RECOVERING HIDDEN CONSTRAINED SUBSETS BY A GAUSSIAN PROCESS CLASSIFIER ACTIVE LEARNING METHOD BASED ON THE STEPWISE UNCERTAINTY REDUCTION STRATEGY

Morgane Menz, Miguel Munoz Zuniga, Delphine Sinoquet

HIDDEN CONSTRAINTS LEARNING PROBLEM

f: a computer code with inputs $x \in \Omega \subset \mathbb{R}^m$ with simulation failures on Ω.

Objective: determination of the feasible set:

$$\boxed{\Gamma^* = \{x \in \Omega : f(x) \neq NAN\} = \{x \in \Omega : \mathbb{1}_{f(x) \neq NAN} = 1\}}$$

Learning hidden constraints is a binary classification problem:

- We have binary observations: $(\mathcal{X}, \mathcal{Y}) = (x_j, y_j)_{j=1,\dots,n}$, with $y_j = \mathbb{1}_{f(x_j) \neq NAN}$.
- Objective: predict the probability of belonging to the failure/non-failure class

OVERVIEW OF HIDDEN CONSTRAINTS LEARNING STRATEGIES

Expensive numerical simulators

Classifiers

Support Vector Classifier

Naive Bayes Classifier

Gaussian Process Classifier

Neural Networks

K-Nearest Neighbors

Random Forest Classifier

…

Active learning

Uncertainty-based query approaches

Disagreement-based query approaches

Diversity- and density-based approaches

Expected error Reduction / One-step-look-ahead

[Cacciarelli and Kulahci, 2021]

One-step-look-ahead

Expected Error Reduction
[Roy and McCallum, 2001]
[Zhao et al, 2021]

→ Generalization error

OVERVIEW OF HIDDEN CONSTRAINTS LEARNING STRATEGIES

Classifiers

Support Vector Classifier

Naive Bayes Classifier

Gaussian Process Classifier

Neural Networks

K-Nearest Neighbors

Random Forest Classifier

...

Active learning

Uncertainty-based query approaches

Disagreement-based query approaches

Diversity- and density-based approaches

Expected error Reduction / One-step-look-ahead

[Cacciarelli and Kulahci, 2021]

One-step-look-ahead

Expected Error Reduction
[Roy and McCallum, 2001]
[Zhao et al, 2021]
→ Generalization error

Stepwise Uncertainty Reduction strategies ?

LEARNING HIDDEN CONSTRAINT LEARNING WITH GAUSSIAN PROCESS CLASSIFIER

f: a computer code with inputs $x \in \Omega \subset \mathbb{R}^m$ with simulation failures on Ω.

Objective: determination of the feasible set:

$$\boxed{\Gamma^* = \{x \in \Omega : f(x) \neq NAN\} = \{x \in \Omega : \mathbb{1}_{f(x) \neq NAN} = 1\}}$$

Learning hidden constraints is a binary classification problem:

- We have binary observations: $(\mathcal{X}, \mathcal{Y}) = (x_j, y_j)_{j=1,\ldots,n}$, with $y_j = \mathbb{1}_{f(x_j) \neq NAN}$.
- Objective: predict the probability of belonging to the failure/non-failure class

➜ The formulation of the classification model is based on a Gaussian Process (GP) surrogate

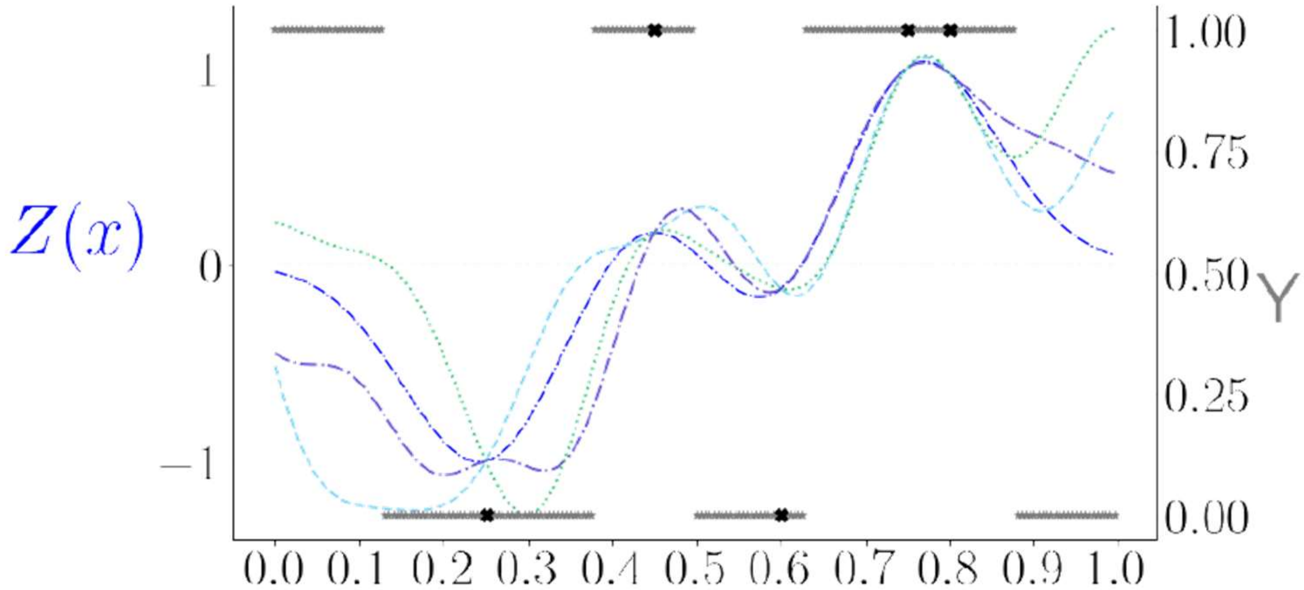
The GPC model allows to predict the probability of success of a simulation:

