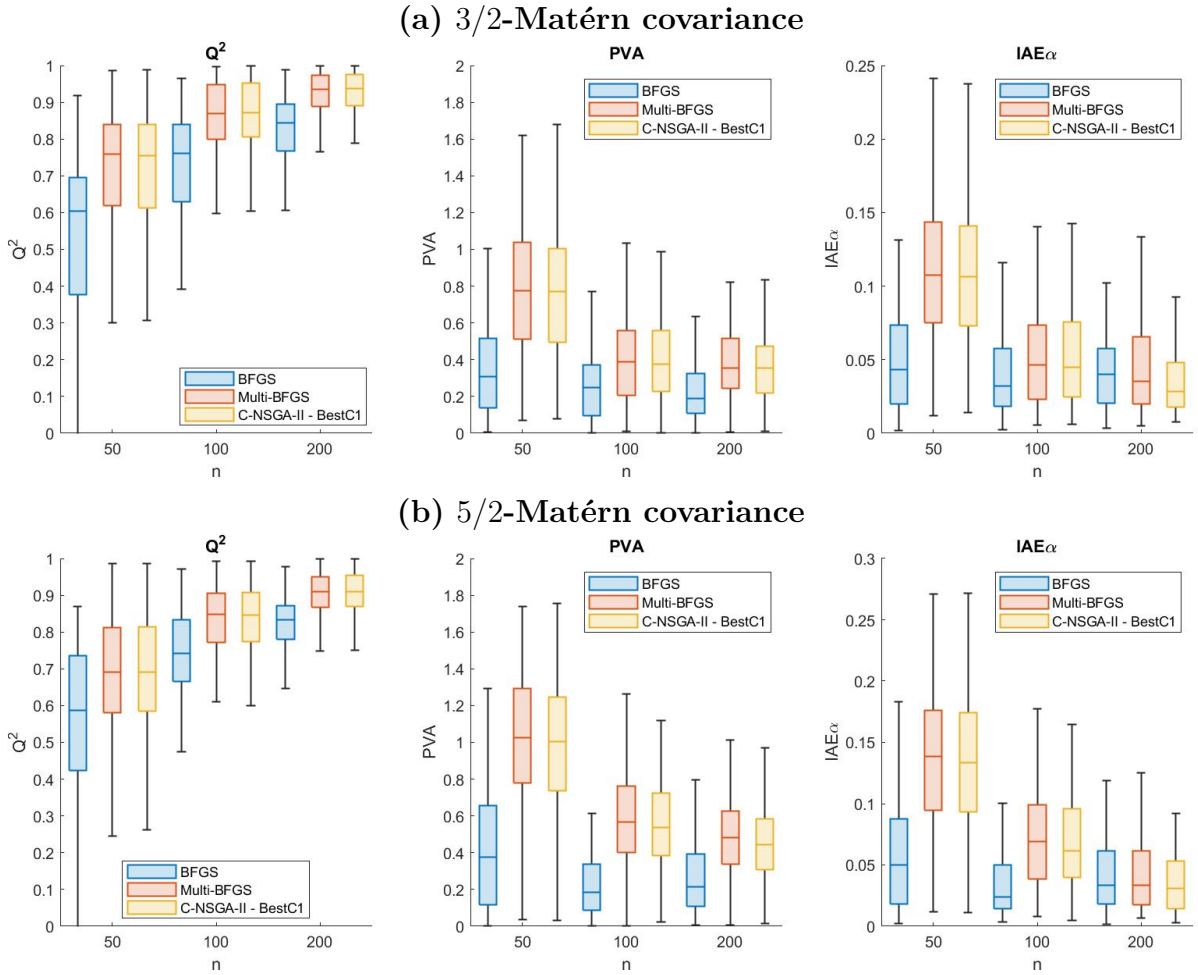

Figure C.14: $\mathcal{M}_{\text{Becker-d9}}$ Function – Evolution of validation criteria, according to sample size n, for different hyperparameter estimation methods (GP with different covariances and without nugget effect).

Appendix D. Benchmark of Section 5: results with other design choices

Similar benchmarks to those performed in Section 5 have been performed with Monte Carlo designs (instead of LHS space-filling designs), without and with an additional estimated nugget effect in covariance function. Similar results to those in Section 5 were obtained for all test functions, with even more significant outperformance for the algorithm C-NSGA-II-BestC1 when Monte Carlo designs are used. An illustration of these results is given below by Figure D.15 for the Friedman function ($d = 5$) with an estimated nugget effect.

