Learning hidden constraints with gaussian process classifiers in the optimization context

Delphine Sinoquet
with Morgane Menz, Miguel Munoz Zuniga

IFP Énergies Nouvelles

CONTEXT : ROBUST/RELIABLE CONCEPTION OF COMPLEX SYSTEMS

Uncertain (random) variables

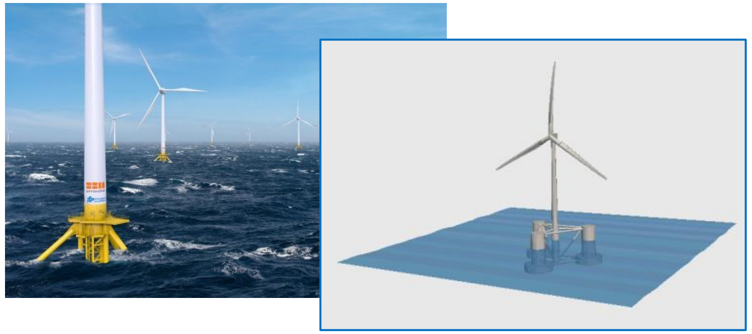
u

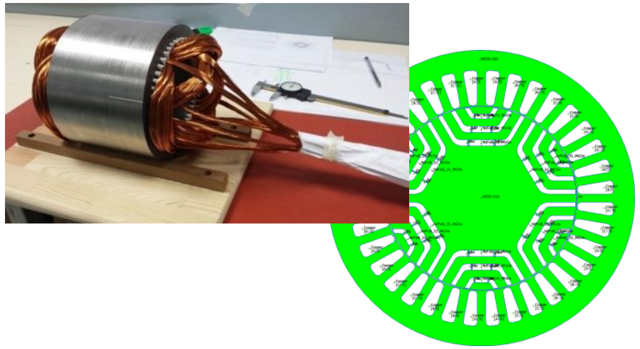
« Controllable » variables
= design variables x

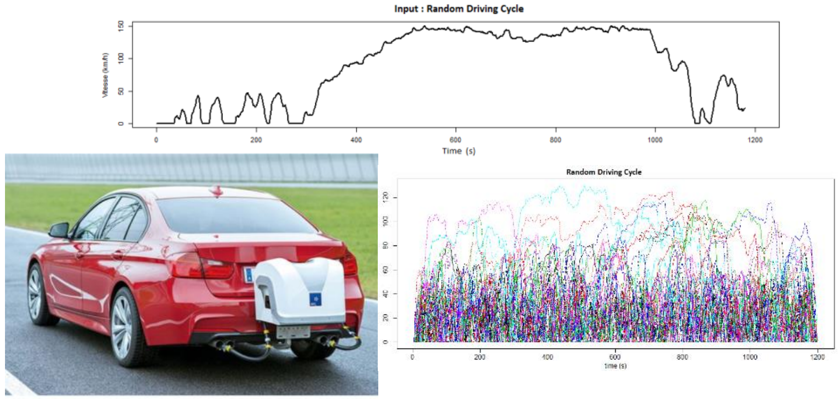
Complex and time consuming black-box simulator

$f(x,u)$
$g(x,u)$ Quantities of interest

minimize/maximize
constrain







HIDDEN CONSTRAINTS

- Crashes or instabilities of the black-box simulator *e.g.* due to convergence issues
- Often, simulation failures are computationally expensive
- And they make the optimization convergence tricky

➔ **Learn hidden constraint from a limited number of “costly” simulations**

OUTLINE

- Active learning of feasible input set for complex simulators
  hidden constraint leading to simulation crashes

- Optimization with hidden constraint
  with derivative free trust region optimization method

OUTLINE

- Active learning of feasible input set for complex simulators
  hidden constraint leading to simulation crashes

- Optimization with hidden constraint
  with derivative free trust region optimization method

PROBLEM STATEMENT

f: output of a black-box simulator with inputs $x \in \Omega \subset \mathbb{R}^m$

Our objective is to determine the feasible set (no simulation crash)

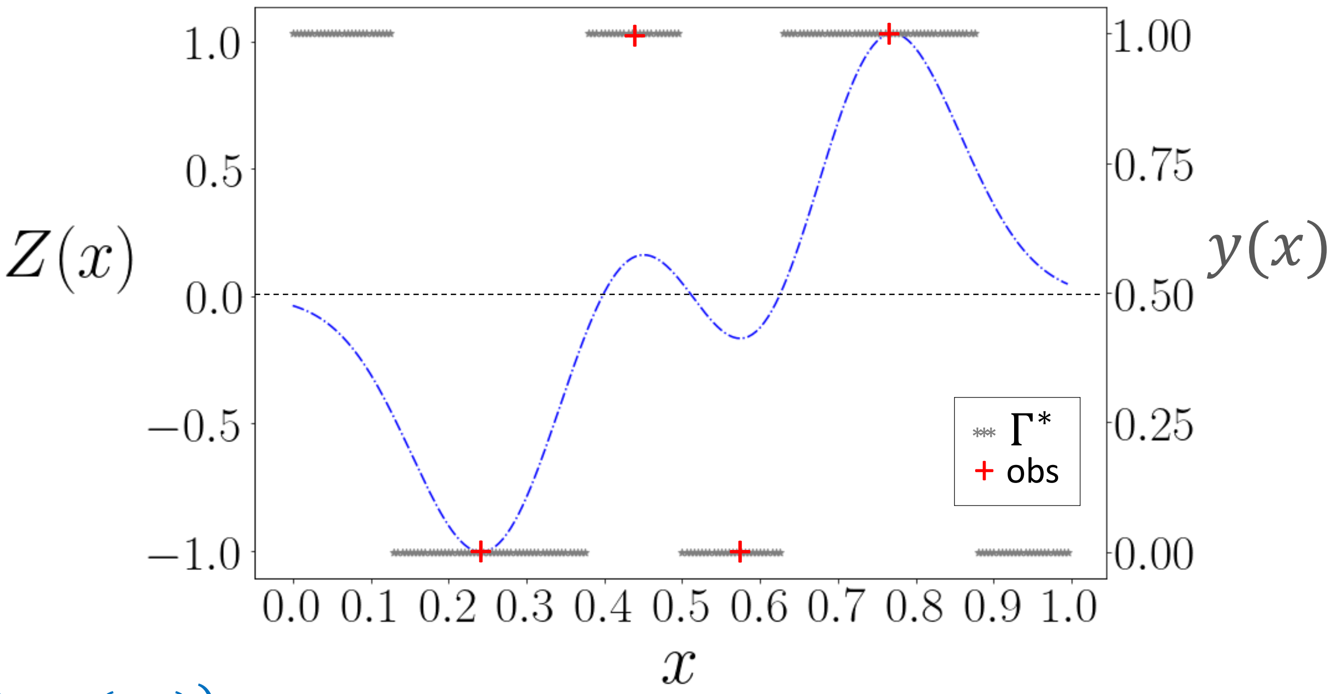
$$\Gamma^* = \{x \in \Omega : f(x) \neq NAN\} = \{x \in \Omega : \mathbb{1}_{f(x) \neq NAN} = 1\}$$

