Black-box optimization with hidden constraints

Delphine Sinoquet* and Sébastien Le Digabel**
joint work with Morgane Menz* and Christophe Tribes**

* IFP Énergies Nouvelles
** GERAD, Polytechnique Montréal

CONTEXT: ROBUST / RELIABLE CONCEPTION OF COMPLEX SYSTEMS

Uncertain variables
u

« Controllable » variables
= design variables x → **Complex and time consuming black-box simulator** → $f(x,u)$, $g(x,u)$ Quantities of interest

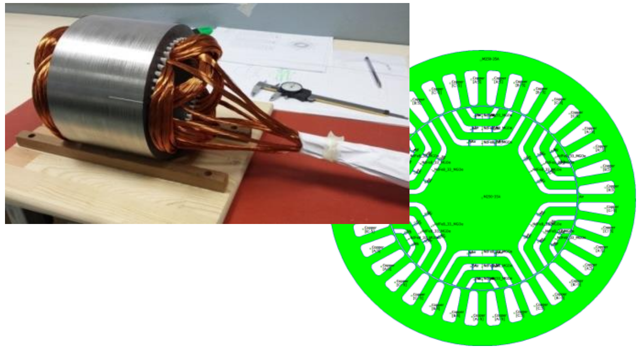
minimize/maximize
constrain

Applications in conception

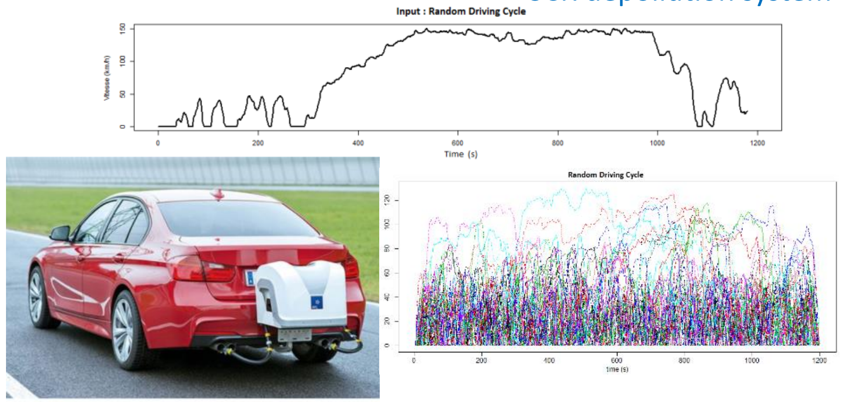
- **Offshore wind turbines:** reliability regarding environmental conditions (*e.g.* wind, wave)
- **Electrical machines:** robustness w.r.t. design parameter dispersions (manufactoring), variability of component characteristics (*e.g.* electromagnetic properties of magnets), …

phD thesis of A. Cousin, A. Hirvoas, C. Duhamel

phD thesis of A. Reyes Reyes
robust design of electrical engines



phD thesis of R El Amri
SCR depollution system



HIDDEN CONSTRAINTS IN OPTIMIZATION

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

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

➔ **Avoid non feasible areas**

HIDDEN CONSTRAINTS IN OPTIMIZATION

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

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

➔ Use Gaussian Process Classifier and Archissur active learning procedure (previous talk)

➔ **Avoid non feasible areas**

➔ Coupling GPC learning and optimization procedure

BIBLIOGRAPHY OVERVIEW

A few studies on **surrogate-based optimization** coupled with a classifier to learn hidden constraints

- Sacher et al, 2018, A classification approach to efficient global optimization in presence of non-computable domains
- Müller and Day, 2019, Surrogate optimization of computationally expensive black-box problems with hidden constraints
- Bussemaker et al., 2024, Surrogate-Based Optimization of System Architectures Subject to Hidden Constraints
  - ➢ Next talk by Nathalie Bartoli

A first study on **direct search method MADS**

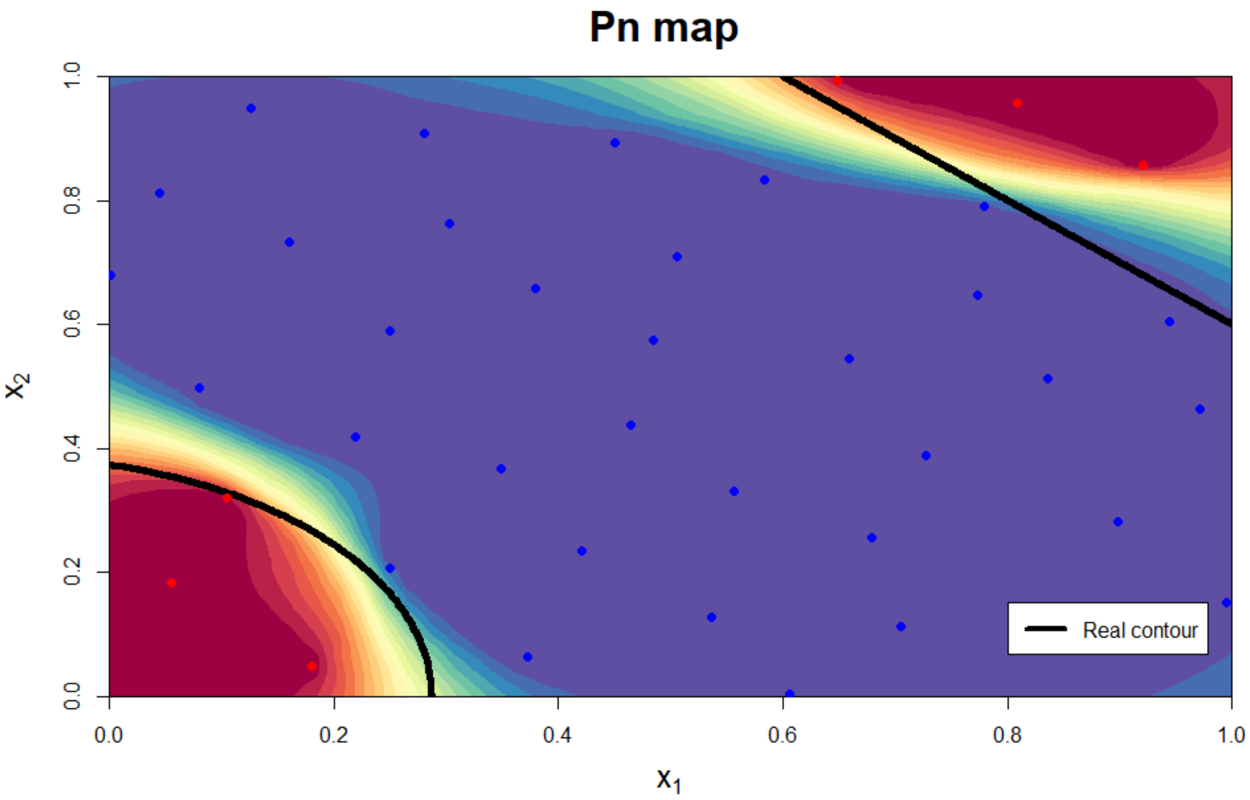
- C. Audet et al., 2020, Binary unrelaxable and hidden constraints in blackbox optimization

➔ **Design strategies based on GPC/Archissur for various optimizers**

REMINDER: GAUSSIAN PROCESS CLASSIFIER

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

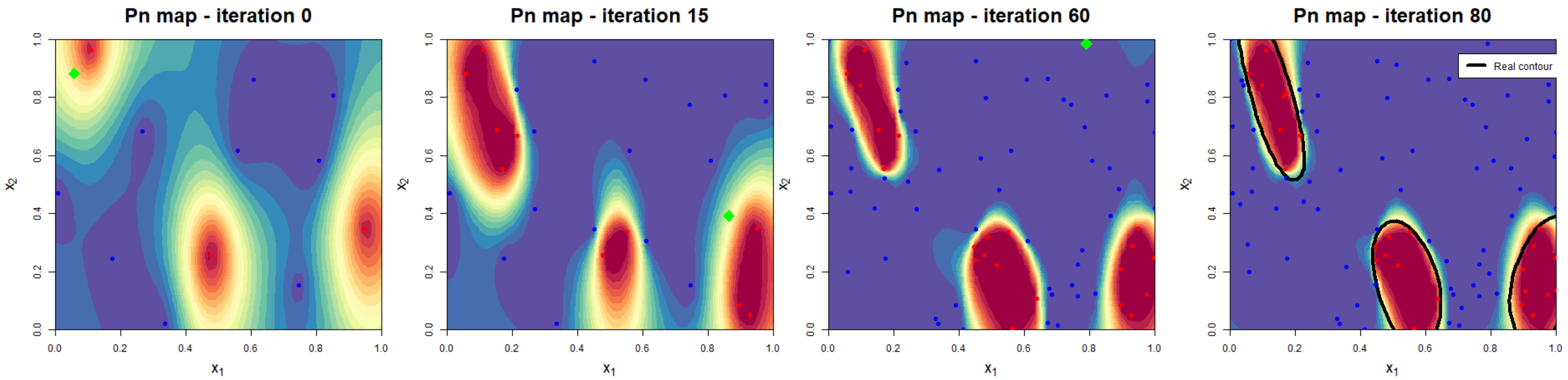
$$p_n(x) = \mathbb{P}[Simulation(x) \neq NaN \,|\mathcal{X}_n, \mathcal{Y}_n]$$



Blue : feasible simulated points

Red : non-feasible simulated points

REMINDER: ACTIVE LEARNING BY ARCHISSUR



Blue : current feasible simulated points
Red : current non-feasible simulated points
Green : new point to be simulated (Archissur)

OUTLINE

- Strategies to handle hidden constraints in optimization

- Coupling with various optimization methods
  - Mesh Adaptive Direct Search (NOMAD)
  - Trust Region Derivative Free optimization method (SQA)
  - Bayesian Optimization

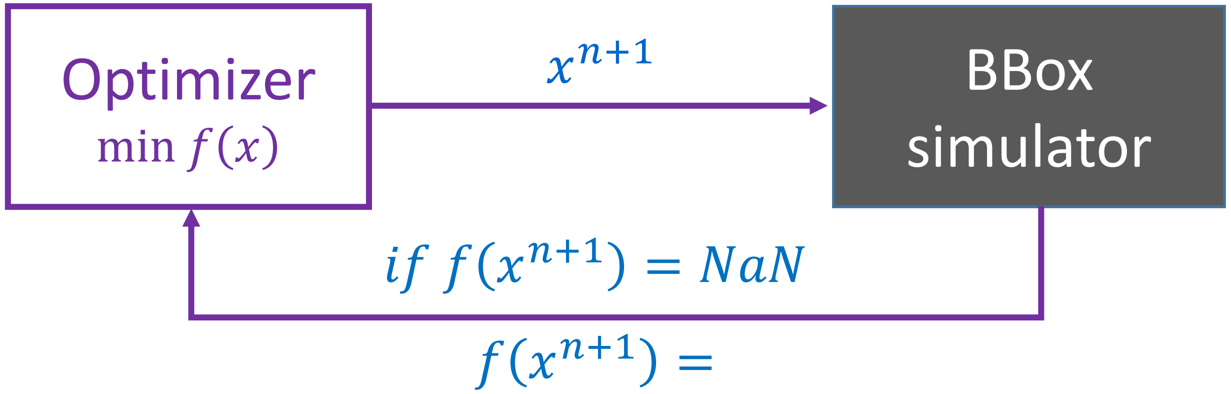
- Application to a calibration problem

OPTIMIZING WITH HIDDEN CONSTRAINTS

- Naïve approach

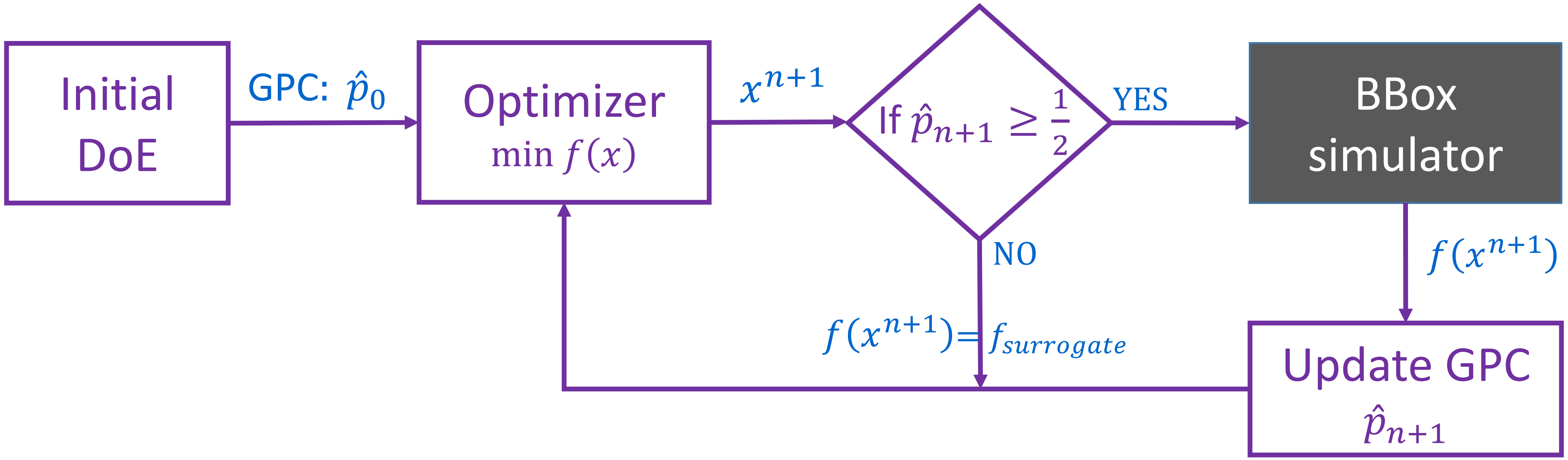
  In case of a simulator crash: replace the NaN outputs by large « surrogate » values

  - *i.e.* maximal value of the objective functions associated with closest points
    in order to avoid a further exploration of this “risky” area