$$\boxed{p_n(x) = \mathbb{P}[Y_n(x) = 1] = \mathbb{P}[Y(x) = 1 | \mathcal{X}, \mathcal{Y}]}$$

GAUSSIAN PROCESS CLASSIFIER FORMULATION

In [Bachoc et al., 2020], the probability of non-failure is modeled using the sign of a latent GP $Z(x) \sim GP(m_n(x, z_n), k_n(x))$:

$$p_n(x) = \mathbb{P}[\mathbb{1}_{Z(x)>0} = 1 | x, \mathcal{X}, \mathcal{Y}] = \int_{\mathbb{R}^n} \phi_{\mathcal{Y}}^{Z_n}(z_n) \bar{\Phi}\left(\frac{-m_n(x, z_n)}{\sqrt{k_n(x)}}\right) \mathrm{d}z_n$$



GAUSSIAN PROCESS CLASSIFIER FORMULATION

Practical building of the GPC model $p_n(x)$ for any x:

- Optimization of the hyperparameters of the latent GP to maximize the likelihood: $\mathbb{P}[sign(Z_n) = \mathcal{Y}]$
- Generation of realizations $z_n^{(i)}$ of $Z_n | sign(Z_n) = \mathcal{Y}$
  ➜ Approximation of $p_n(x)$:

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

LEARNING HIDDEN CONSTRAINT WITH GPC

The feasible set learned by a GPC model can be characterized by a percentile of the feasibility probability: $\boxed{Q_\alpha = \{x \in \Omega : p_n(x) \geq \alpha\}, \alpha \in (0,1]}$

Objective: Obtaining an accurate approximation of Q_α

Idea: draw a methodology from an existing paradigm used in Gaussian Process Regression active learning for feasible set estimation: stepwise uncertainty reduction strategy using the notion of random set [Bect et al., 2012, Molchanov, 2005]

STEPWISE UNCERTAINTY REDUCTION STRATEGY FOR HIDDEN CONSTRAINTS LEARNING

Let us consider the random set:

$$\boxed{\Gamma = \{x \in \Omega : Y_n(x) = 1\}}$$

with $Y_n(x)$ the conditional Bernoulli [Dai et al., 2013] random variable $Y(x)|\mathcal{X}, \mathrm{Y_n} = \mathcal{Y}$

The SUR strategy based on the uncertainty on Γ defined by the **vorob'ev deviation** Var_n^{Γ} [Chevalier, 2013, El Amri et al., 2020, Vorobyev and Lukyanova, 2013] aim to minimize at each step the following learning function:

$$\boxed{J_n(x_{n+1}) = \mathbb{E}_n\left[Var_{n+1}^{\Gamma}(x_{n+1}, Z(x_{n+1}))\right]}$$

STEPWISE UNCERTAINTY REDUCTION STRATEGY FOR HIDDEN CONSTRAINTS LEARNING

We propose to minimize the expectation of uncertainty using the expectation on the Bernoulli process $\mathrm{Y_n}$

$$\mathbb{E}_n[\cdot] = \mathbb{E}_{Y_n(x_{n+1})}[\cdot|\mathcal{X}, \mathrm{Y_n} = \mathcal{Y}]$$

Hence:

$$\boxed{\begin{aligned} J_n(x_{n+1}) &= (1 - p_n(x_{n+1}))\ Var^{\Gamma}_{n+1}(x_{n+1}, Y_n(x_{n+1}) = 0) \\ &\quad + p_n(x_{n+1})\ Var^{\Gamma}_{n+1}(x_{n+1}, Y_n(x_{n+1}) = 1) \end{aligned}}$$