This is a binary classification problem
with binary observations $(\mathcal{X}, \mathcal{Y}) = (x_j, y_j)_{j=1,\dots,n}$ with $y_j = \mathbb{1}_{f(x_j) \neq NAN}$
which aims to predict the probability of belonging to the failure/non-failure class

→ Our choice: a classification model based on a Gaussian Process (GP)

GAUSSIAN PROCESS CLASSIFIER (GPC)

A GPC is based on a latent GP Z conditioned on the sign observations characterizing the belonging to a class ($Z_n = Z(x_1), \dots, Z(x_n)$ are not available) [Bachoc et al, 2020]



$Z(x) \sim GP\left(m_n(.), k_n(.,.)\right)$
$m_n(.), k_n(.,.)$ conditioned mean and kernel of $Z(x)$

GAUSSIAN PROCESS CLASSIFIER (GPC)

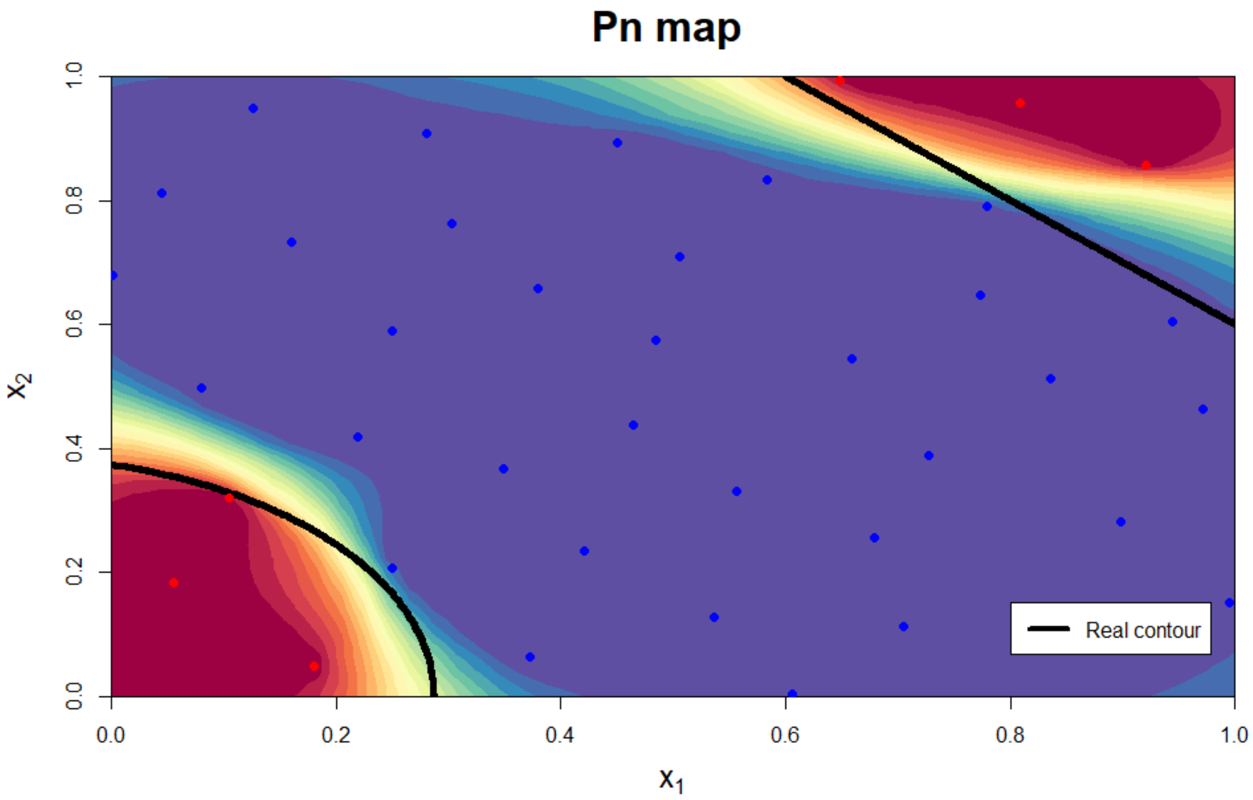
The GPC model allows to predict the probability of non-failure of a simulation

$$p_n(x) = \mathbb{P}[Y_n(x) = 1] = \mathbb{P}[\, Y(x) = 1 \mid \mathcal{X}, \mathcal{Y} \,]$$
$$= \mathbb{P}\left[\, \mathbb{1}_{Z(x)>0} = 1 \mid x, \mathcal{X}, \mathcal{Y} \,\right]$$

GAUSSIAN PROCESS CLASSIFIER (GPC)

The GPC model allows to predict the probability of non-failure of a simulation

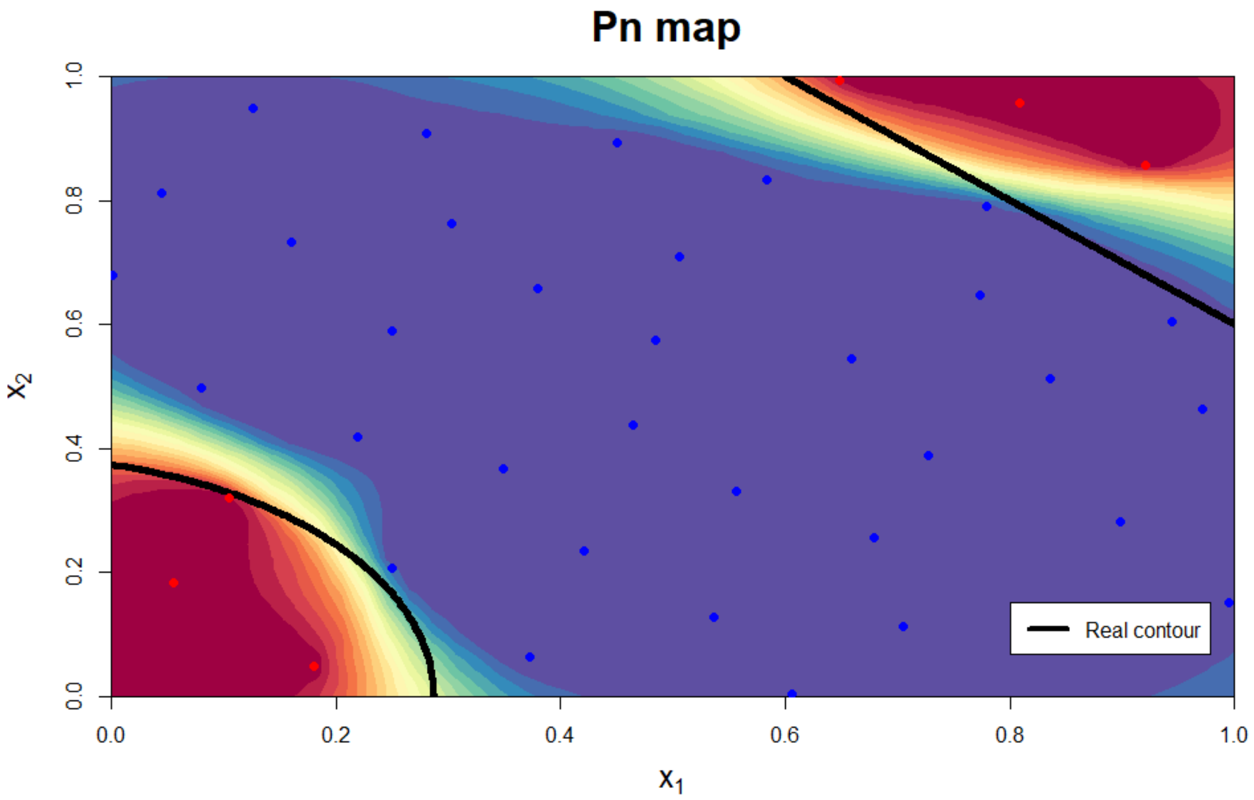
$$p_n(x) = \mathbb{P}[Y_n(x) = 1] = \mathbb{P}[\,Y(x) = 1 \mid \mathcal{X}, \mathcal{Y}\,]$$
$$= \mathbb{P}\left[\,\mathbb{1}_{Z(x)>0} = 1 \mid x, \mathcal{X}, \mathcal{Y}\,\right]$$