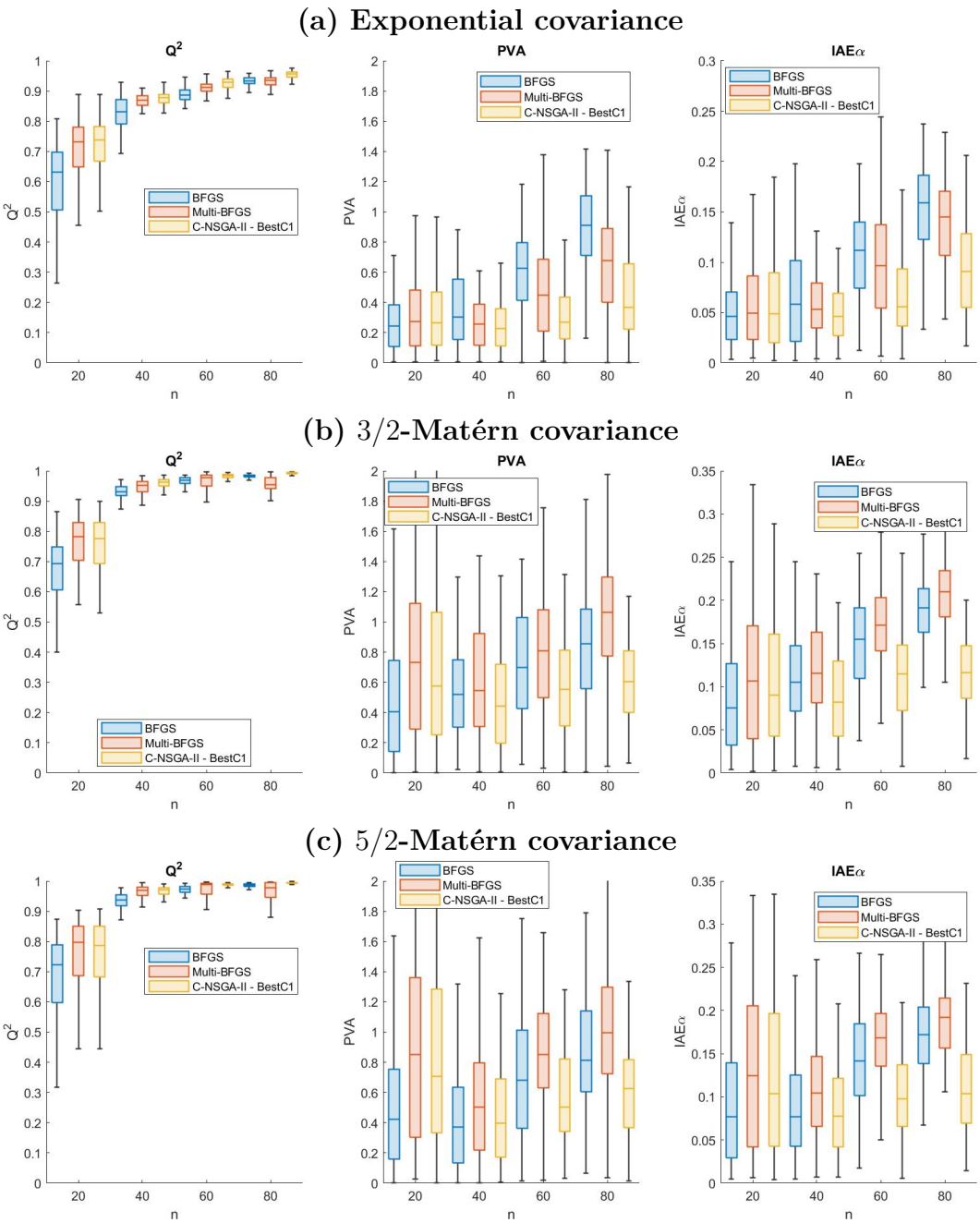

Figure D.15: $\mathcal{M}_{\text{Fried-d5}}$ Function – Evolution of validation criteria, according to sample size n, for different estimation methods and from **Monte Carlo learning samples** (different covariances, all with an estimated nugget effect).

References

[1] N. Villa-Vialaneix, M. Follador, M. Ratto, A. Leip, A comparison of eight metamodeling techniques for the simulation of N2O fluxes and N leaching from corn crops, Environmental Modelling & Software 34 (2012) 51–66.

[2] S. Afshari, F. Enayatollahi, X. Xu, X. Liang, Machine learning-based methods in structural reliability: A review, Reliability Engineering & System Safety 219 (2022) 108223.

[3] A. Marrel, B. Iooss, Probabilistic surrogate modeling by Gaussian process: A review on recent insights in estimation and validation, Preprint (2023).

[4] R. Ghanem, D. Higdon, H. Owhadi (Eds.), Springer Handbook on Uncertainty Quantification, Springer, 2017.

[5] J. N. Fuhg, A. Fau, U. Nackenhorst, State-of-the-art and comparative review of adaptive sampling methods for kriging, Archives of Computational Methods in Engineering 28 (2021) 2689–2747.