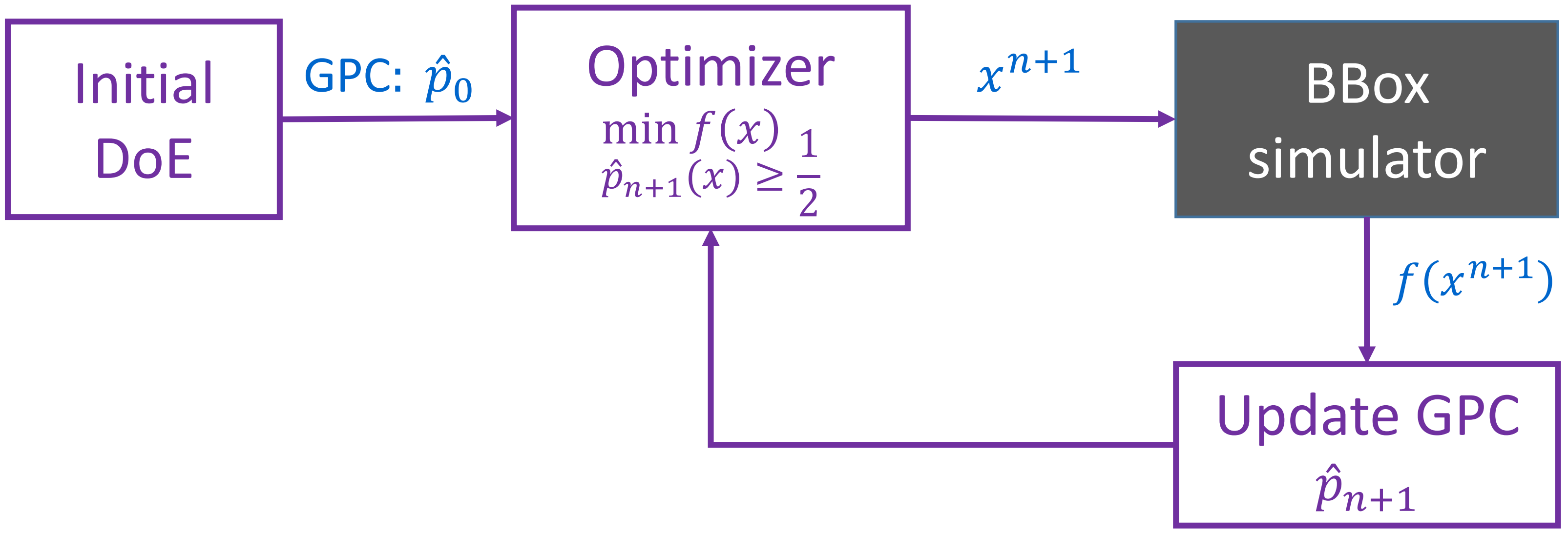
OPTIMIZING WITH HIDDEN CONSTRAINTS

- Our first 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
  - **Prior constraint :** do not simulate the point in case of a high probability of crash $\hat{p}_n(x) < \frac{1}{2}$
    ➔ save some simulations in risky regions predicted by the GPC classifier



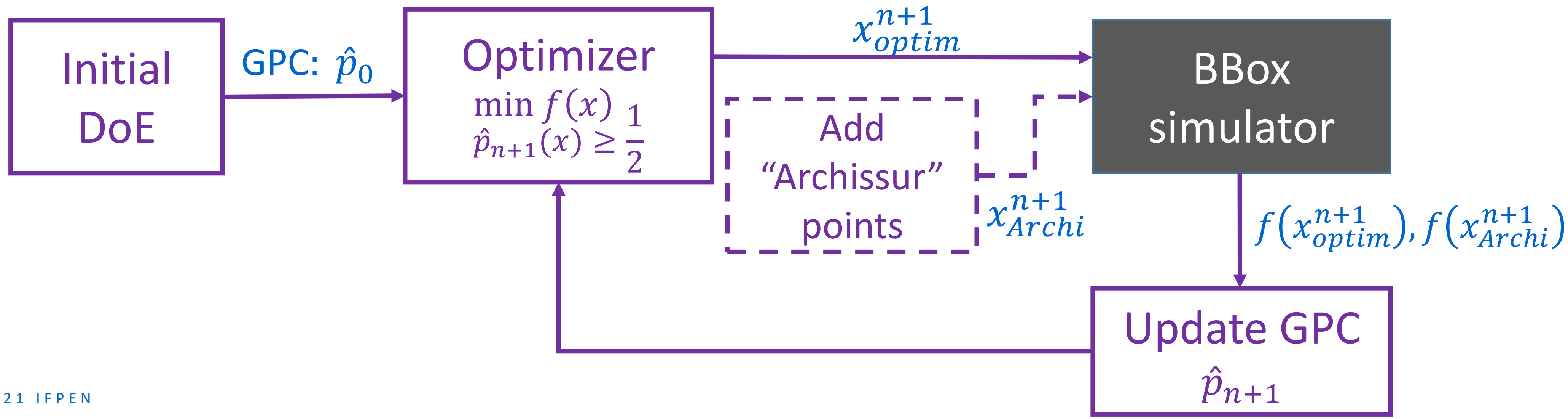
OPTIMIZING WITH HIDDEN CONSTRAINTS

- Our second 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
  - **Additional constraint on $\hat{\boldsymbol{p}}_{\boldsymbol{n}}(\boldsymbol{x})$** (cheap constraint to evaluate) $\hat{p}_n(x) < \frac{1}{2}$
    ➔ additional constraint to avoid the *risky* regions



OPTIMIZING WITH HIDDEN CONSTRAINTS

- Our third 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
  - **Additional constraint on $\hat{p}_n(x)$ updated with GPC model improvement steps (Archissur points)** when close to convergence and the current iterate is close to infeasible set (points of 2 classes around current iterate)
    ➔ additional constraint to avoid the *risky* regions
    ➔ additional simulations to improve the GPC classifier (model improvement steps)



OUTLINE

- Strategies to handle hidden constraints in optimization

- Coupling with various optimization methods
  - Mesh Adaptive Direct Search (NOMAD)
  - Trust Region Derivative Free optimization method (SQA)
  - Bayesian Optimization

- Application to a calibration problem

[0] Initializations $(\mathbf{x}_0, \delta^0)$

[1] Iteration k

- **[1.1] Search** (flexible part)
  - select a finite number of mesh points
  - evaluate candidates opportunistically
- **[1.2] Poll** (if **Search** failed) ("rigid" part)
  - construct poll set $P_k = \{\mathbf{x}_k + \delta^k \mathbf{d} : \mathbf{d} \in D_k\}$
  - sort(P_k)
  - evaluate candidates opportunistically

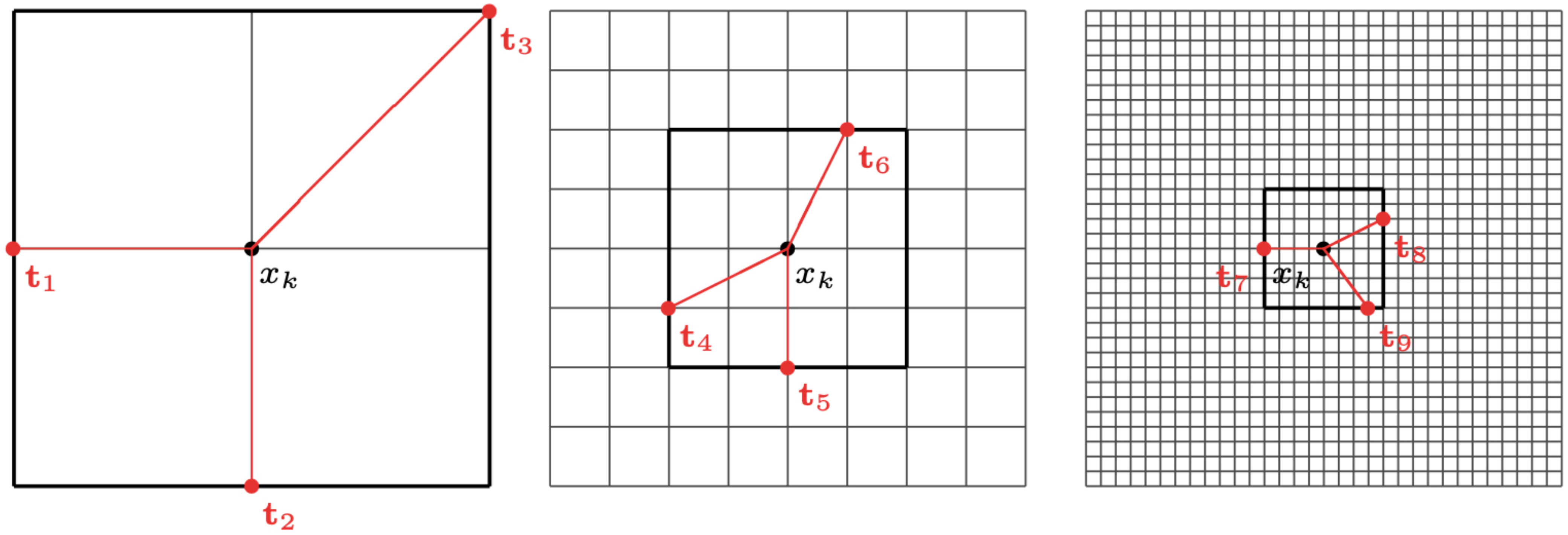
[2] Updates

- if success
  - $\mathbf{x}_{k+1} \leftarrow$ success point
  - increase δ^k
- else
  - $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k$
  - decrease δ^k
- $k \leftarrow k + 1$, stop or go to **[1]**

The MADS algorithm [Audet and Dennis, Jr., 2006] (unconstrained version)

MADS: ILLUSTRATION IN 2D

$\delta^k = \Delta^k = 1$ $\delta^{k+1} = 1/4$ $\Delta^{k+1} = 1/2$ $\delta^{k+2} = 1/16$ $\Delta^{k+2} = 1/4$



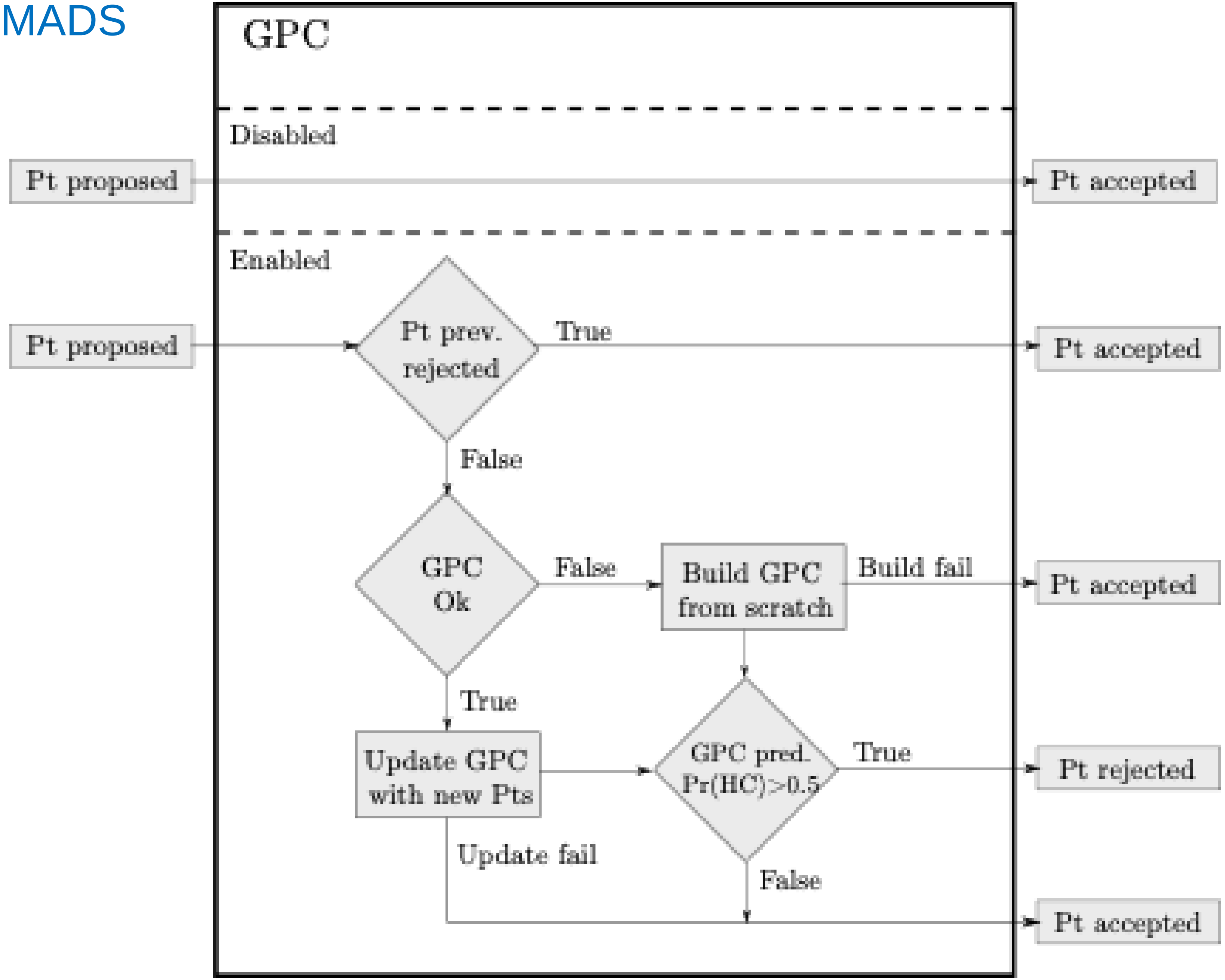
poll trial points $=\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3\}$ $= \{\mathbf{t}_4, \mathbf{t}_5, \mathbf{t}_6\}$ $= \{\mathbf{t}_7, \mathbf{t}_8, \mathbf{t}_9\}$

NOMAD: GITHUB.COM/BBOPT/NOMAD

- C++ implementation of the MADS algorithm [Audet and Dennis, Jr., 2006]
- Standard C++. Runs on Linux, Mac OS X and Windows
- Parallel versions
- MATLAB versions; Multiple interfaces (Python, Julia, etc.)
- Open and free – LGPL license
- Download at `https://www.gerad.ca/nomad`
- Support at `nomad@gerad.ca`