➜ Advantage: no integration over the realizations of the latent GP $Z(x_{n+1})$

➜ Simplified computation of $Var^{\Gamma}_{n+1}(x_{n+1}, Y_n(x_{n+1}) = y_{n+1})$ with our proposed GPC update formulae

STEPWISE UNCERTAINTY REDUCTION STRATEGY FOR HIDDEN CONSTRAINTS LEARNING

The Vorob'ev deviation is given by:

$$Var^{\Gamma}_{n+1}(x_{n+1}, Y_n(x_{n+1}) = y_{n+1}) = \int (1-p_{n+1}(x))\mathbb{1}_{p_{n+1}(x)\geq\alpha^*} + p_{n+1}(x)\mathbb{1}_{p_{n+1}(x)<\alpha^*}\mu(dx)$$

with $p_{n+1}(x)$ the future feasibility probability knowing $Y_n(x_{n+1}) = y_{n+1}$ given by:

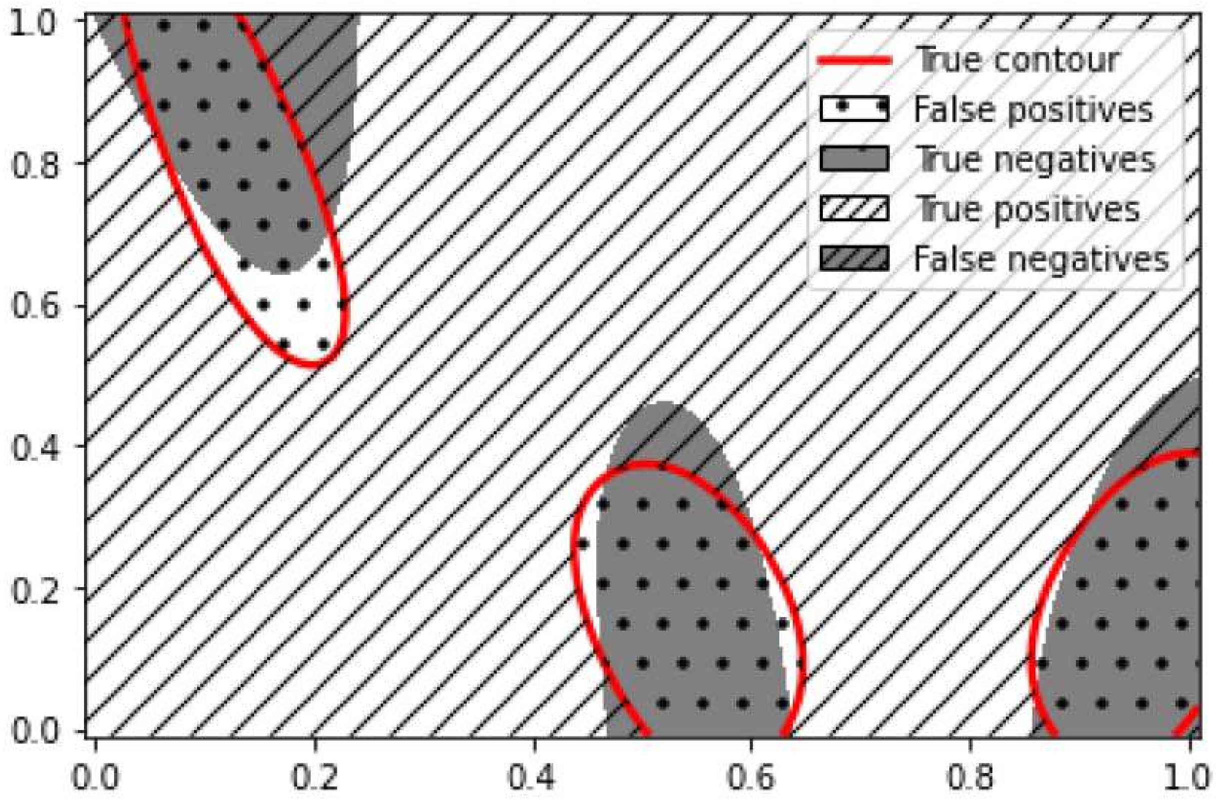
$$p_{n+1}(x) = \frac{1}{N}\sum_{i=1}^{N}\bar{\Phi}\left(\frac{-m_{n+1}(x, \mathrm{z}^{\mathrm{i}}_{\mathrm{n+1}})}{\sigma_{n+1}(x)}\right)$$

This probability can be computed by:

- using GP update formulae provided in [Chevalier, 2013]
- sampling $\mathrm{z}^{\mathrm{i}}_{\mathrm{n+1}}$ using $\mathrm{z}^{\mathrm{i}}_{\mathrm{n}}$ and:

$z(x_{n+1})^i \sim \mathcal{N}\left(m_n(x_{n+1}, \mathrm{z}^{\mathrm{i}}_{\mathrm{n}}), k_n(x_{n+1}, x_{n+1})\right)$ truncated such that $sign(z(x_{n+1})^i) = y_{n+1}$

RESULTS: COMPARISON CRITERIA



$$crit_F = \frac{FN + FP}{P} = \frac{\mu(\Gamma^* \Delta Q_{\alpha^*})}{\mu(\Gamma^*)}$$

RESULTS: COMPARISON OF DIFFERENT ENRICHMENT CRITERIA

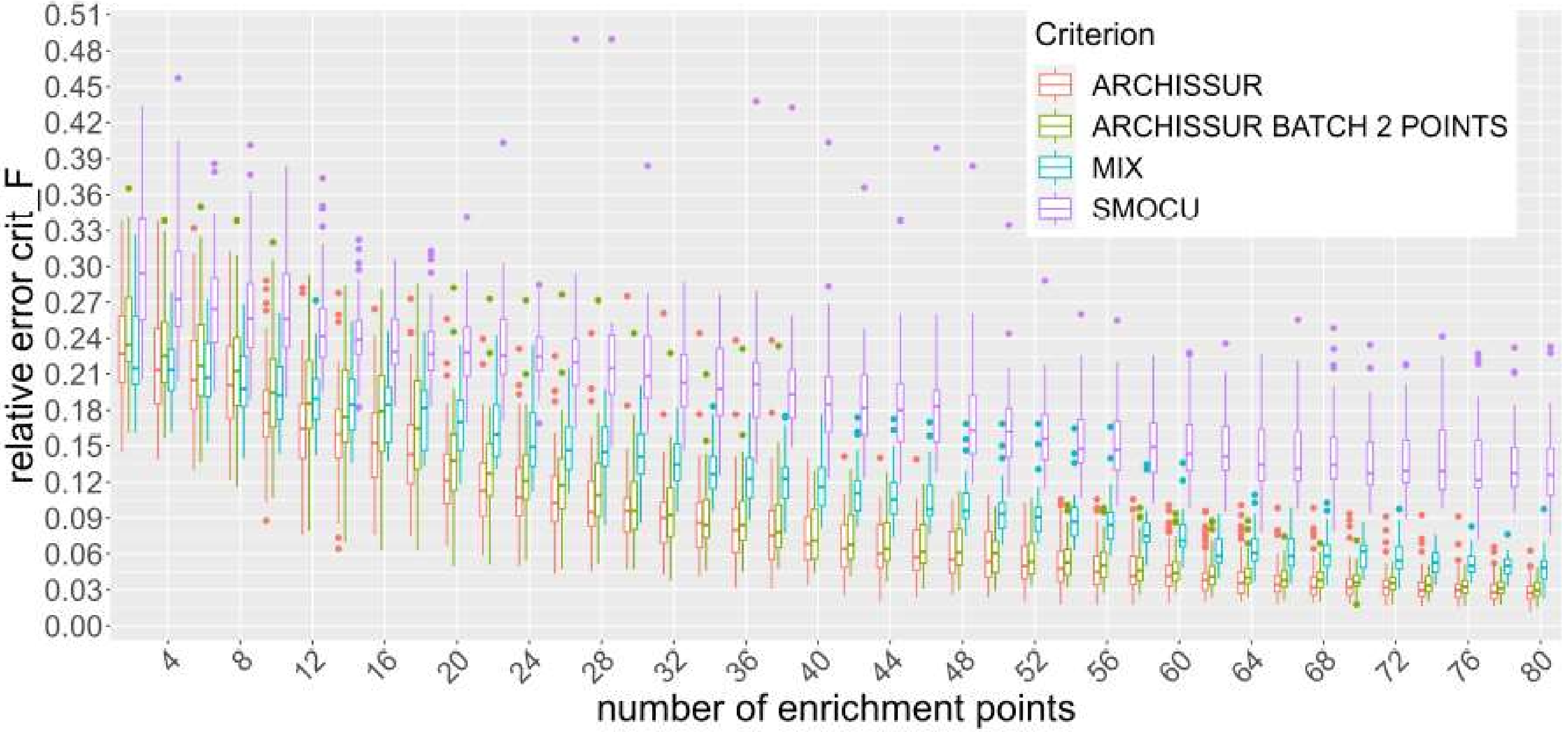
Compared strategies:

- ARCHISSUR: An Active Recovery of a Constrained and Hidden Subset by SUR method based on the Vorob'ev deviation $\Rightarrow$ one point and batch versions [Menz et al.,2023]
- Mix enrichment criterion: add the point corresponding to the maximum of the GP variance (exploration) and the one where $p_n(x)$ value is the closest to $\frac{1}{2}$ (exploitation) simultaneously
- SMOCU enrichment measure: Soft-MOCU (Mean Objective Cost of Uncertainty) method [Zhao et al., 2021] `https://github.com/QianLab/Soft_MOCU`