[6] M. Moustapha, S. Marelli, B. Sudret, Active learning for structural reliability: Survey, general framework and benchmark, Structural Safety 96 (2022) 102174.

[7] C. Demay, B. Iooss, L. L. Gratiet, A. Marrel, Model selection for Gaussian process regression: an application with highlights on the model variance validation, Quality and Reliability Engineering International Journal 38 (2022) 1482–1500.

[8] S. Petit, J. Bect, P. Feliot, E. Vazquez, Model parameters in Gaussian process interpolation: an empirical study of selection criteria, 2023. URL: https://hal-centralesupelec.archives-ouvertes.fr/hal-03285513.

[9] N. Acharki, A. Bertoncello, J. Garnier, Robust prediction interval estimation for Gaussian processes by cross-validation method, Computational Statistics & Data Analysis 178 (2023) 107597.

[10] O. Dubrule, Cross validation of kriging in a unique neighborhood, Journal of the International Association for Mathematical Geology 15 (1983) 687–699.

[11] S. Petit, Improved Gaussian process modeling: Application to Bayesian optimization, Thèse de l'Université Paris-Saclay, 2022.

[12] T. Karvonen, C. J. Oates, Maximum likelihood estimation in Gaussian process regression is ill-posed, Journal of Machine Learning Research 24 (2023) 1–47.

[13] M. Gu, X. Wang, J. O. Berger, Robust Gaussian stochastic process emulation, The Annals of Statistics 46 (2018) 3038 – 3066.

[14] L. W. Schruben, A coverage function for interval estimators of simulation response, Management Science 26 (1980) 18–27.

[15] V. Picheny, T. Wagner, D. Ginsbourger, A benchmark of kriging-based infill criteria for noisy optimization, Structural and Multidisciplinary Optimization 48 (2013) 607–626.

[16] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.

[17] M. Morris, T. Mitchell, Exploratory designs for computationnal experiments, Journal of Statistical Planning and Inference 43 (1995) 381–402.

[18] K.-T. Fang, R. Li, A. Sudjianto, Design and Modeling for Computer Experiments, Chapman & Hall/CRC, 2006.

[19] R. L. Thorndike, Who belongs in the family?, Psychometrika 18 (1953) 267–276.

[20] L. Kaufman, P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, 1990.

[21] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, K. Hornik, et al., Cluster: cluster analysis basics and extensions, R package version 1 (2012) 56.

[22] J. H. Friedman, Multivariate Adaptive Regression Splines, The Annals of Statistics 19 (1991) 1 – 67.

[23] H. Dette, A. Pepelyshev, Generalized latin hypercube design for computer experiments, Technometrics 52 (2010) 421–429.

[24] A. Marrel, B. Iooss, V. Chabridon, The ICSCREAM methodology: Identification of penalizing configurations in computer experiments using screening and metamodel – Applications in thermal-hydraulics, Nuclear Science and Engineering 196 (2022) 301–321.

[25] C. Ciric, P. Ciffroy, S. Charles, Use of sensitivity analysis to discriminate non-influential and influential parameters within an aquatic ecosystem model, Ecological Modelling 246 (2012) 119–130.

[26] B. Iooss, A.-L. Popelin, G. Blatman, C. Ciric, F. Gamboa, S. Lacaze, M. Lamboni, Some new insights in derivative-based global sensitivity measures, in: Proceedings of the PSAM11 ESREL 2012 Conference, Helsinki, Finland, 2012, pp. 1094–1104.

[27] O. Roustant, F. Gamboa, B. Iooss, Parseval inequalities and lower bounds for variance-based sensitivity indices, Electronic Journal of Statistics 14 (2020) 386–412.

[28] A. G. Journel, The lognormal approach to predicting local distributions of selective mining unit grades, Journal of the International Association for Mathematical Geology 12 (1980) 285–303.

[29] N. Cressie, M. Pavlicová, Lognormal kriging: Bias adjustment and kriging variances, in: O. Leuangthong, C. V. Deutsch (Eds.), Geostatistics Banff 2004, Springer Netherlands, Dordrecht, 2005, pp. 1027–1036.

[30] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning, second ed., Springer, 2009.

[31] M. Gu, J. Palomo, J. Berger, RobustGaSP: Robust Gaussian Stochastic Process Emulation, 2022. URL: https://CRAN.R-project.org/package=RobustGaSP, R package version 0.6.5.

[32] T. Ishigami, T. Homma, An importance quantification technique in uncertainty analysis for computer models, in: Proceedings of the ISUMA'90, First International Symposium on Uncertainty Modelling and Analysis, University of Maryland, USA, 1990.

[33] W. Becker, Metafunctions for benchmarking in sensitivity analysis, Reliability Engineering & System Safety 204 (2020) 107189.