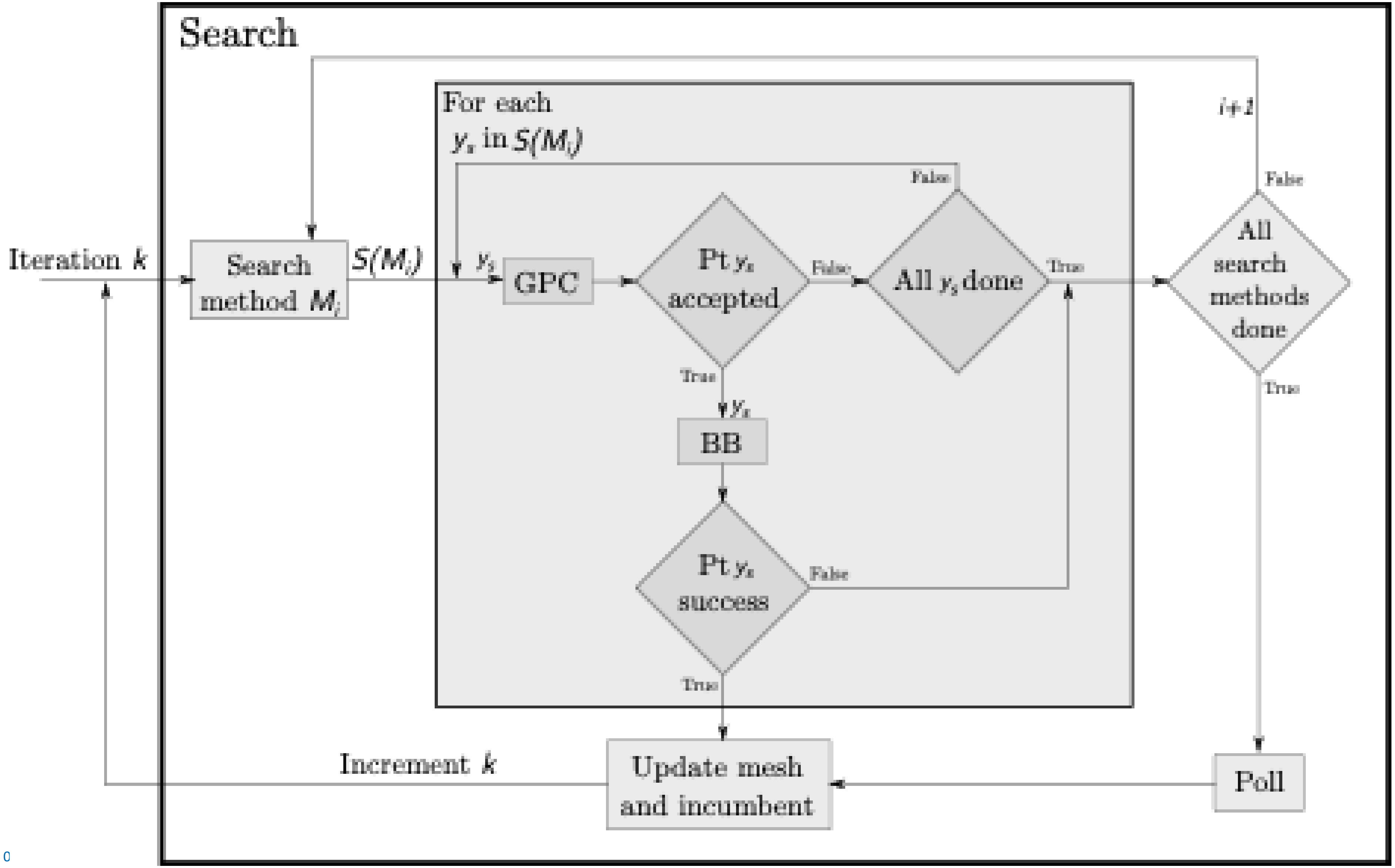
- Related articles in TOMS [Le Digabel, 2011] and [Audet et al., 2022]

GPC IN MADS

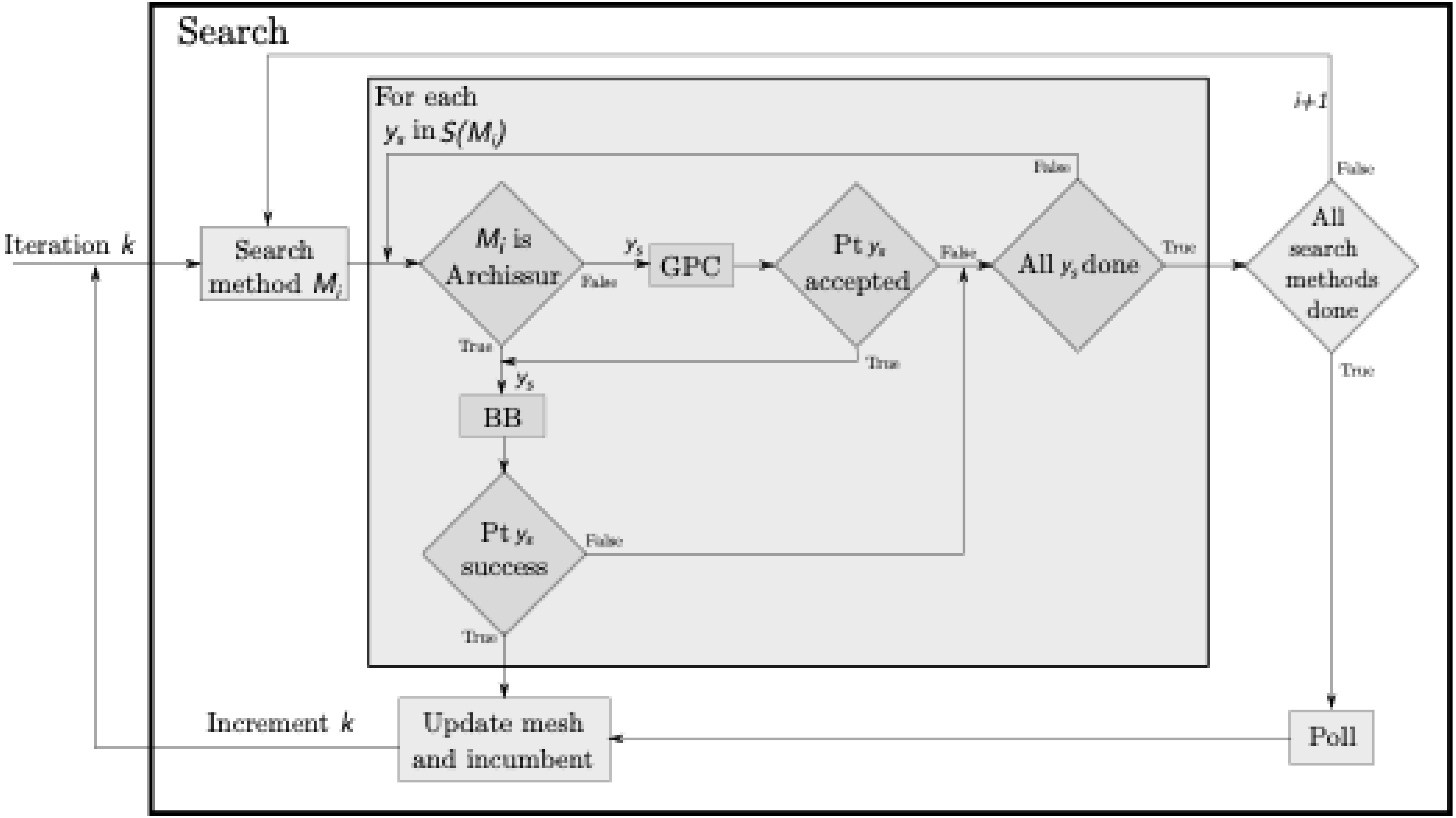


GPC
Disabled
Pt proposed
Pt accepted
Enabled
Pt proposed
Pt prev. rejected
True
Pt accepted
False
GPC Ok
False
Build GPC from scratch
Build fail
Pt accepted
True
Update GPC with new Pts
GPC pred. Pr(HC)>0.5
True
Pt rejected
Update fail
False
Pt accepted

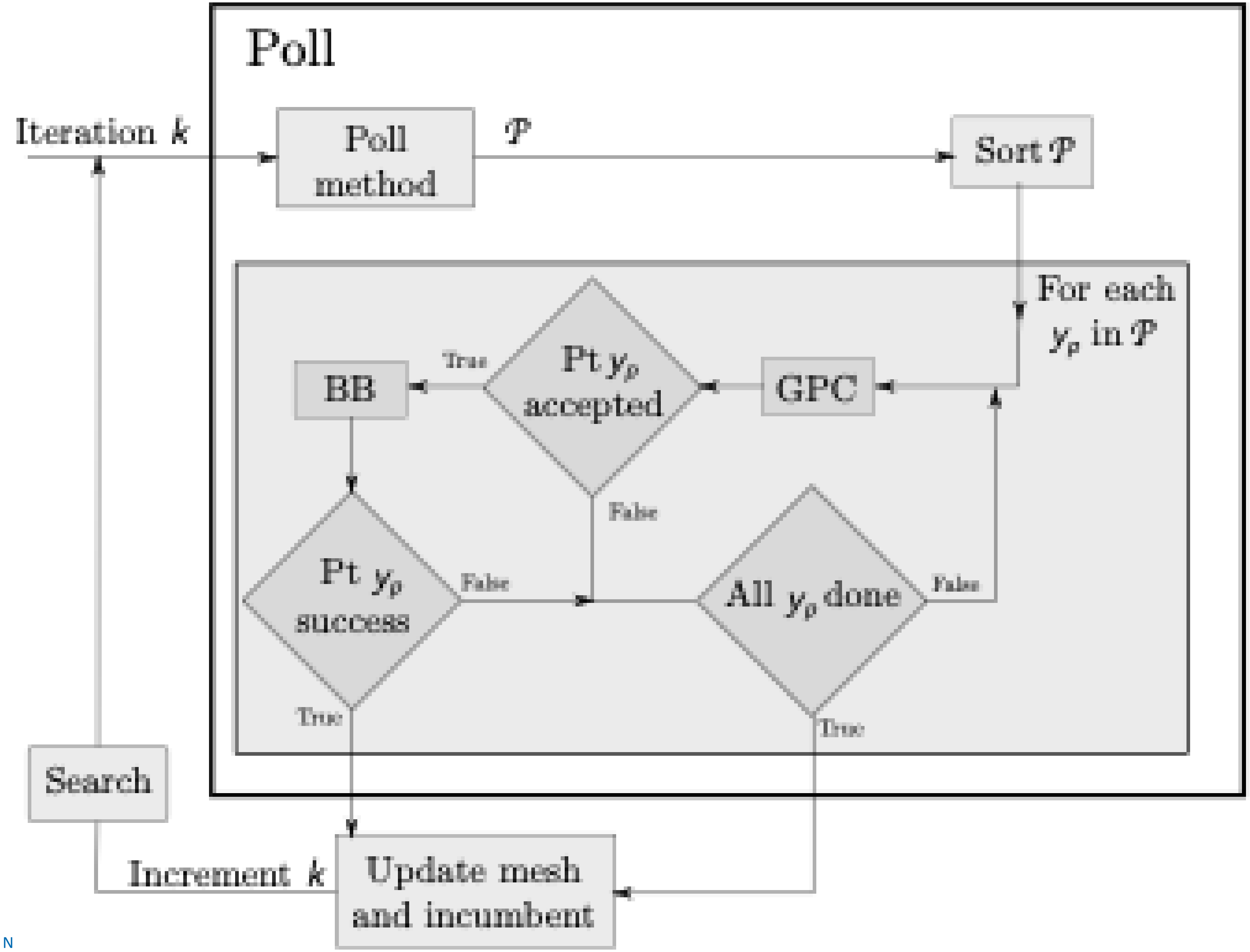
GPC IN MADS SEARCH



GPC IN MADS SEARCH (WITH ARCHISSUR)