GAUSSIAN PROCESS CLASSIFIER (GPC)

Characterization of the feasible set by quantiles

$$Q_\alpha = \{x \in \Omega : p_n(x) \geq \alpha\}, \alpha \in (0, 1]$$



ARCHISSUR STRATEGY: ACTIVE LEARNING OF FEASIBLE SET

Stepwise Uncertainty Reduction (SUR) strategy

Sequential choice of additional simulation point(s) x_{n+1} in order to minimize the *future* uncertainty on the feasible set [Bect et al., 2012, Molchanov, 2005]

$$\min_{x_{n+1}} J_n(x_{n+1}) := \mathbb{E}_n[\mathrm{Var}_{n+1}(\Gamma)]$$

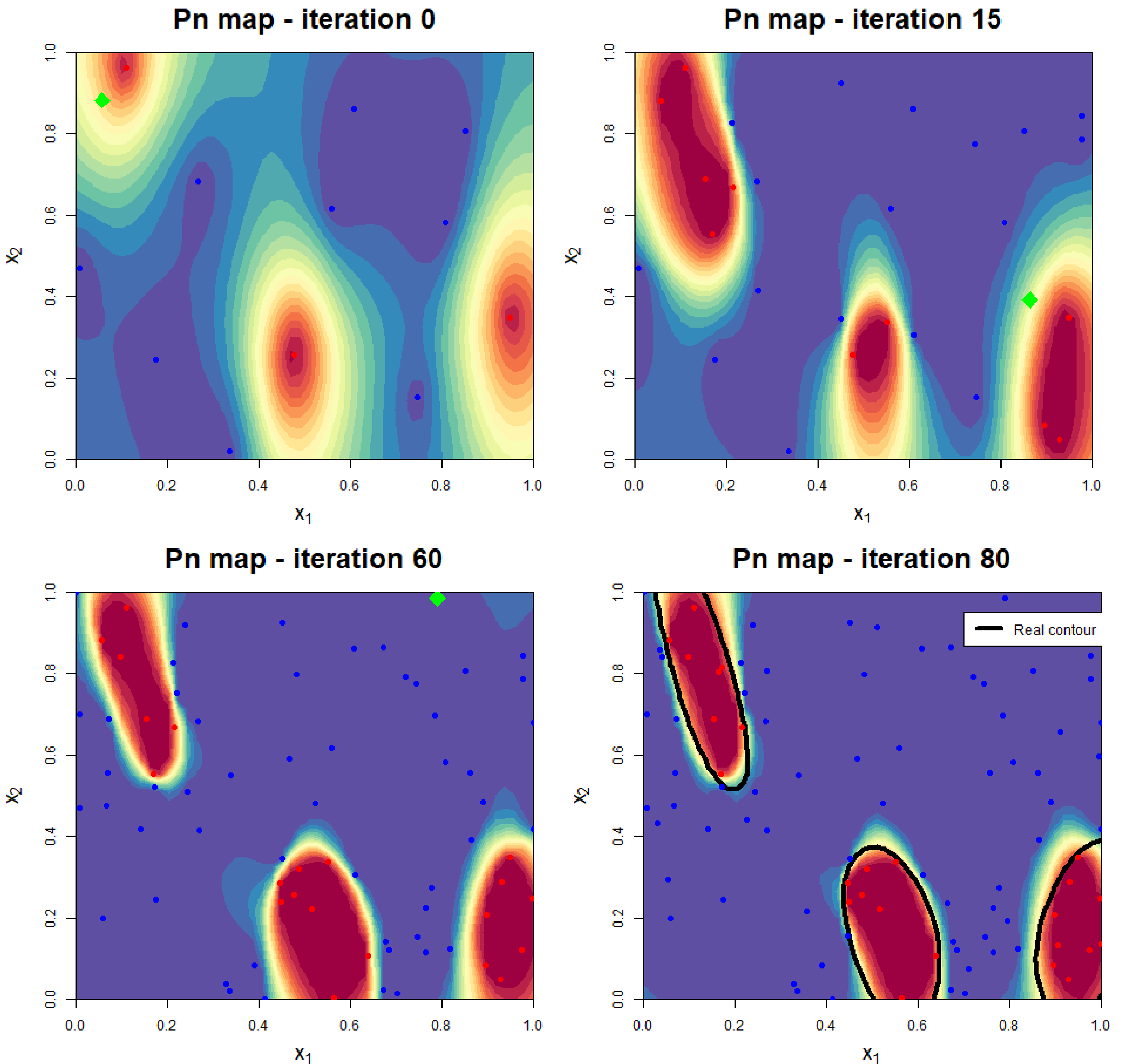
with $\mathrm{Var}_n(\Gamma)$, the Vorob'ev deviation (variance of the feasible set) computed from the current GP model Z_n. [Chevalier, 2013, El Amri et al., 2021, Vorob'ev and Lukyanova, 2013].