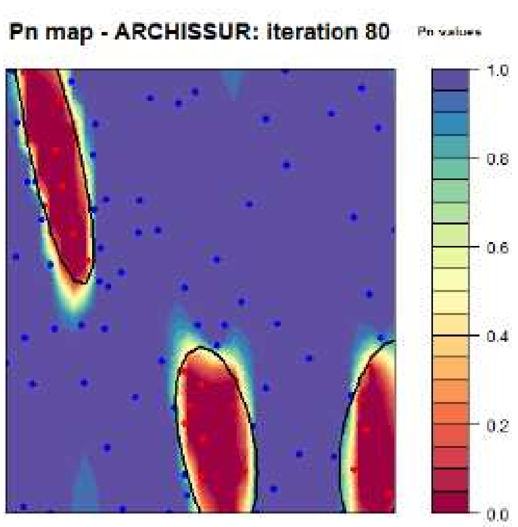
 |

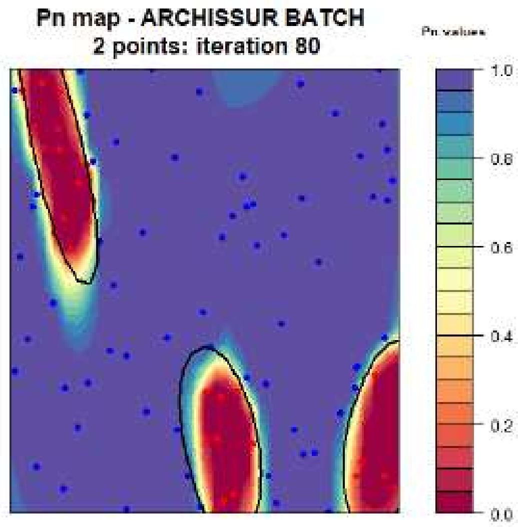
TOY EXAMPLE BASED ON BRANIN

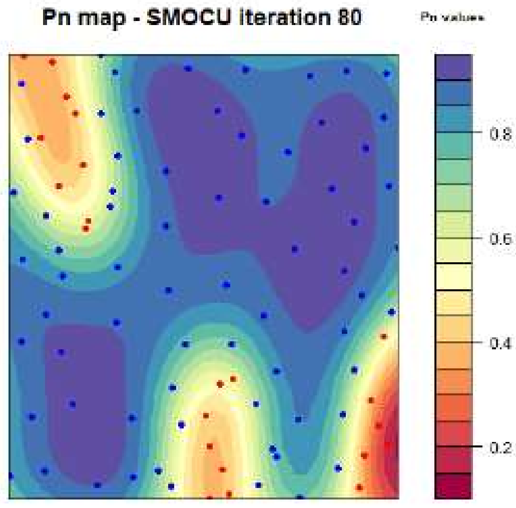
Evolution of $crit_F$ for MIX, ARCHISSUR, ARCHISSUR BATCH and SMOCU



TOY EXAMPLE BASED ON BRANIN







TOY EXAMPLE BASED ON BRANIN

Mean CPU times (in seconds) of the learning function evaluation on 20 repetitions with 2000 integration points

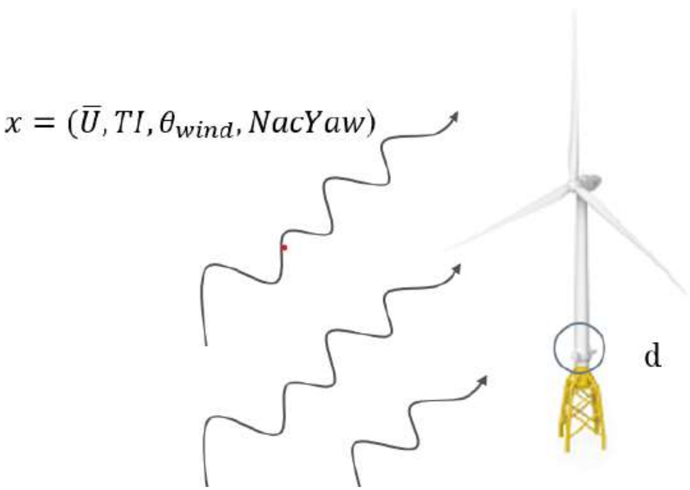
Method	ARCHISSUR with old formulation	ARCHISSUR	ARCHISSUR BATCH (2 points)	SMOCU
	0.311	0.079	0.412	0.849

* old formulation: $\mathbb{E}_n[\cdot] = \mathbb{E}_{Z_n(x_{n+1})}[\cdot|\mathcal{X}, \mathrm{Z_n} = z_n]$

DAMAGE PREDICTION OF A WIND TURBINE

Wind loads are described by 3 parameters:

- wind speed $\bar{U}$,
- turbulence intensity TI,
- misalignment angle $NacYaw$.