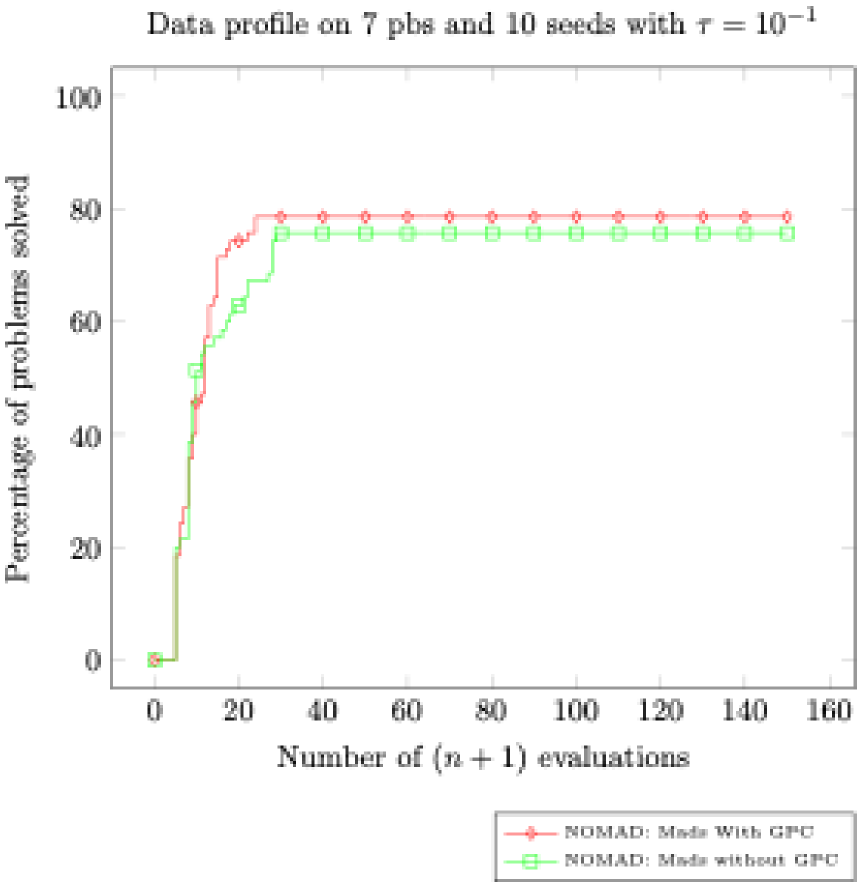


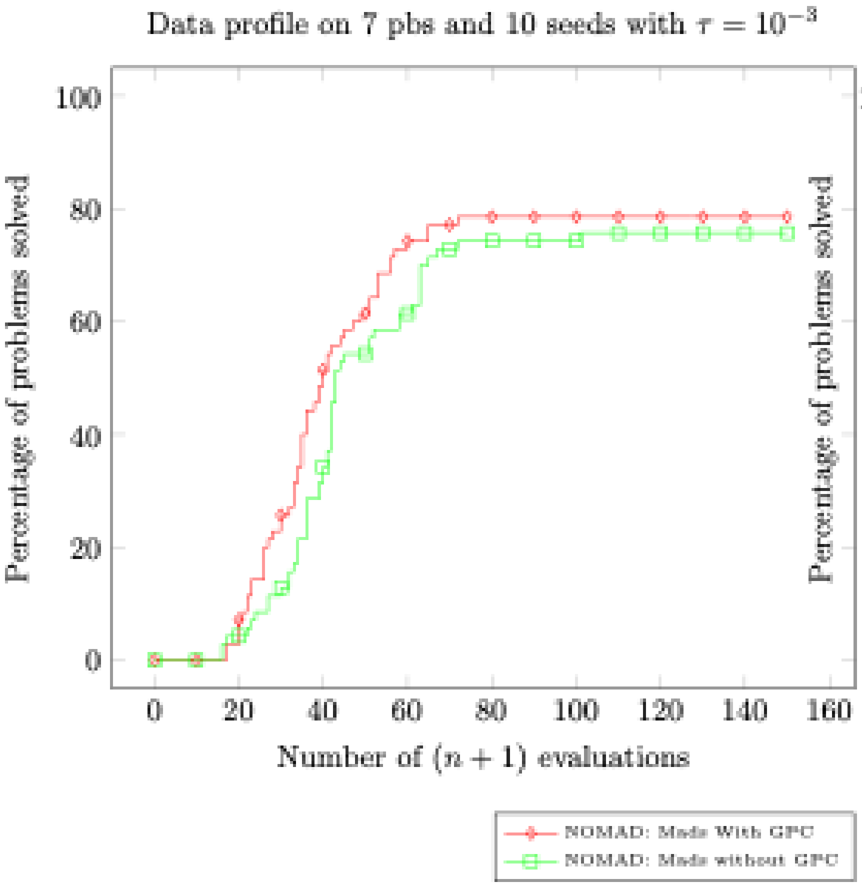
GPC IN MADS POLL

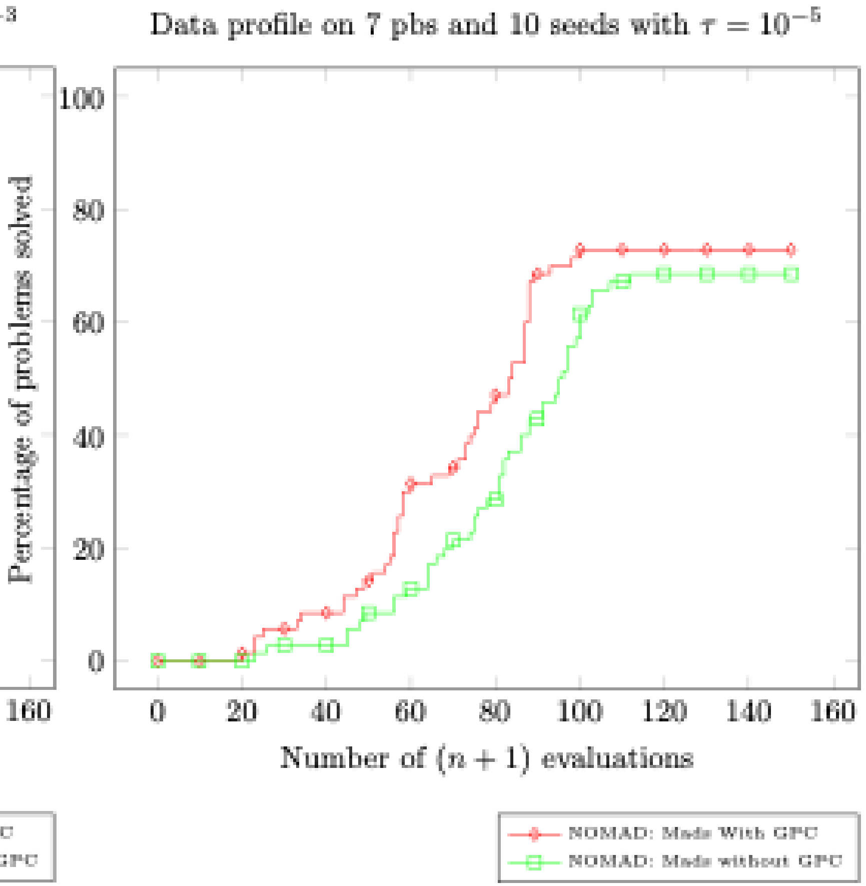


RESULTS: MADS WITH OR WITHOUT GPC (NO ARCHISSUR)

- Data profiles:
  - 7 analytical problems
  - 10 runs for each problem (different seeds)







OUTLINE

- Strategies to handle hidden constraints in optimization

- Coupling with various optimization methods
  - Mesh Adaptive Direct Search (NOMAD)
  - Trust Region Derivative Free optimization method (SQA)
  - Bayesian Optimization

- Application to a calibration problem

A DERIVATIVE FREE TRUST REGION OPTIMIZATION METHOD

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

= extension of NEWUOA [Powell, 2007] to constrained optimization

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

- **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)
- Subproblems solved by a SQP method

SQA WITH HIDDEN CONSTRAINTS

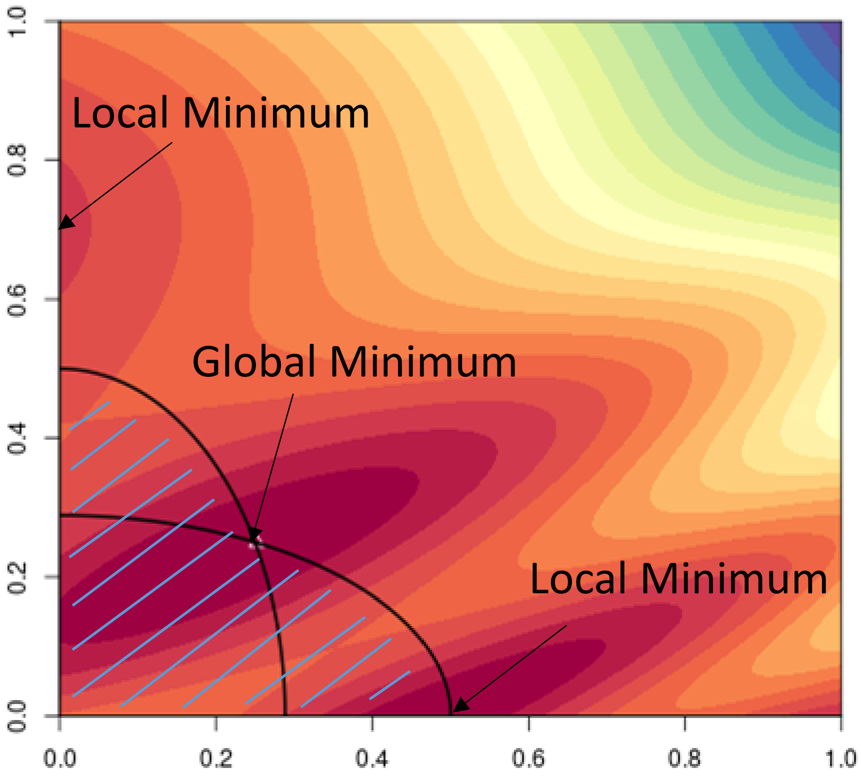
1) **Prior constraint** and NA replaced by maximal objective function value among simulations inside the trust region

2) **Constraint** $\boldsymbol{\hat{p}_n(x) \geq \frac{1}{2}}$ is introduced as an explicit constraint (with derivatives) considered in the subproblems solved by SQP

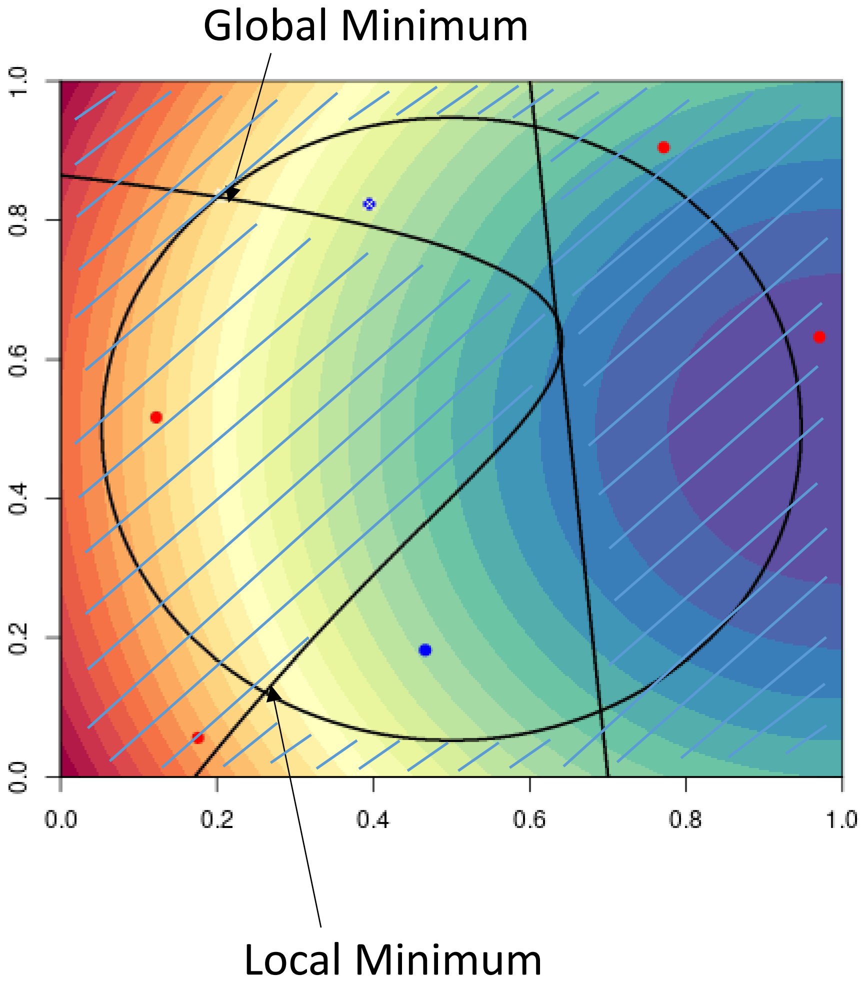
3) **Additional "Archissur" points**
From the current GPC model learnt from the available simulations, add points that minimize the *future* uncertainty on the feasible set [Menz et al, 2023]
   - when the trust region size becomes small (close to convergence)
   - and when the current iterate is close to the infeasible region (points of 2 classes in the current trust region)

NUMERICAL TESTS

1)



2)



Inspired from Sacher et al, 2018

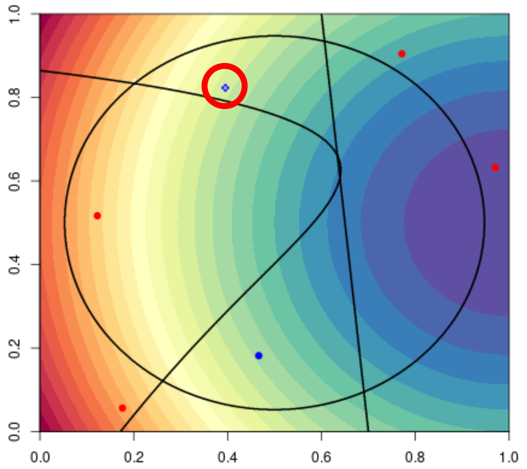
Inspired from Sasena, 2002

TEST METHODOLOGY

Run 10 optimizations for each example

1) 10 initial points for SQA and 10 LHS design of experiments of size 6 for EGO

2) 10 LHS design of experiments of size 6 for both methods but SQA starts with the best point as its initial point

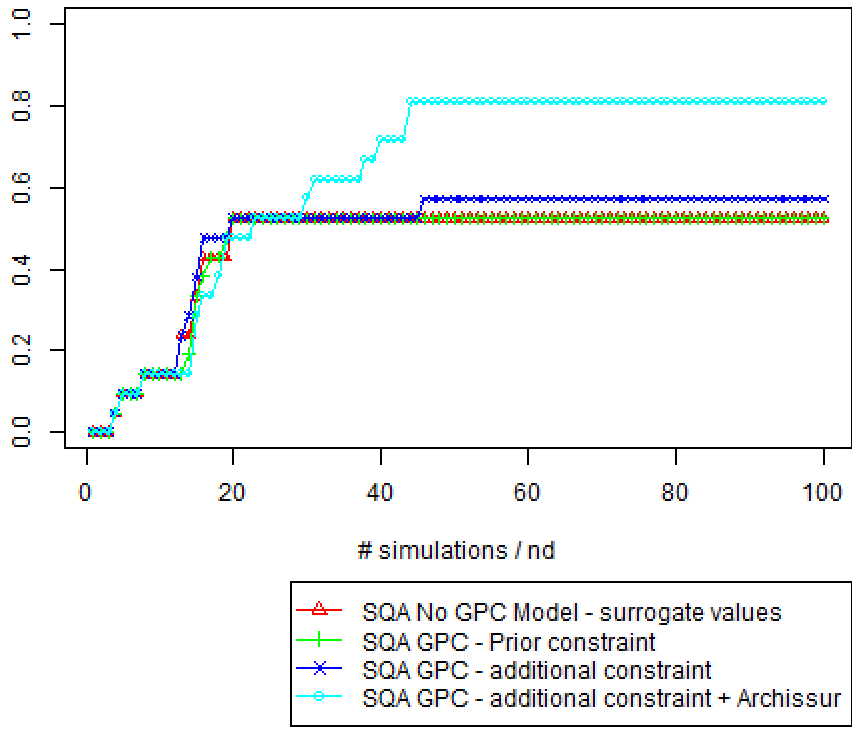


Red circles indicate the best feasible point used as initial point of SQA

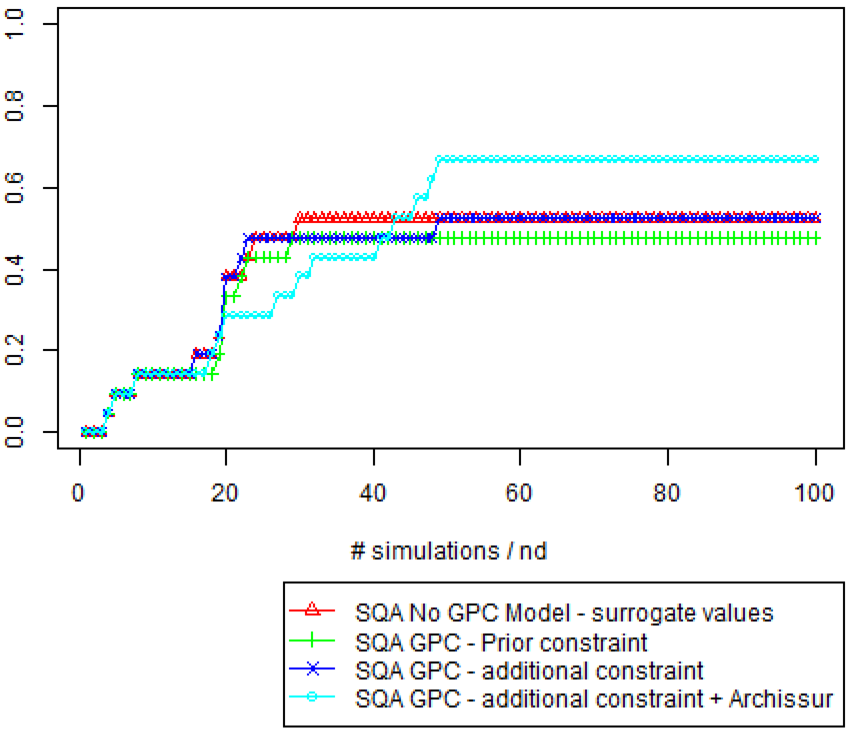
COMPARISON OF THE VARIANTS OF SQA

Data profiles [More and Wild, 2009] for various accuracies on the objective function