→**ARCHISSUR method:** *Active Recovery of Constrained and Hidden Subset by SUR*

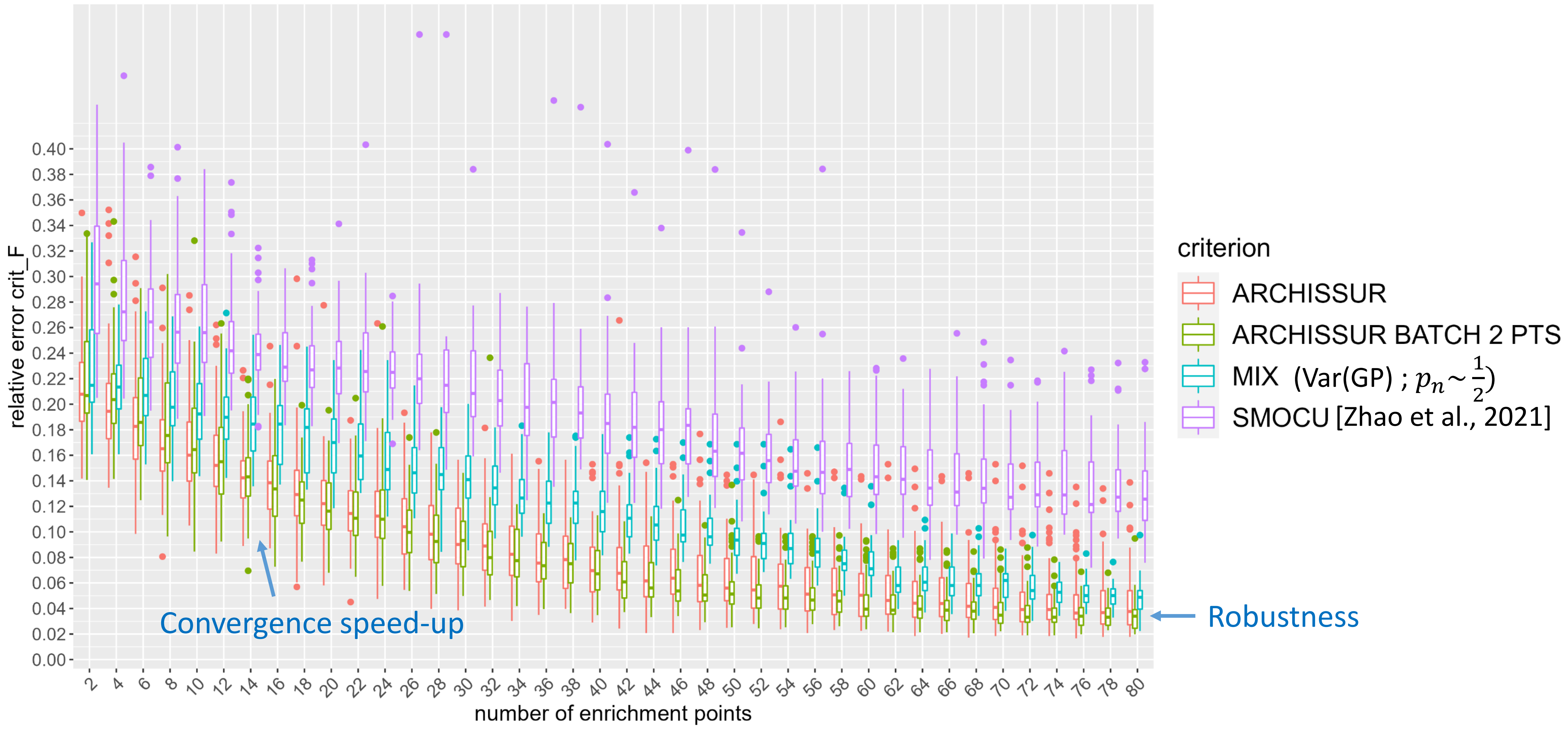
[Menz et al, hal-03688224]

A 2D EXAMPLE (Branin function)

Intensification and exploration ability of Archissur criterion

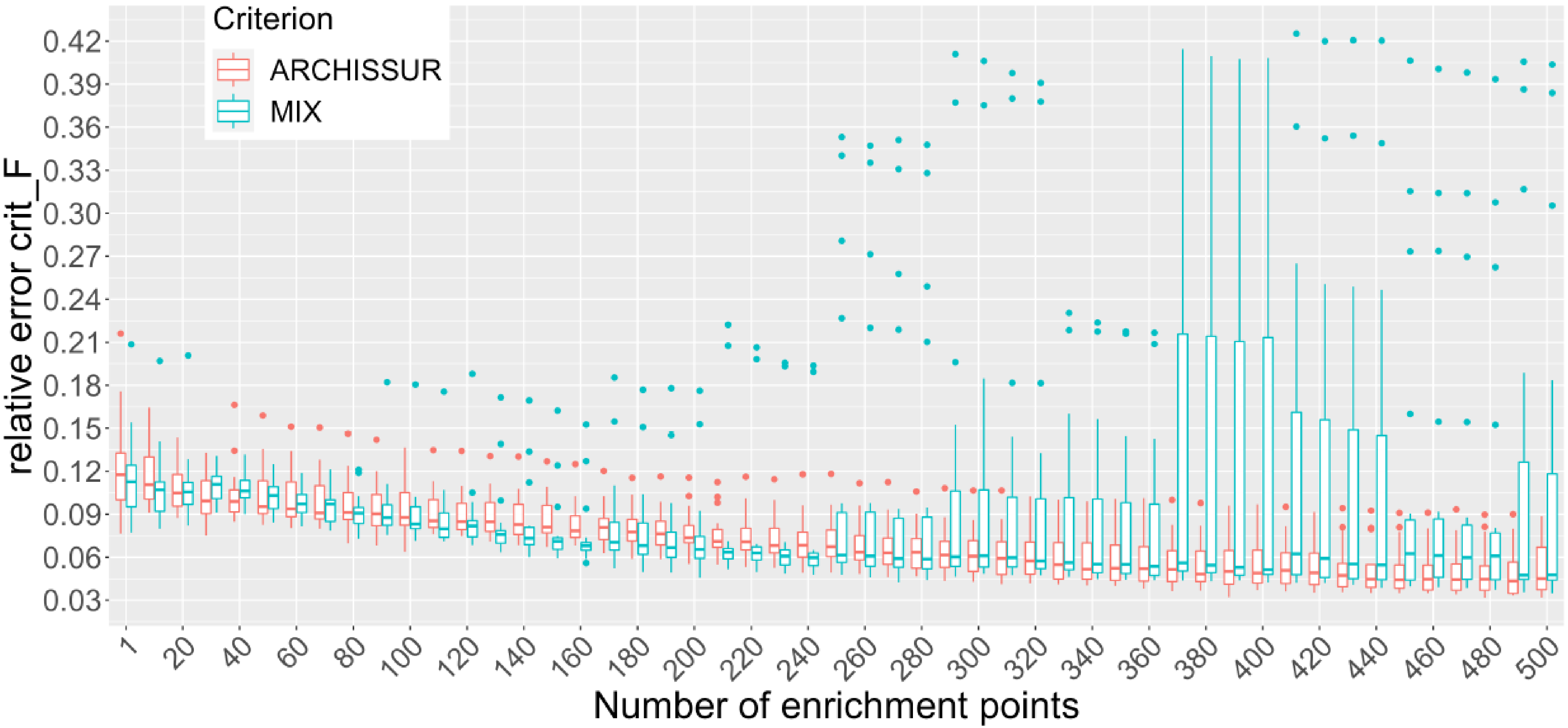


RESULTS ON BRANIN FUNCTION (2D) - 80 initial DOE of 12 points



RESULTS FOR A PROBLEM WITH 10 VARIABLES - 80 initial DOE of 60 points

Feasible set ~ 86% of the total domain



OUTLINE

- Active learning of feasible input set for complex simulators
  hidden constraints leading to simulation crashes