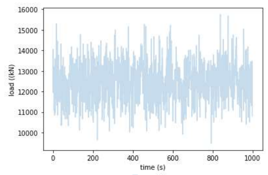
Wind parameters: $\bar{U}$, TI, $NacYaw$

↓

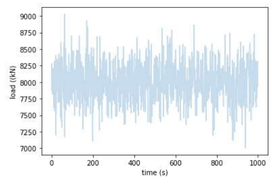
Turbsim

↓

n_{seed} simulations of wind



…



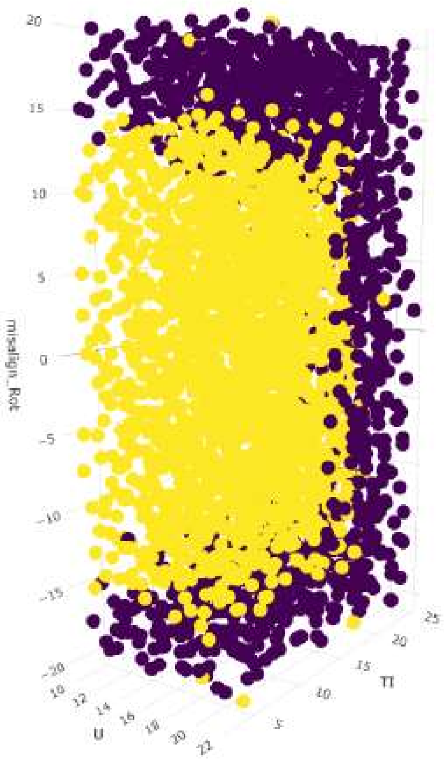
↓

Multiphysics simulator **OpenFast** … Multiphysics simulator **OpenFast**

↓

Mean damage across wind turbine tower base

DAMAGE PREDICTION OF A WIND TURBINE



failure / success

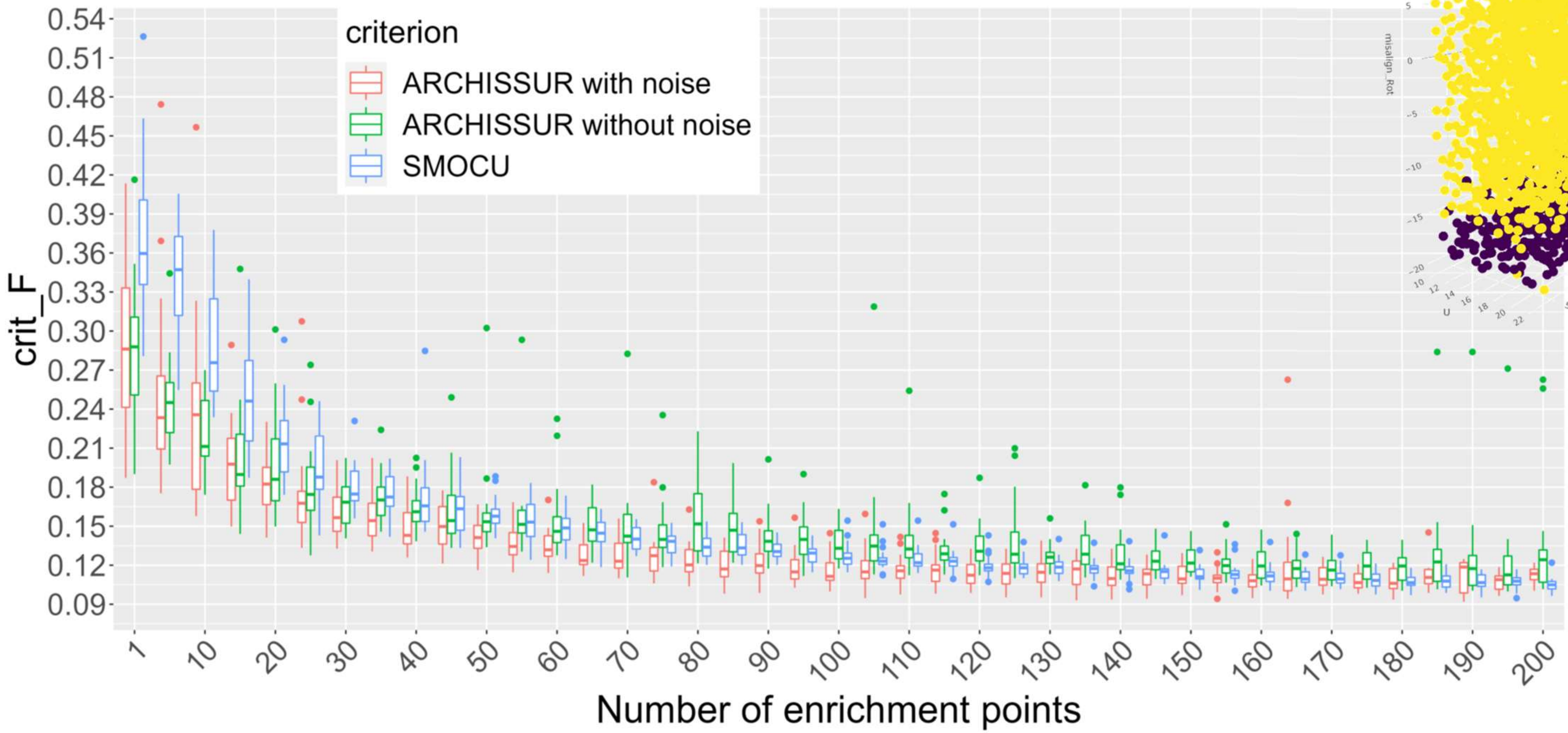
Specificity of this test case:

- non-regular frontier between feasible/unfeasible sets
- Presence of outliers

→ GPC model with noise on observed points

RESULTS FOR THE DAMAGE PREDICTION OF A WIND TURBINE

Evolution of $crit_F$ for ARCHISSUR and SMOCU



failure / success

CONCLUSIONS AND OUTLOOK

- Great potential to learn hidden constraints with application to an industrial test case
- Application on a analytic case in 10 dimensions tackled in the preprint
- Faster decrease of the error measure as a function of number of simulation on the test cases

→Can this work be integrated within space-filling or optimization strategies to help in the presence of hidden constraints?

Innover les énergies

Retrouvez-nous sur :

www.ifpenergiesnouvelles.fr

@IFPENinnovation

Energies nouvelles

REFERENCES I

- Charles Audet and John E Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. SIAM Journal on optimization, 17(1):188–217, 2006.
- Charles Audet, Sébastien Le Digabel, Viviane Rochon Montplaisir, and Christophe Tribes. Nomad version 4: Nonlinear optimization with the mads algorithm.
- Yves Auffray, Pierre Barbillon, and Jean-Michel Marin. Maximin design on non hypercube domains and kernel interpolation. Statistics and Computing, 22:703–712, 2012.
- François Bachoc, Céline Helbert, and Victor Picheny. Gaussian process optimization with failures: classification and convergence proof. Journal of Global Optimization, 78(3):483–506, November 2020.
  Julien Bect, David Ginsbourger, Ling Li, Victor Picheny, and Emmanuel Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. Statistics and Computing, 22(3):773–793, May 2012.
  Clément Chevalier. Fast uncertainty reduction strategies relying on Gaussian process models. PhD Thesis, 2013.
- Bin Dai, Shilin Ding, and Grace Wahba. Multivariate Bernoulli distribution. Bernoulli, 19(4):1465 – 1483, 2013.
- Sébastien Le Digabel and Stefan M Wild. A taxonomy of constraints in simulation-based optimization.
- Mohamed Reda El Amri, Céline Helbert, Olivier Lepreux, Miguel Munoz Zuniga, Clémentine Prieur, and Delphine Sinoquet. Data-driven stochastic inversion via functional quantization. Statistics and Computing, 30(3):525–541, 2020.

REFERENCES II

- Chaofan Huang, V Roshan Joseph, and Douglas M Ray. Constrained minimum energy designs. Statistics and Computing, 31:1–15, 2021.
- Hoël Langouët. Optimisation sans dérivées sous contraintes : deux applications industrielles en ingénierie de réservoir et en calibration des moteurs. Theses, Université Nice Sophia Antipolis, June 2011.
- Morgane Menz, Miguel Munoz-Zuniga, and Delphine Sinoquet. Learning hidden constraints using a Stepwise Uncertainty Reduction strategy based on Gaussian Process Classifiers. working paper or preprint, November 2022.
- Ilya Molchanov. Theory of random sets, volume 19. Springer, 2005.
- Michael JD Powell. The newuoa software for unconstrained optimization without derivatives. Large-scale nonlinear optimization, pages 255–297, 2006.
- Oleg Yu. Vorobyev and Natalia A. Lukyanova. A Mean Probability Event for a Set of Events. Journal of Siberian Federal University. Mathematics and Physics., page 128–136, 2013.
- Guang Zhao, Edward R Dougherty, Byung-Jun Yoon, Francis J Alexander, and Xiaoning Qian. Efficient Active Learning for Gaussian Process Classification by Error Reduction. page 13, 2021.
- Huang. Chaofan. V. Roshan Joseph. et Douglas M. Ray. « Constrained Minimum Energy Designs ». Statistics and Computing 31. no 6 (novembre 2021): 80.