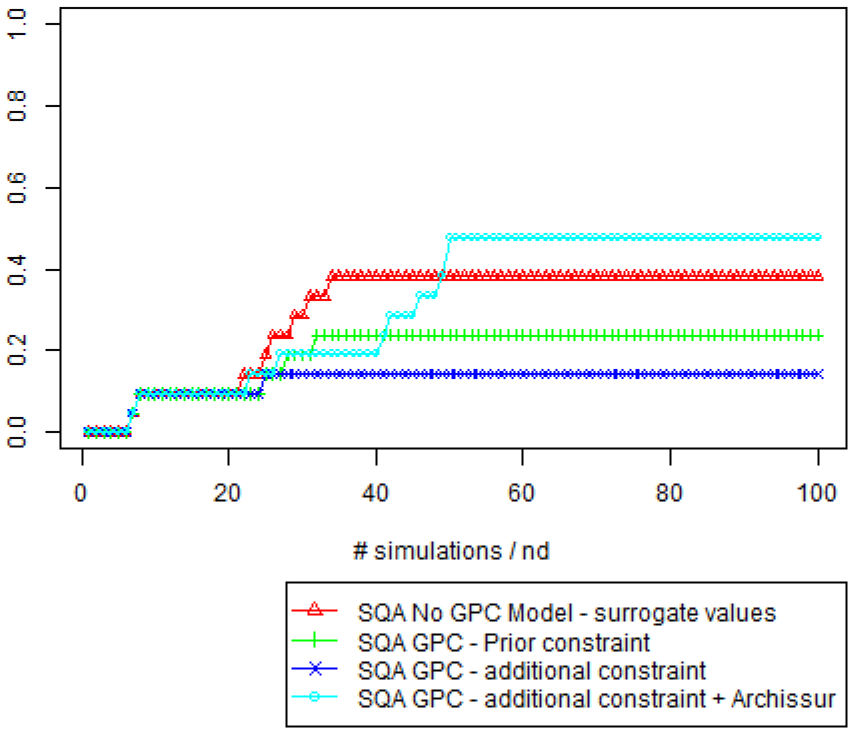
$\tau = 10^{-2}$



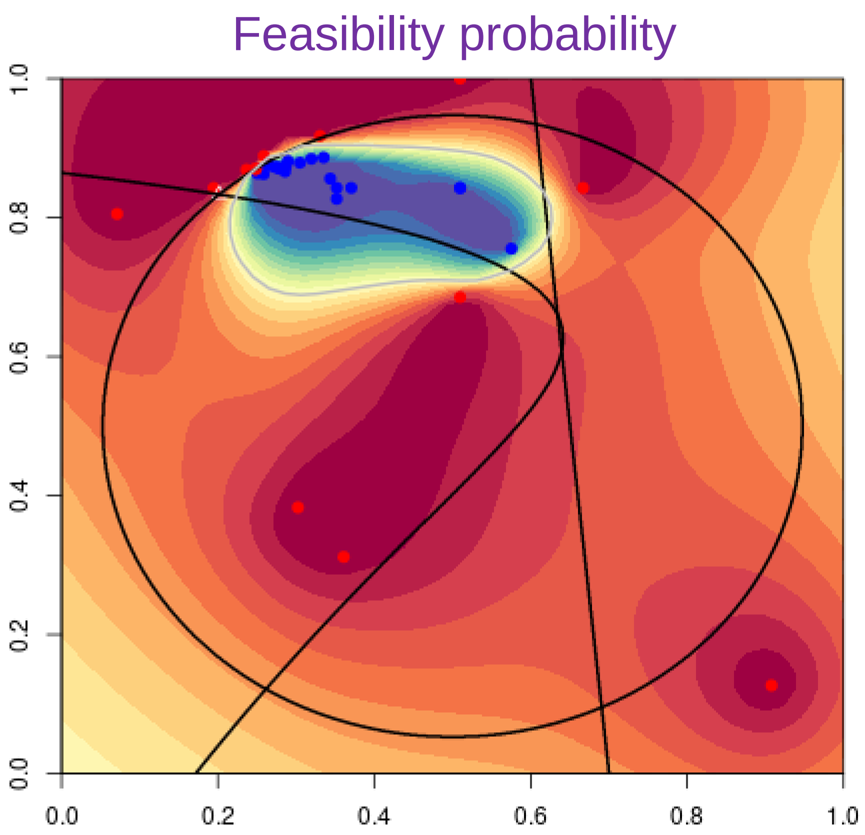
$\tau = 10^{-3}$

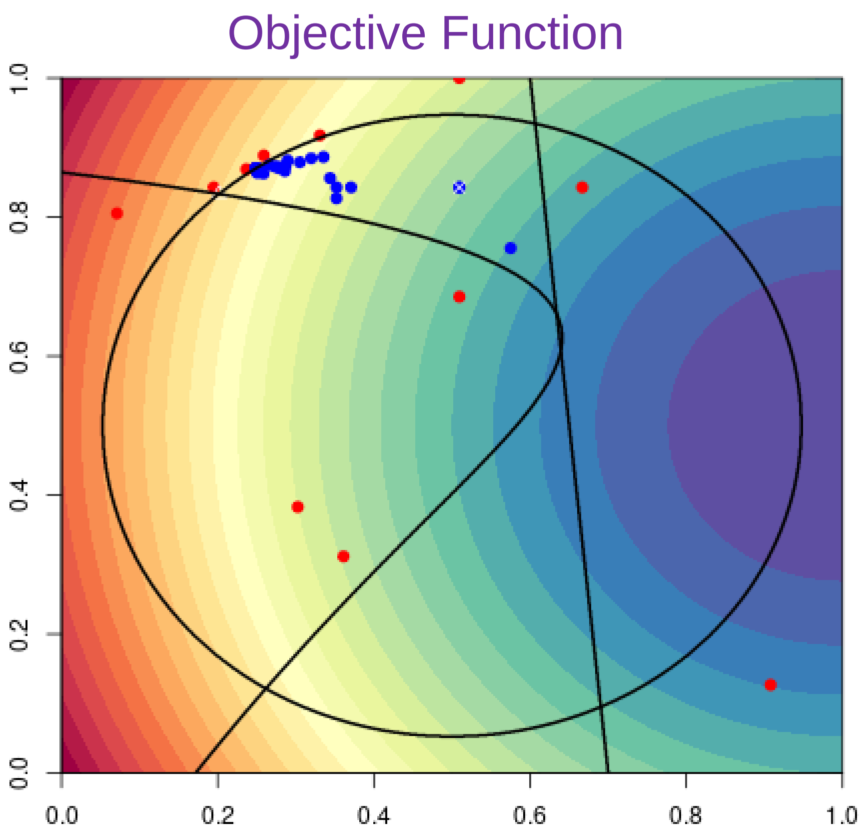


$\tau = 10^{-5}$

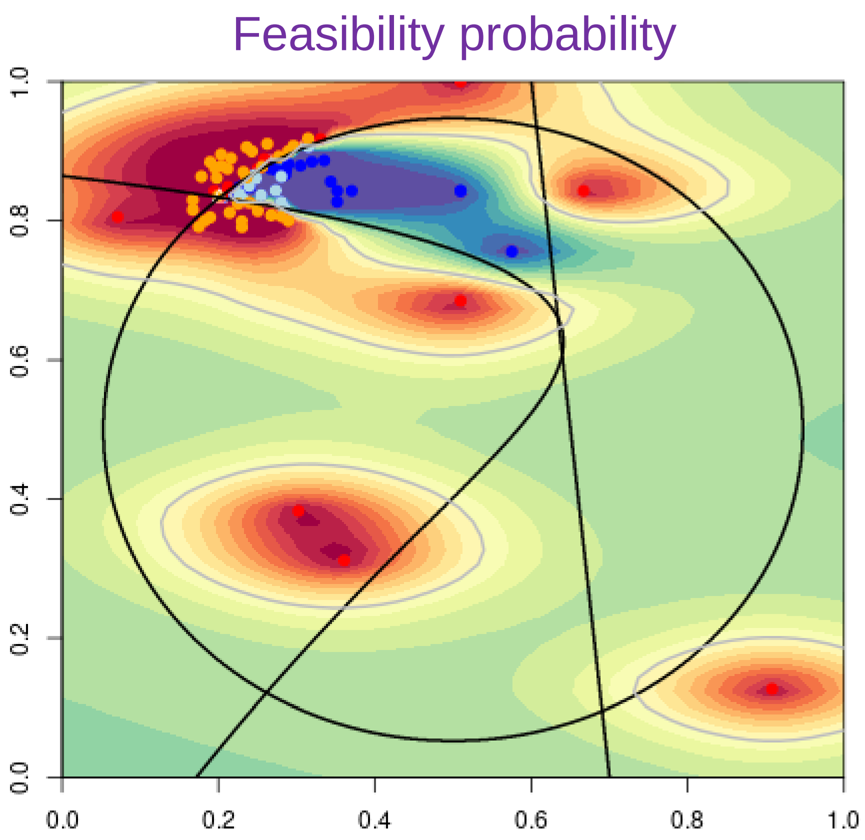


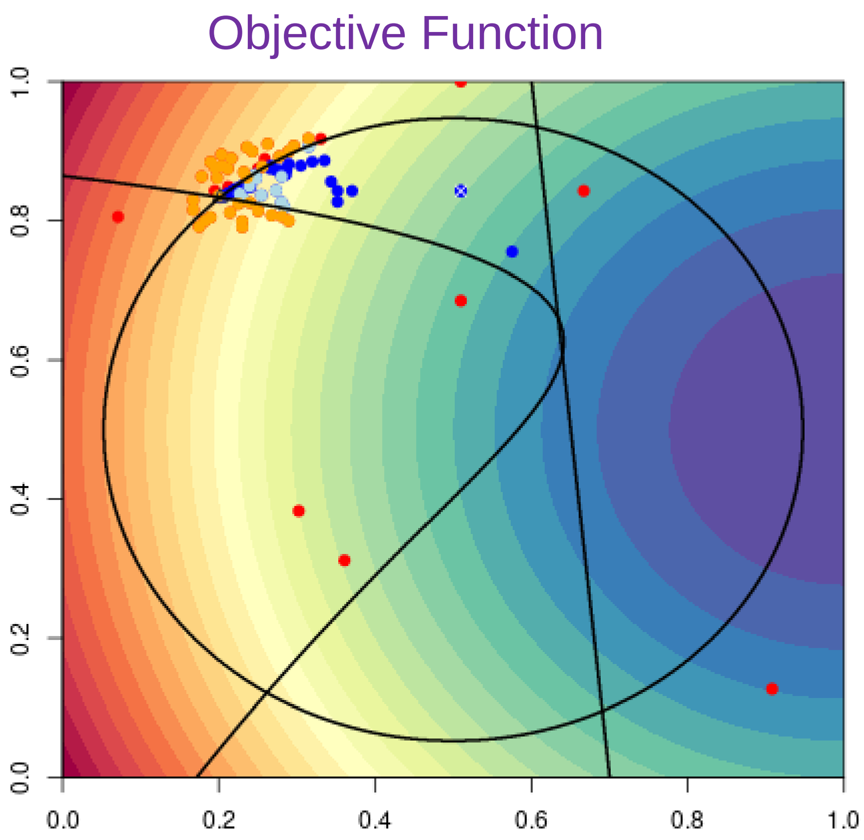
SQA WITH ADDITIONAL CONSTRAINT ON GPC (SECOND EXAMPLE)





SQA WITH ADDITIONAL CONSTRAINT ON GPC + ARCHISSUR POINTS (SECOND EXAMPLE)





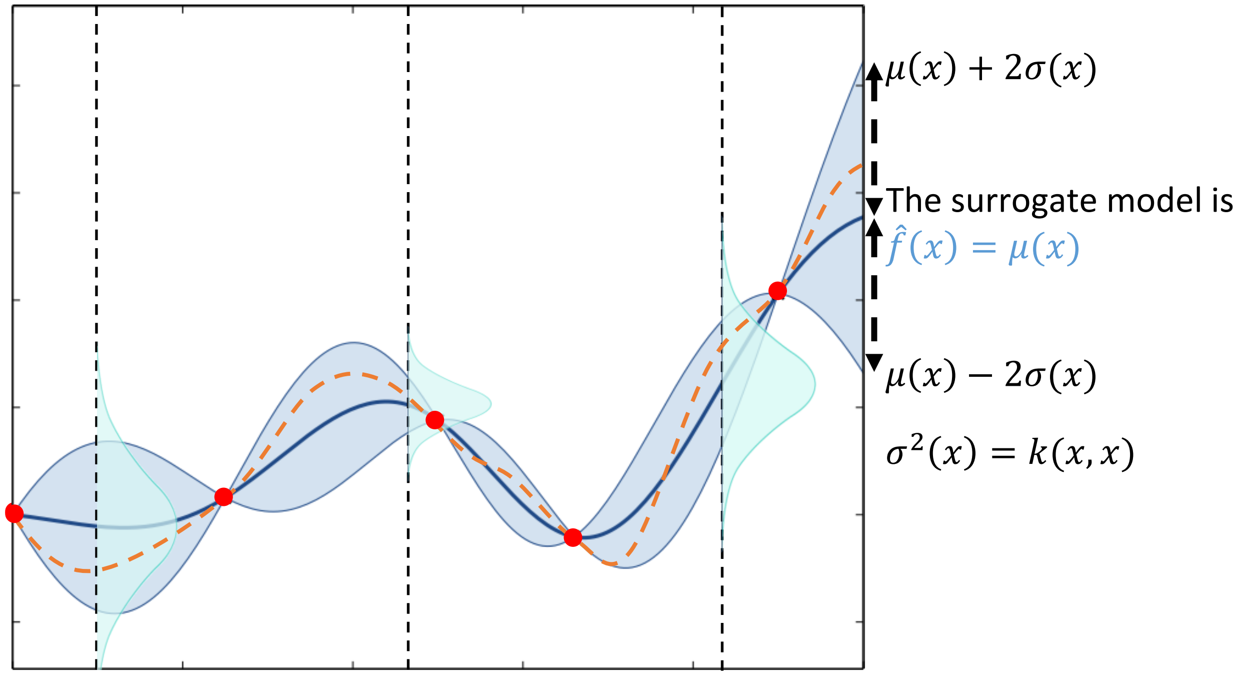
OUTLINE

- Strategies to handle hidden constraints in optimization
- Coupling with various optimization methods
  - Mesh Adaptive Direct Search (NOMAD)
  - Trust Region Derivative Free optimization method (SQA)
  - Bayesian Optimization
- Application to a calibration problem