- Optimization with hidden constraint
  with derivative-free trust region optimization method
  with direct search method (NOMAD) → talk of S. Jacquet (MS245)

DERIVATIVE FREE TRUST REGION OPTIMIZATION METHOD IN A NUTSHELL

SQA : **S**equential **Q**uadratic **A**pproximation [Langouët, 2011]

= extension of NEWUOA (Powell, 2007) to constrained optimization

$$\min_{x} f(x)$$
$$\text{s.t.} \begin{cases} l \leq x \leq u & \\ C_{DB}(x) \leq 0 & \text{derivative based constraints} \\ C_{DF}(x) \leq 0 & \text{derivative free constraints} \end{cases}$$

- **Constrained sub-problems** in the trust region of size Δ_k

$$\min_{\|d\| \leq \Delta_k} Q_k(d) \quad \text{s.t.} \begin{cases} C_{DB}(x_k + d) \leq 0 \\ Q_{C_{DF_k}}(d) \leq 0 \end{cases}$$

- Q_k and $Q_{C_{DF_k}}$ are **quadratic interpolation models** of f and C_{DF} (black-box outputs)

OPTIMIZING WITH HIDDEN CONSTRAINTS

- Naïve approach

  In case of a simulator crash: replace the NaN outputs by « surrogate » values

  - Maximal value of the objective functions associated with close points
    in order to avoid a further exploration of this “risky” area

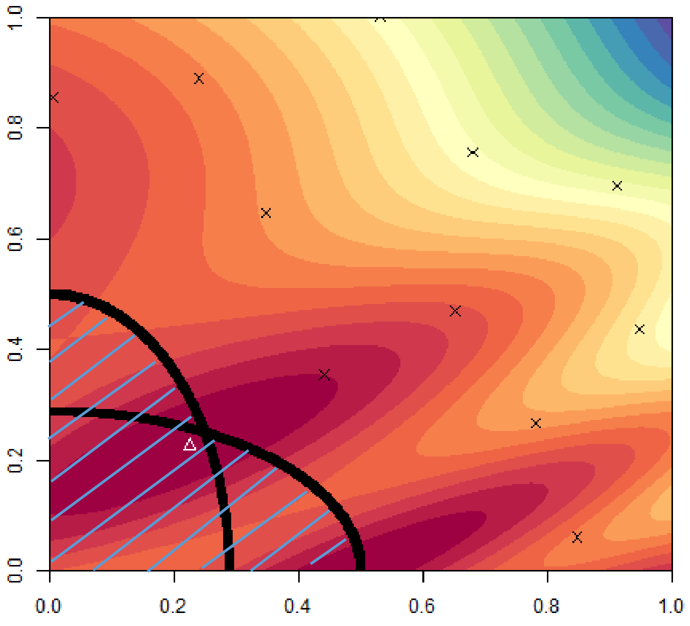
- Our proposal

  - Learn (and update) a GPC model from available simulations during the optimization iterations
    → $\hat{p}_n(x)$: probability of simulation success at iteration n
  - Apply two different strategies to integrate the hidden constraint model in the optimization

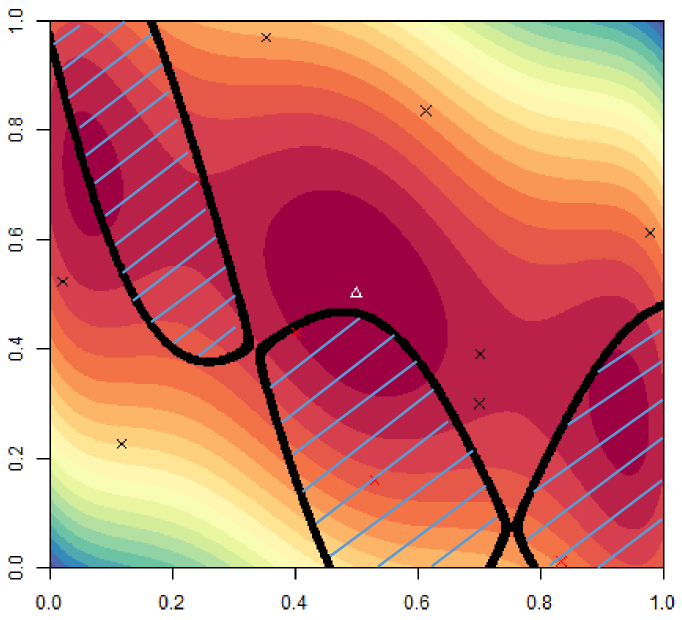
    1. **Prior constraint :** do not simulate the point in case of a high probability of crash $\hat{p}_n(x) < \frac{1}{2}$
    2. **Additional constraint $\hat{p}_n(x) \geq \frac{1}{2}$** as a derivative based constraint (cheap to evaluate)

NUMERICAL TESTS

1)



2)



3) Function in dimension 7 [Sacher et al, 2018]

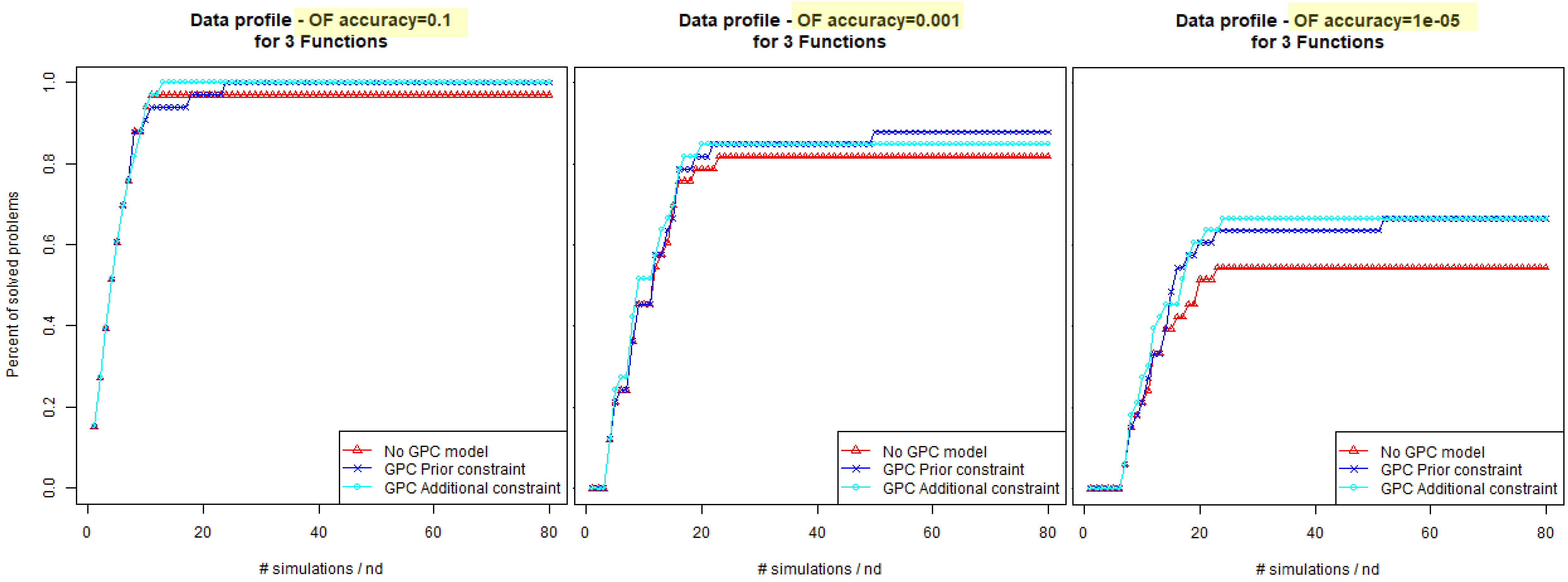
$$\min_{\mathbf{x}\in\Omega_{\text{adm}}} f(\mathbf{x}) = (x_1-10)^2+5(x_2-12)^2+x_3^4+3(x_4-11)^2+10x_5^6+7x_6^2+x_7^4-4x_6x_7-10x_6-8x_7$$