The Vorob'ev deviation is given by:

$$Var^{\Gamma}_{n+1}(x_{n+1}, Y_n(x_{n+1}) = y_{n+1}) = \int (1 - p_{n+1}(x))\mathbb{1}_{p_{n+1}(x)\geq\alpha^*} + p_{n+1}(x)\mathbb{1}_{p_{n+1}(x)<\alpha^*}\mu(dx)$$

with $p_{n+1}(x)$ the future feasibility probability knowing $Y_n(x_{n+1}) = y_{n+1}$ given by:

$$p_{n+1}(x) = \frac{1}{N}\sum_{i=1}^{N}\bar{\Phi}\left(\frac{-m_{n+1}(x, \mathrm{z}^{\mathrm{i}}_{\mathrm{n+1}})}{\sigma_{n+1}(x)}\right)$$

This probability can be computed by:

- using GP update formulae provided in [Chevalier, 2013]
- sampling $\mathrm{z}^{\mathrm{i}}_{\mathrm{n+1}}$ using $\mathrm{z}^{\mathrm{i}}_{\mathrm{n}}$ and:

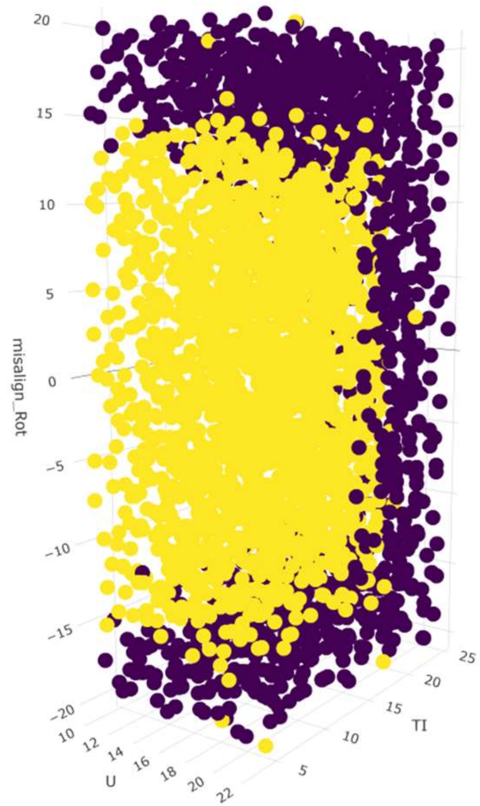
$z(x_{n+1})^i \sim \mathcal{N}\left(m_n(x_{n+1}, \mathrm{z}^{\mathrm{i}}_{\mathrm{n}}), k_n(x_{n+1}, x_{n+1})\right)$ truncated such that $sign(z(x_{n+1})^i) = y_{n+1}$

APPLICATION IFPEN : DOMMAGE EN PIED DE TOUR D'UNE ÉOLIENNE OFFSHORE

Climat, Environnement et Économie circulaire

Failure for at least one of the wind seeds n_{seed} = simulation failure

failure / success



Vorob'ev expectation and deviation

Definition

Vorob'ev expectation is defined as the α^*-percentile of Γ, where α^* is determined by:

$$\mathbb{E}[\mu(\Gamma)] = \mu(Q_{\alpha^*})$$

with μ the Lebesgue measure and $Q_\alpha = \{x \in \Omega : p_n(x) \geq \alpha\}, \alpha \in (0, 1]$.

Vorob'ev expectation is a global minimiser of the vorob'ev deviation $Var_n(\Gamma)$ among closed sets of volume equal to the mean volume of Γ:

$$Var_n(\Gamma) = \mathbb{E}[\mu(Q_\alpha \Delta \Gamma) | \mathcal{X}, \mathcal{Y}]$$

with $\Gamma \Delta Q_\alpha = (\Gamma \setminus Q_\alpha) \cup (Q_\alpha \setminus \Gamma)$

MOCU/SMOCU

The enrichment criterion based on MOCU (Mean Objective Cost of Uncertainty) formulation given in [Zhao et al., 2021] can be rewritten as:

$$U^{MOCU}(x_{n+1}) = \mathbb{E}_n[Var_{n+1}(\Gamma)] - Var_n(\Gamma) \text{ with } \alpha = 0.5$$

Soft-MOCU learning function is a smooth concave approximation of MOCU given by:

$$\begin{aligned} U^{SMOCU}(x_{n+1}) &= \mathbb{E}_X[\mathbb{E}_n[\tfrac{1}{k} ln(exp(k * p_{n+1}(X)) + exp(k(1 - p_{n+1}(X))))] \\ &- \tfrac{1}{k} ln(exp(k * p_n(X)) + exp(k(1 - p_n(X))))] \end{aligned}$$

with k a smoothness parameter.