EGO [Jones et al., 1998]
DiceOptim [Roustant et al., 2012]

Optimization based on Gaussian Process (GP)

- Assumption: the (blackbox) objective function f is a realization of a GP $Z \sim \mathcal{N}(\mu(x), k(x,x))$
- Z_n is the GP conditionned to the available simulations



A BAYESIAN OPTIMIZATION METHOD

EGO [Jones et al., 1998]
DiceOptim [Roustant et al., 2012]

Optimization based on Gaussian Process (GP)

At each iteration, a new simulation point x_{n+1} is chosen as

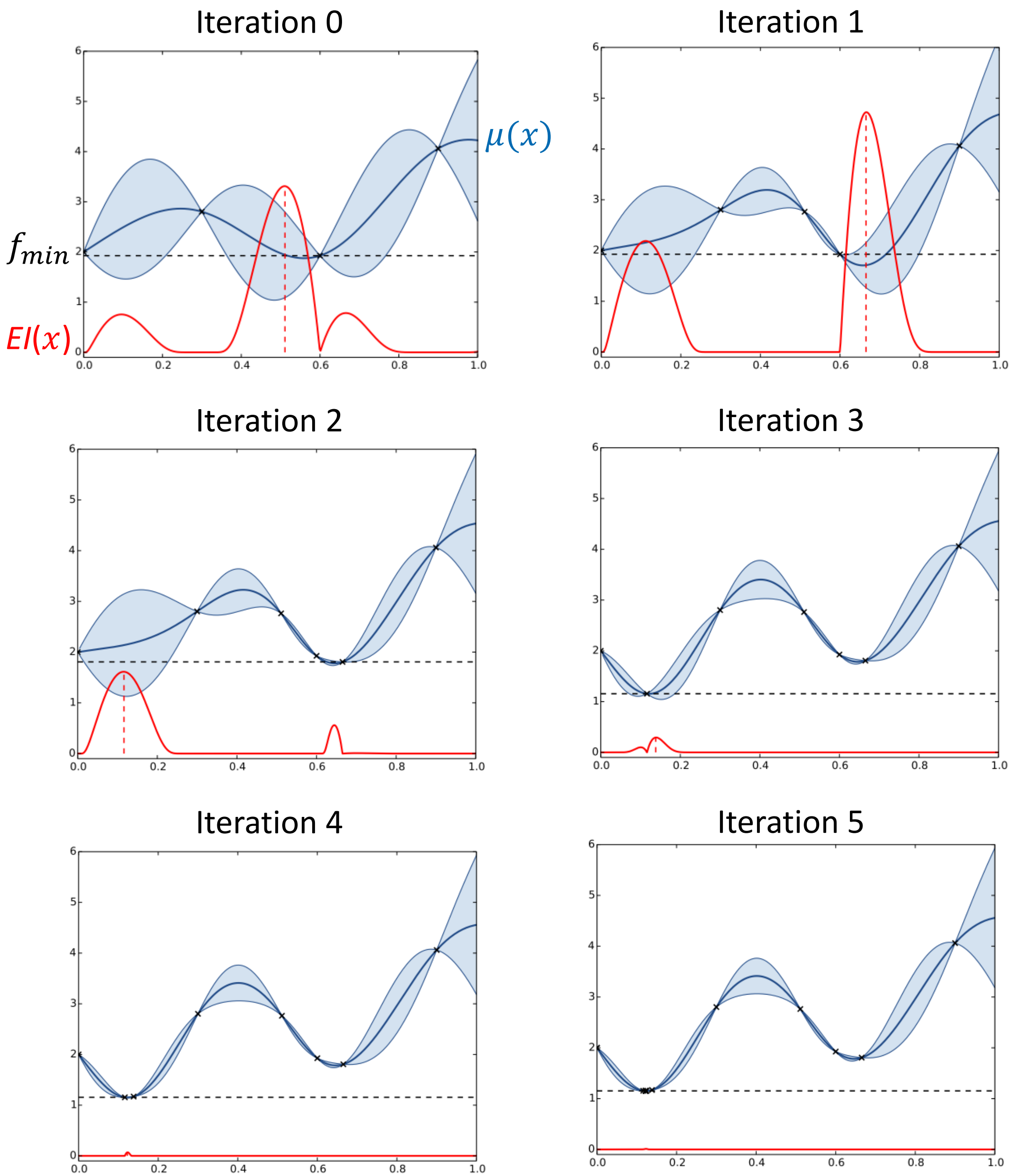
$$x_{n+1} = \underset{x}{argmax}(\mathbb{E}[I(x)])$$

with the infill criterion

$$\mathbb{E}[I(x)] = \mathbb{E}[max(f_{min} - Z_n(x), 0)]$$

f_{min} is the current minimal simulated value of f

EI criterion is a trade-off between exploration/exploitation



Source: R. Le Riche

EGO WITH HIDDEN CONSTRAINTS

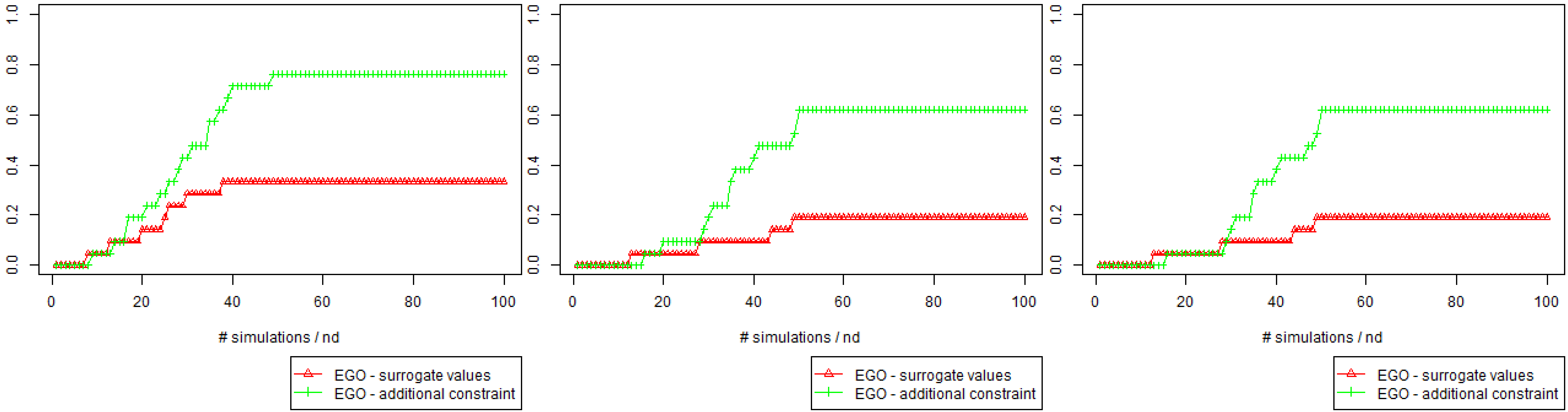
1) **Prior constraint** and NA replaced by GP prediction: mean + 3σ

2) **Constraint** introduced in the infill criterion: $\boldsymbol{\widehat{p}_n(x)EI(x)}$

COMPARISON OF THE VARIANTS OF BAYESIAN OPTIMIZATION (EGO)

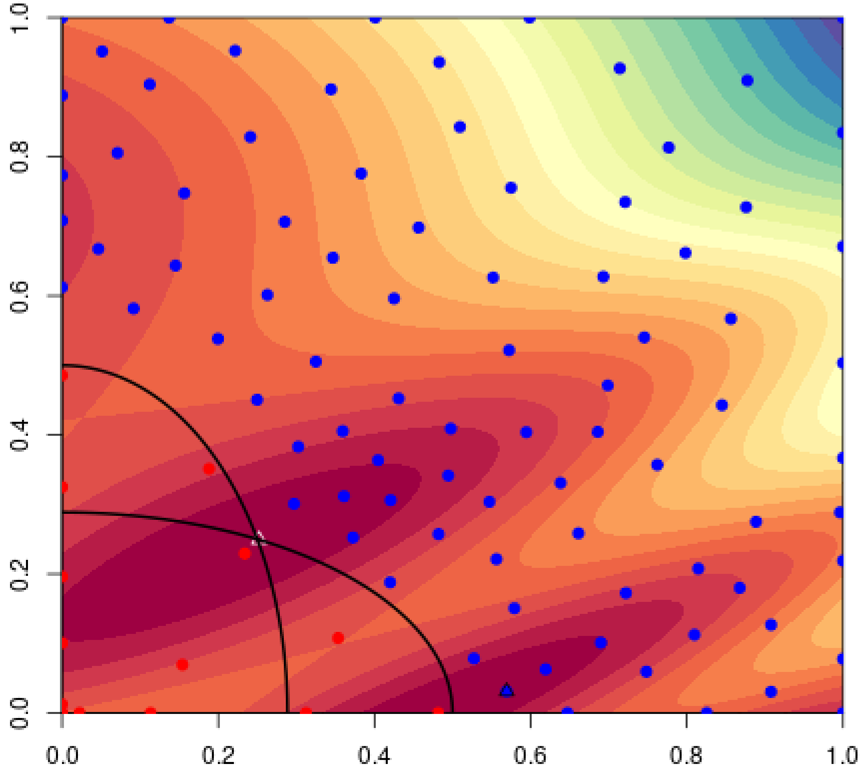
Data profiles [More and Wild, 2009] for various accuracies on the objective function