$$\Omega_{\text{adm}} = \{\mathrm{x}\in\Omega, c_{i=1,\dots,4}(\mathrm{x})\leq 0\}$$

$$c_1(\mathrm{x}) = 2x_1^2+3x_2^4+x_3+4x_4^2+5x_5-127, c_2(\mathrm{x}) = 7x_1+3x_2+10x_3^2+x_4-x_5-282$$

$$c_3(\mathrm{x}) = 23x_1+x_2^2+6x_6^2-8x_7-196, c_4(\mathrm{x}) = 4x_1^2+x_2^2-3x_1x_2+2x_3^2+5x_6-11x_7$$

NUMERICAL RESULTS FOR 3 FUNCTIONS AND MULTIPLE INITIAL POINTS



CONCLUSIONS

- Active learning Archissur method has a good potential to learn disconnected feasible sets [Menz et al, hal-03688224]
- The GPC model of hidden constraint is useful in the optimization context to help and speed-up convergence

On-going work

- Coupling Archissur with optimization: use not only GPC model but also active learning strategy

Future work

- Comparison with other approaches:
  e.g., Bayesian Optimization coupled with SVM: EGO-LS-SVM [Sacher et al, 2018]
- Application to the reliability-based design optimization of a wind turbine

Innovating for energy

Find us on:

www.ifpenergiesnouvelles.com

@IFPENinnovation

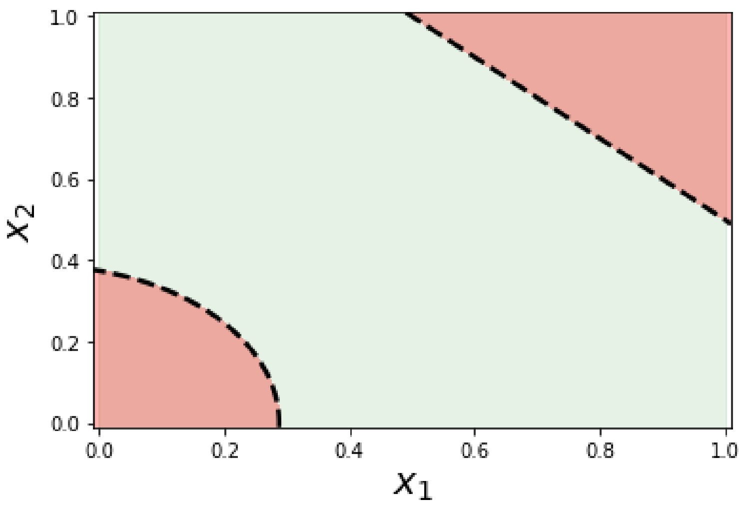
delphine.sinoquet@ifpen.fr

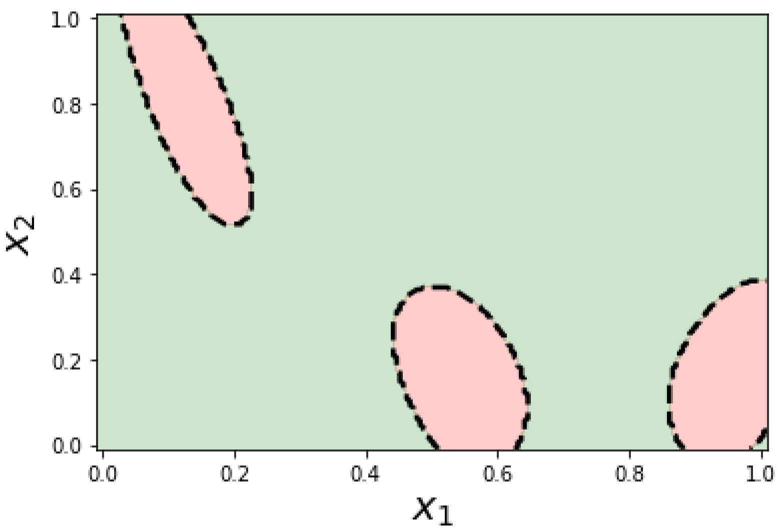
https://www.ifpenergiesnouvelles.fr/page/delphine-sinoquet

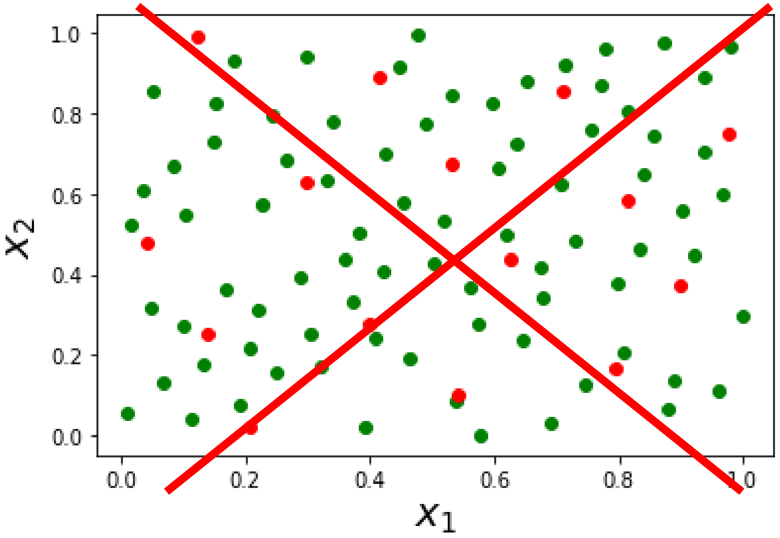
HIDDEN CONSTRAINTS

- Crashes or instabilities of the black-box simulator *e.g.* due to convergence issues
- Often, simulation failures are computationally expensive
- And they make the optimization convergence tricky

➔ **Learn hidden constraint from a limited number of “costly” simulations**







GAUSSIAN PROCESS CLASSIFIER (GPC) FORMULATION

The GPC model allows to predict the probability of non-failure of a simulation

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

The probability $p_n(x)$ is modeled on the basis of [Bachoc et al, 2020] by using the sign of the latent GP Z

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

with $\phi_{\mathcal{Y}}^{Zn}(z_n)$ the conditioned p.d.f of Z_n truncated to respect $sign(Z_n) = \mathcal{Y}$, and

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

where Φ is the c.d.f of the normal standard distribution.