$\tau = 10^{-2}$ | $\tau = 10^{-3}$ | $\tau = 10^{-5}$



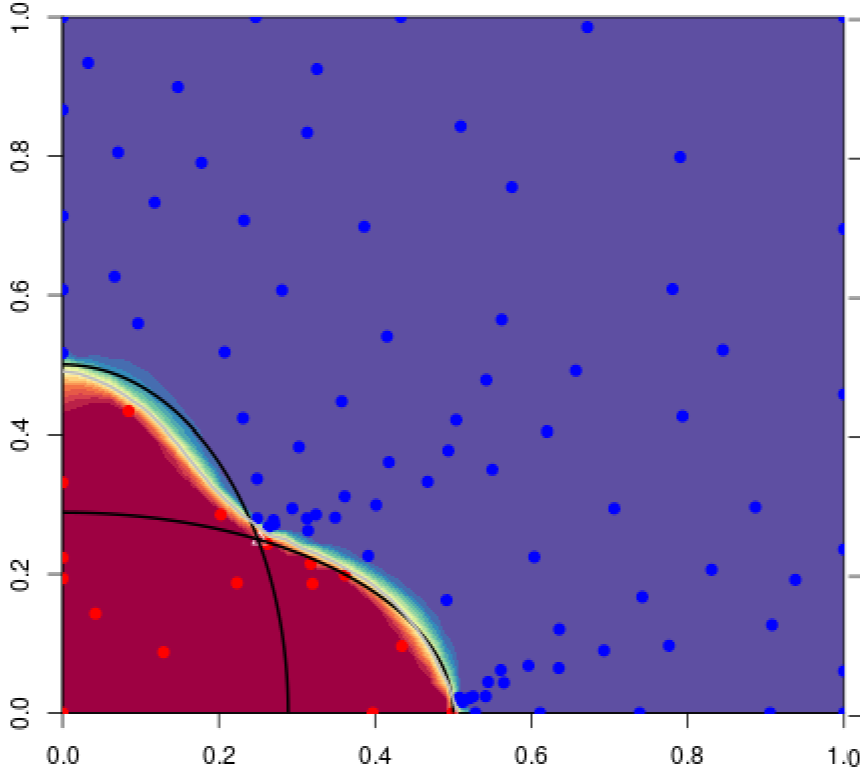
BAYESIAN OPTIMIZATION ON FIRST EXAMPLE

Objective function

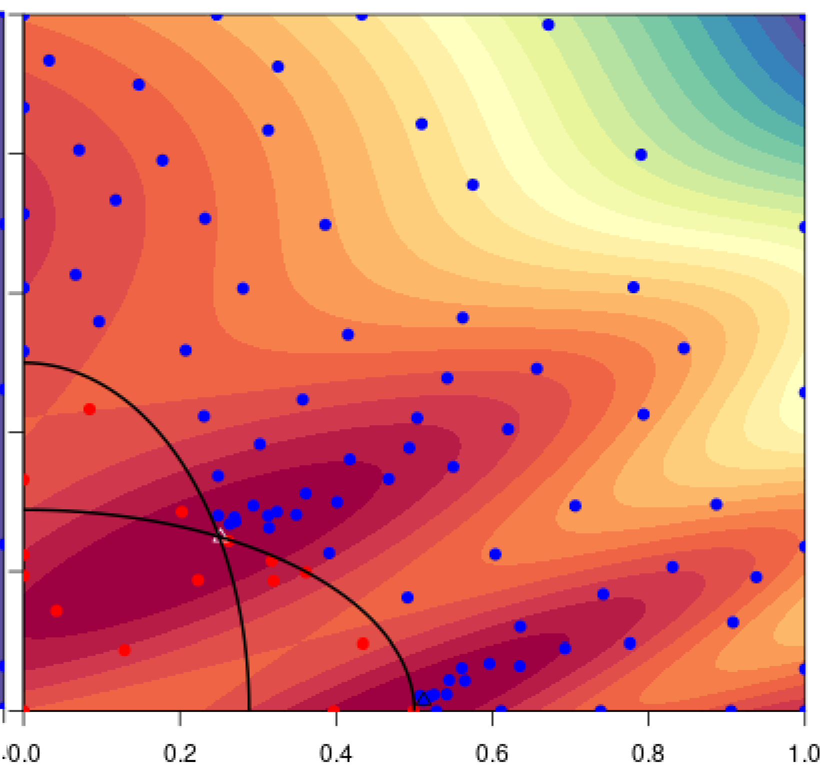


with surrogate values for NaN

Feasibility probablity



Objective function

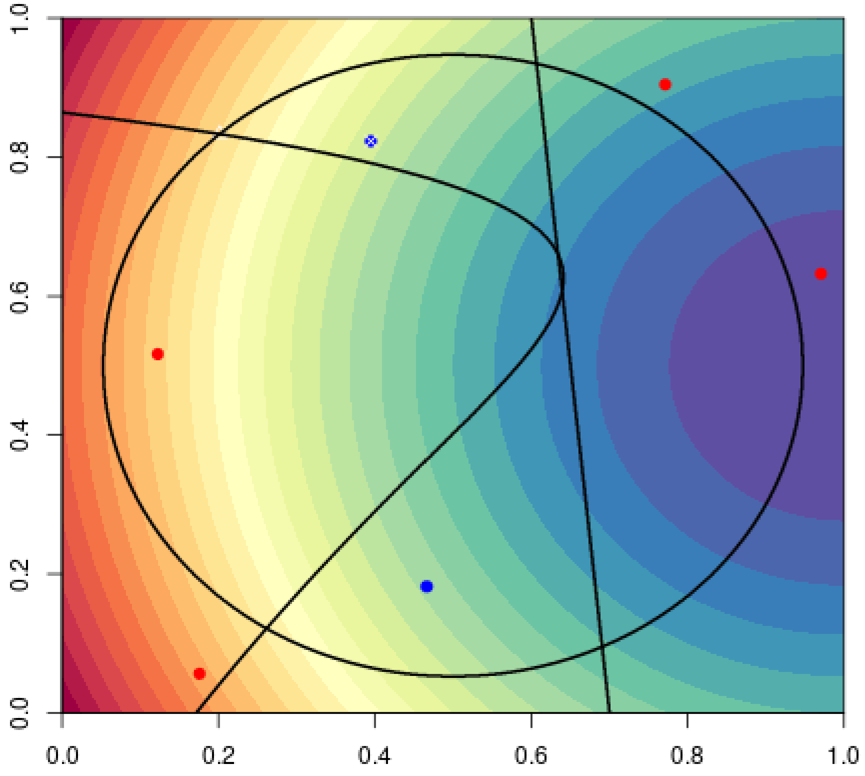


with additional constraint on GPC

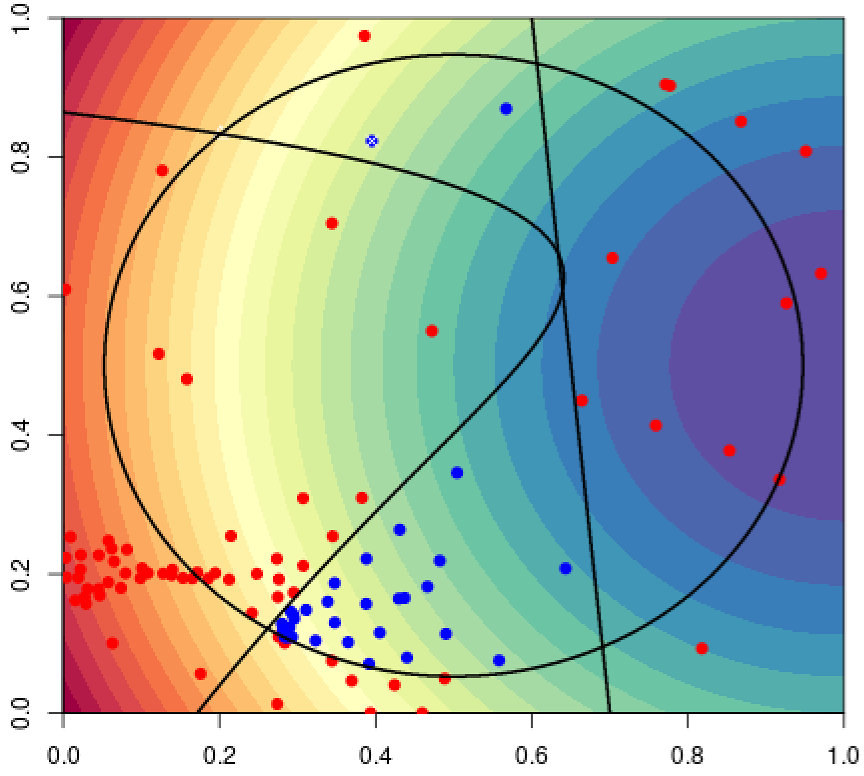
EGO WITH SURROGATE VALUES FOR NAN (SECOND EXAMPLE)

Initial DOE 10

Initial points

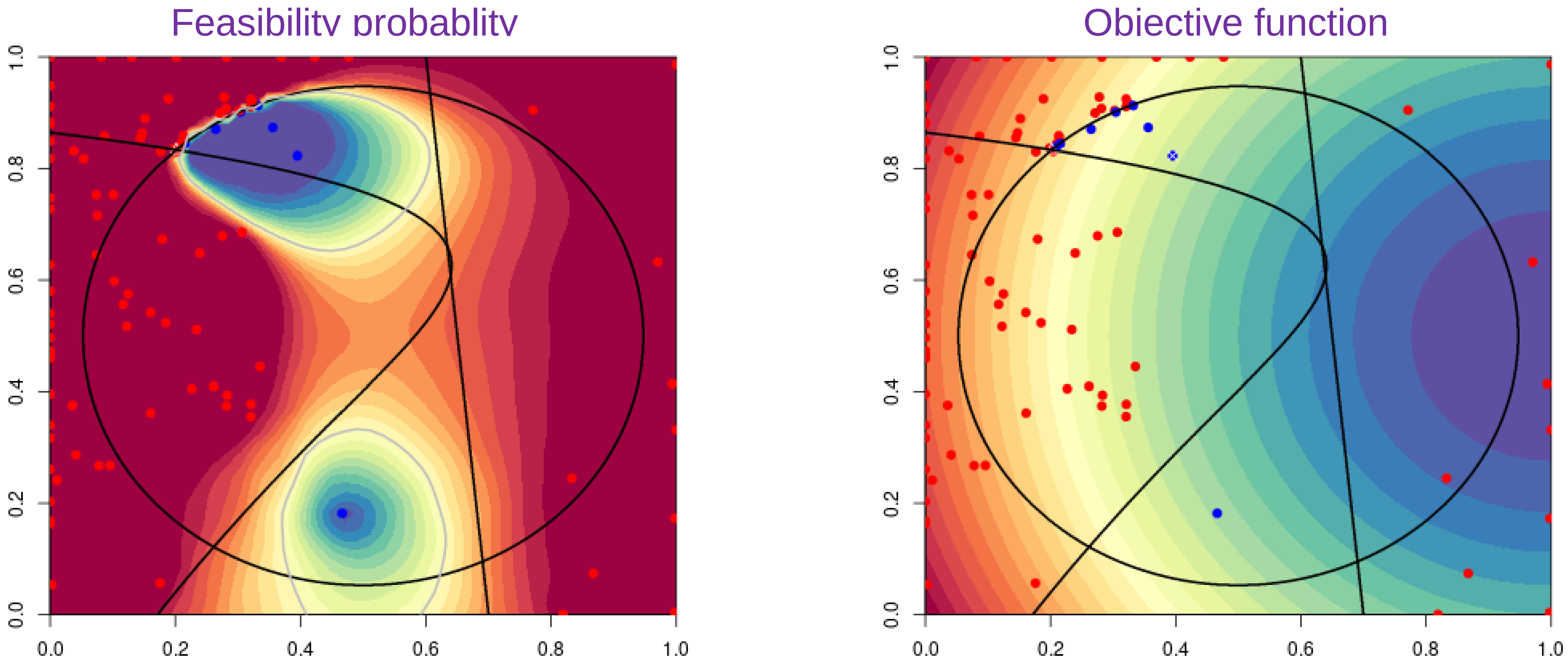


All iterates



EGO WITH ADDITIONAL CONSTRAINT ON GPC (SECOND EXAMPLE)

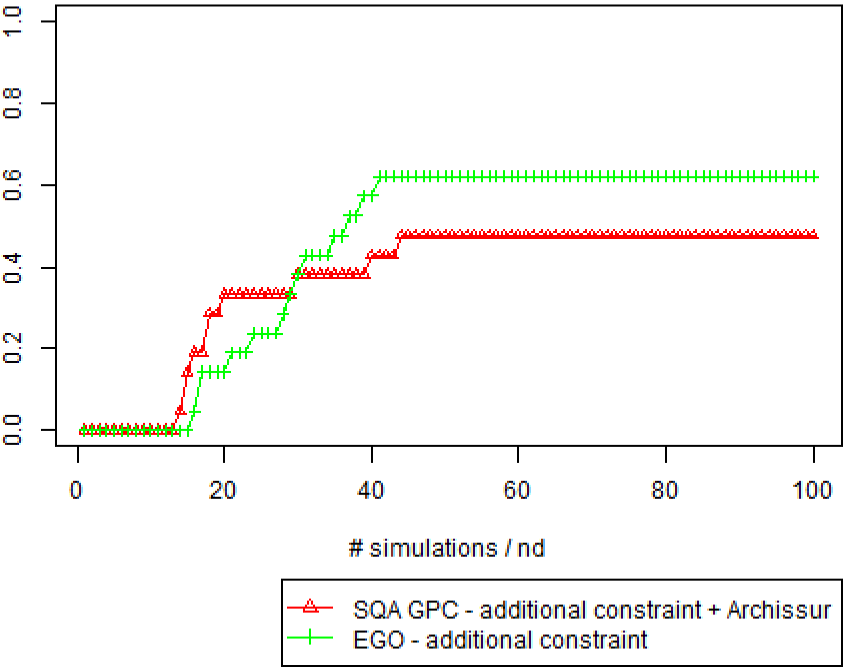
Initial DOE 10



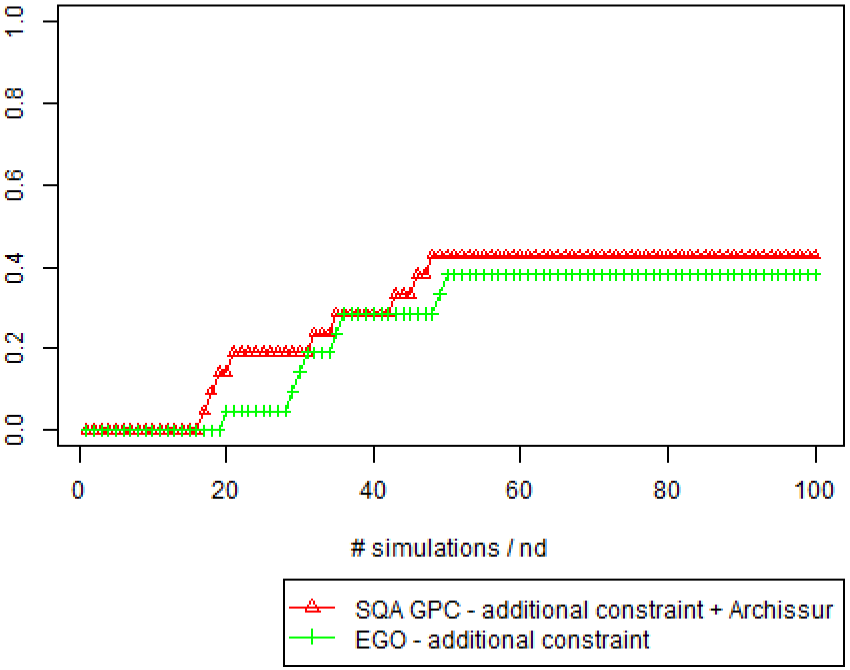
COMPARISON OF EGO AND SQA (BEST VARIANTS)

Data profiles [More and Wild, 2009] for various accuracies on the objective function

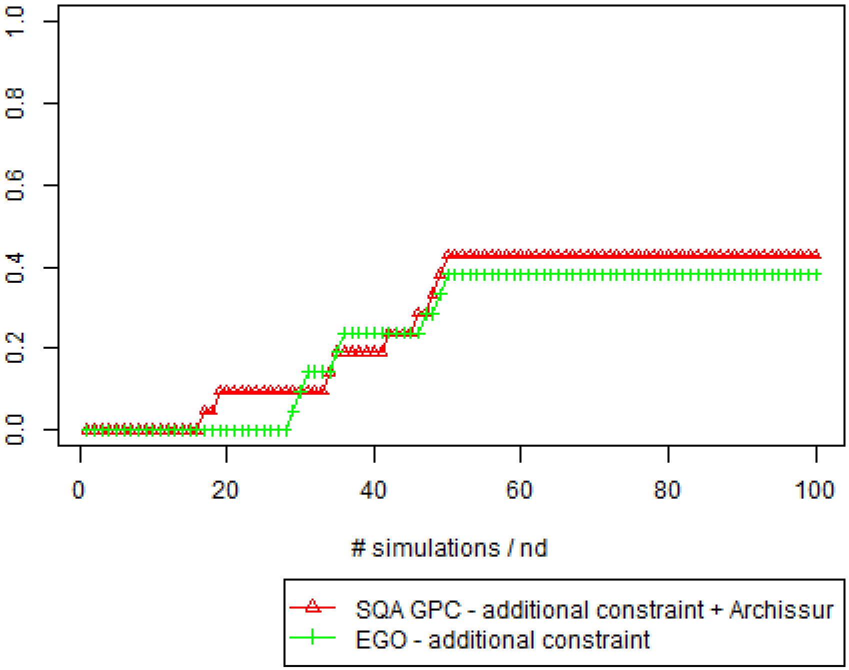
$\tau = 10^{-2}$



$\tau = 10^{-3}$



$\tau = 10^{-5}$

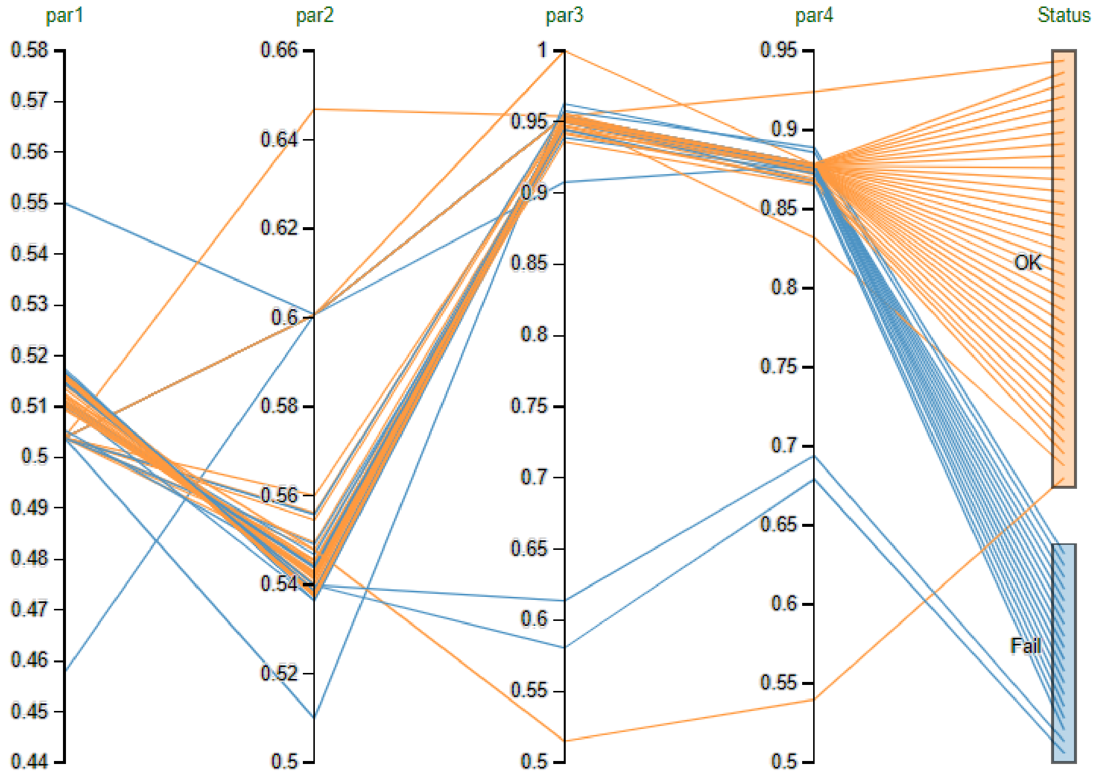


OUTLINE

- Strategies to handle hidden constraints in optimization
- Coupling with various optimization methods
  - Mesh Adaptive Direct Search (NOMAD)
  - Trust Region Derivative Free optimization method (SQA)
  - Bayesian Optimization
- Application to a calibration problem