GAUSSIAN PROCESS CLASSIFIER (GPC) 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 $p_n(x)$:

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

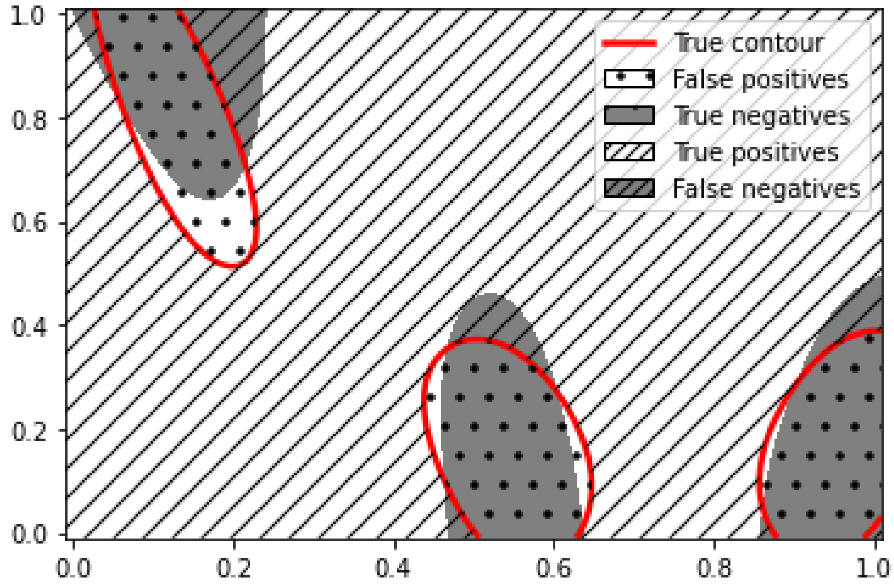
COMPARISON OF DIFFERENT ENRICHMENT CRITERIA

Compared strategies

- ARCHISSUR criterion: *Active Recovery of Constrained and Hidden Subset by SUR*
- Mixed 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]

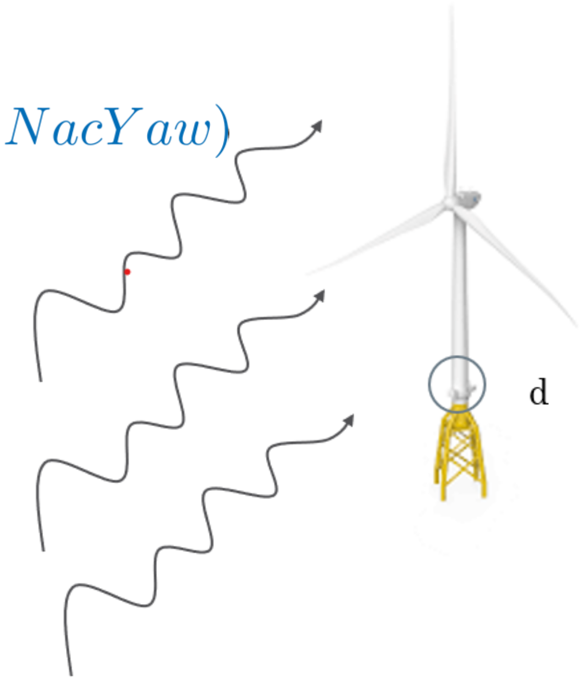
Comparison criterion

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



RESULTS FOR DAMAGE PREDICTION OF A WIND TURBINE

$$x = (\bar{U}, TI, NacYaw)$$



Wind turbine subject to wind loads described by 3 parameters:
$\bar{U}$ mean of wind speed (10mn), TI turbulence intensity,
$NacYaw$ misalignment angle

TurbSim to simulate multiple realizations $(\bar{U}, TI, NacYaw)$

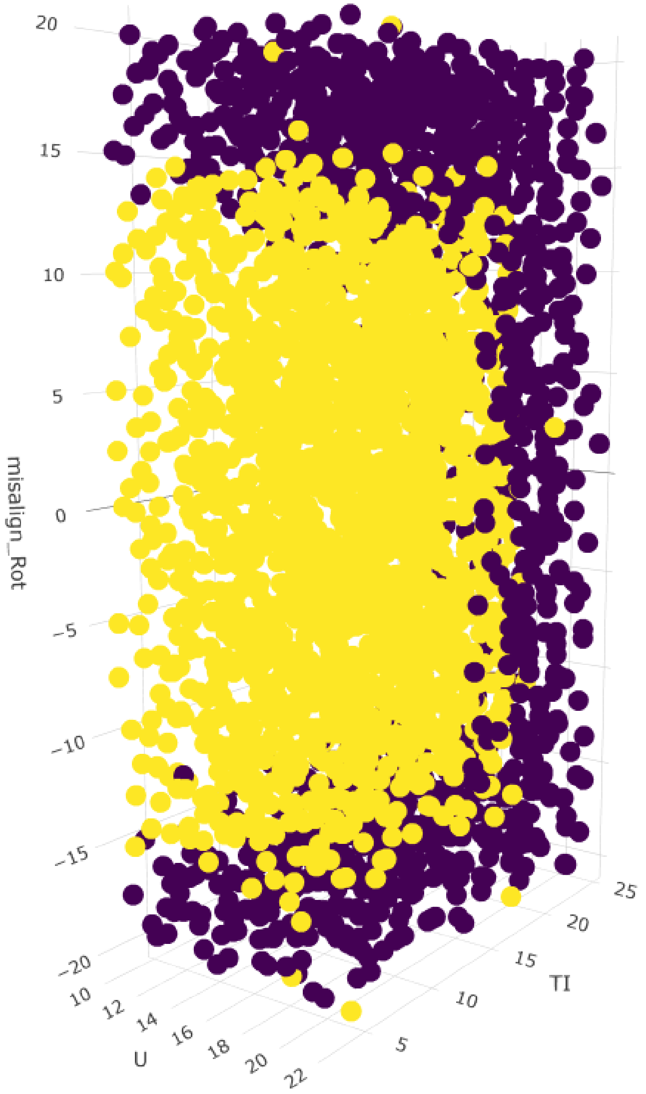
+

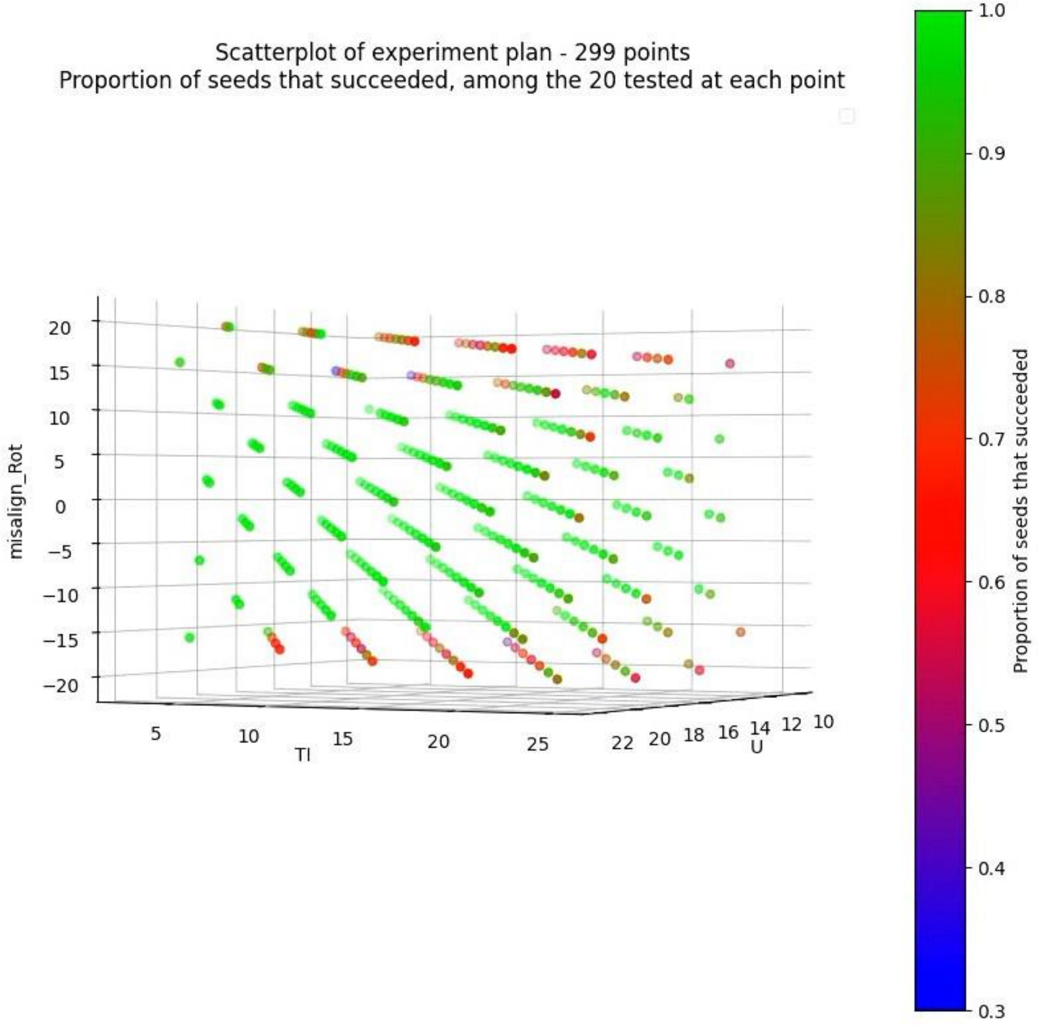
FAST simulator
+ Python scripts

↓

Predictions of damage at the bottom of the tower

RESULTS FOR DAMAGE PREDICTION OF A WIND TURBINE





TurbSim to simulate multiple realizations $(\bar{U}, TI, NacYaw)$

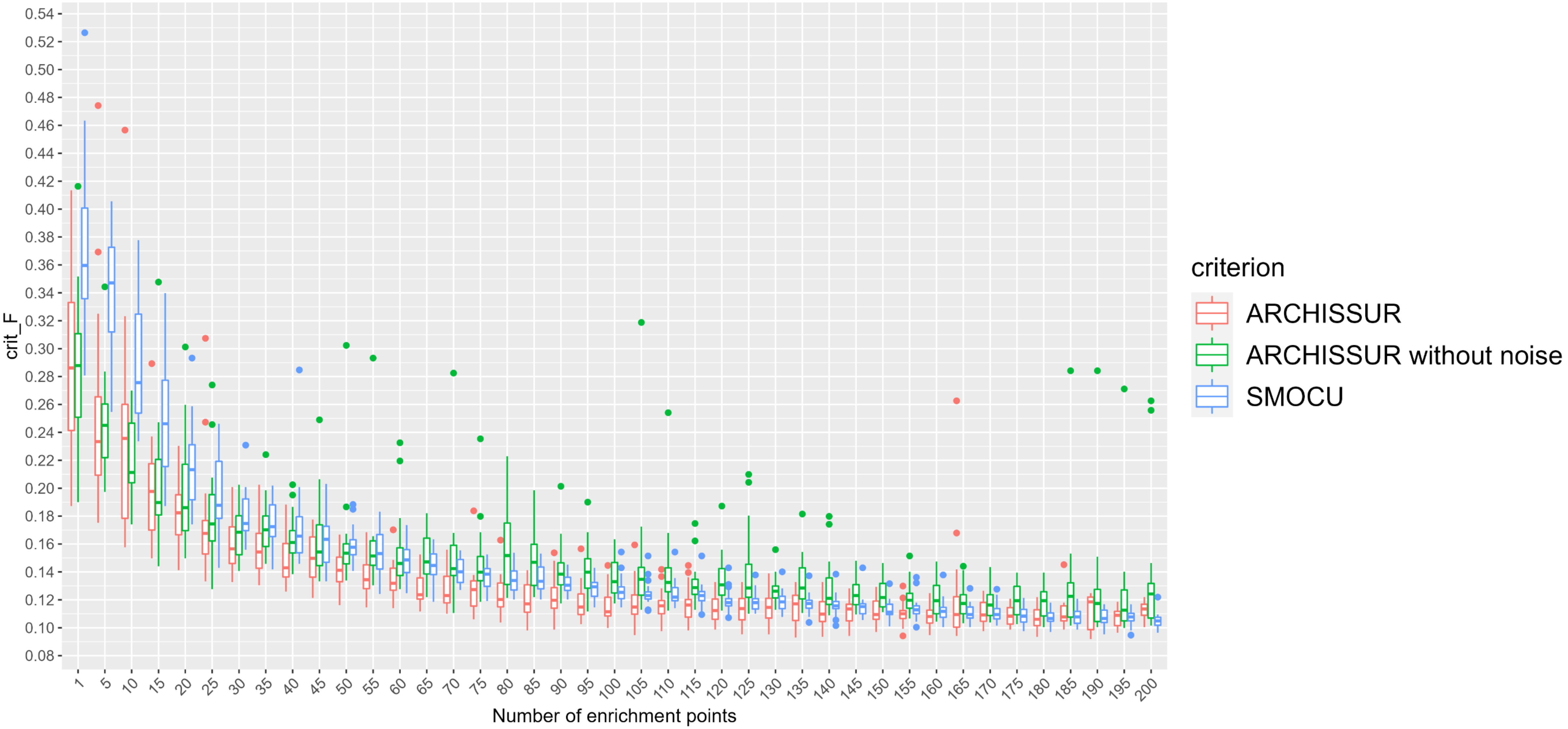
+

FAST simulator + Python scripts

↓

Predictions of damage at the bottom of the tower

RESULTS FOR DAMAGE PREDICTION OF A WIND TURBINE



RESULTS ON A 10D FUNCTION