CALIBRATION OF THERMODYNAMIC MODELS

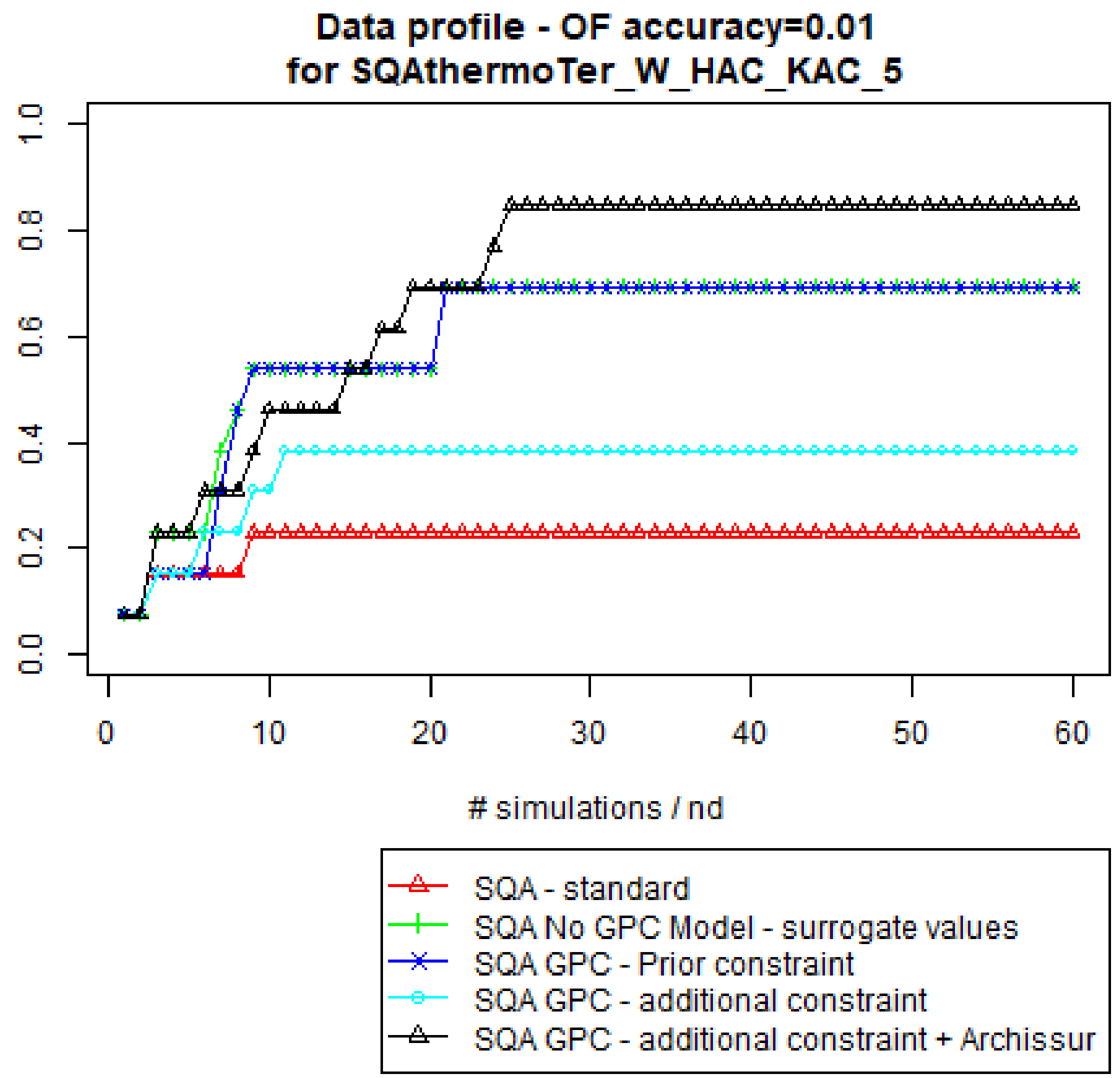
- ENRTL* model calibration for a mixture of water and methanol with partial pressure experimental data with 4 parameters
- Numerical instabilities in the model produce NaN outputs

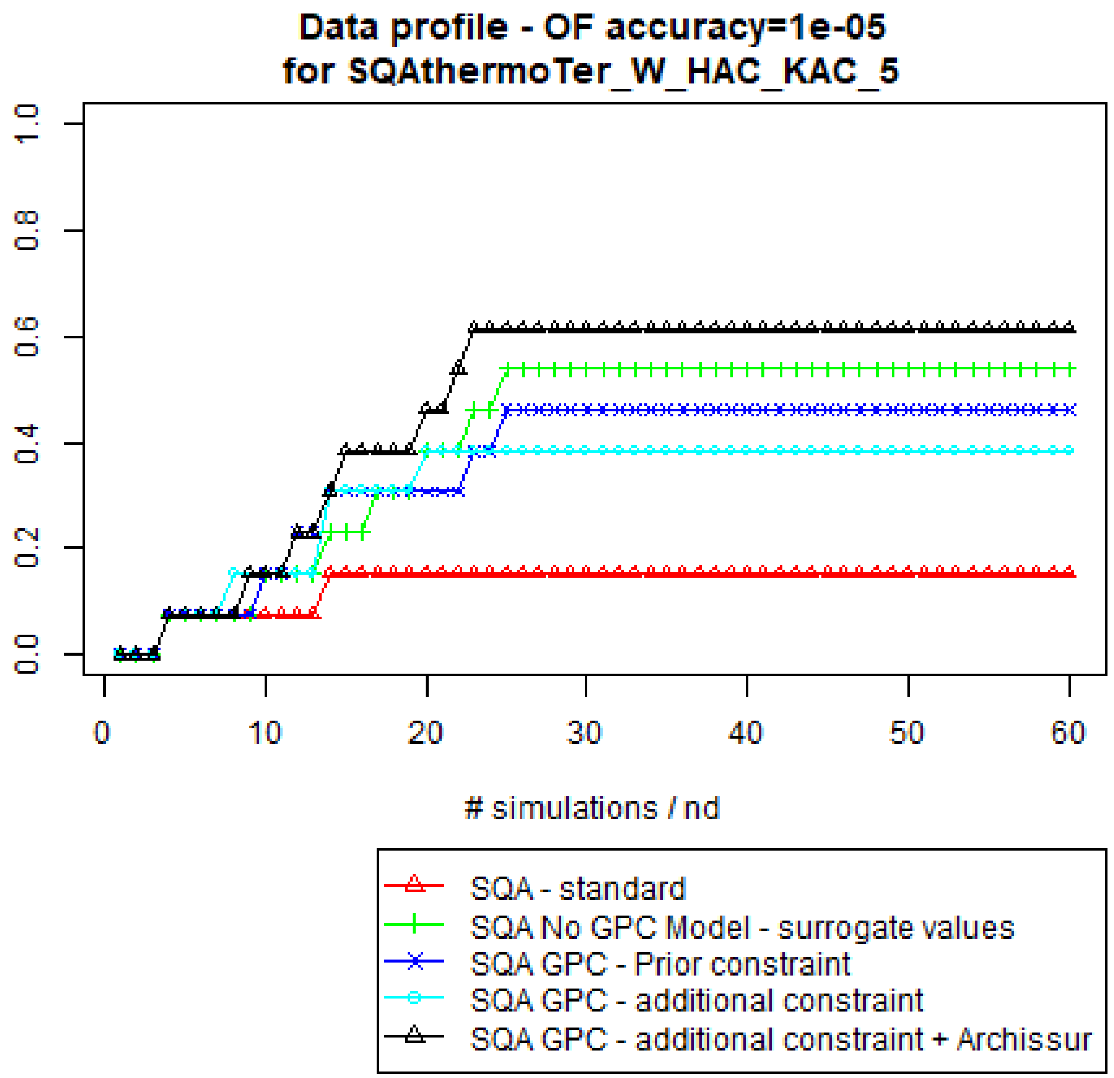


* ENRTL : Electrolyte Non-Random Two Liquid

CALIBRATION OF THERMODYNAMIC MODELS (4 PARAMETERS)

Results with SQA (10 initial points)





CONCLUSIONS

- Active learning Archissur method has a good potential to learn disconnected feasible sets defined by hidden constraints [Menz et al, 2022, hal-03848238]

- The GPC model of hidden constraint is useful in the optimization context to help and speed-up convergence

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

PERSPECTIVES

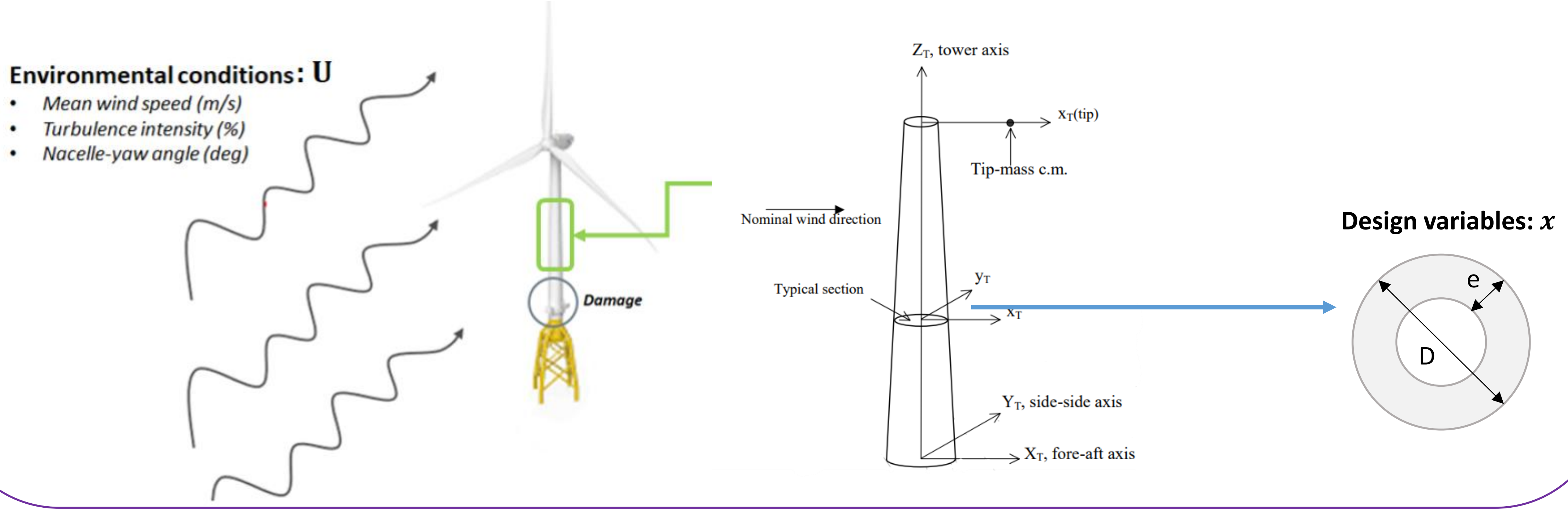
- Refine the heuristics for tuning the different learning strategies in the optimization process
- High dimensional problems

PERSPECTIVES

- Refine the heuristics for tuning the different learning strategies in the optimization process
- High dimensional problems
- A challenging application in progress: robust design of a wind turbine

Tower design with minimal mass that satisfies a reliability constraint for various wind conditions

$$\min_{x} m(x) \quad \text{s.t.} \;\; \mathbb{P}(x) := \mathbb{P}_U[d(x,U) \leq d_{max}] \geq 0.95$$



PERSPECTIVES

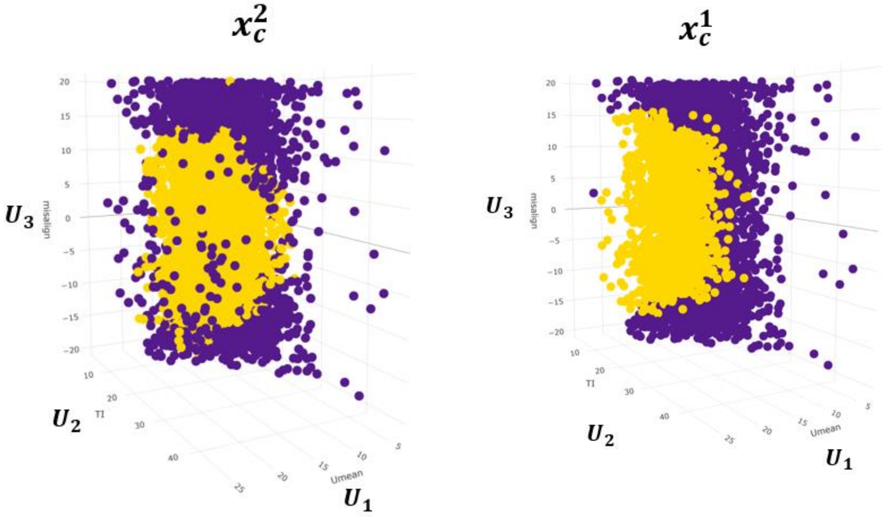
- Refine the heuristics for tuning the different learning strategies in the optimization process
- High dimensional problems
- A challenging application in progress: robust design of a wind turbine

Tower design with minimal mass that satisfies a reliability constraint for various wind conditions

$$\min_x m(x) \quad \text{s.t. } \mathbb{P}(x) := \mathbb{P}_U[d(x,U) \leq d_{max}] \geq 0.95$$

Simulations success/ failure for two different designs x_c and a sample of U

$\exists(x_c, u_c) \;/\; d(x_c, u_c) = NaN$



CODES AND PUBLICATION

- **Publication on Archissur method** : Menz et al, 2023, hal-03848238
- **Opensource codes**
  - GPC model is available in a R opensource package : 10.32614/CRAN.package.GPCsign
  - Package Archissur will also be published soon on CRAN website
- **Integration in opensource platform LAGUN** is also planned
  https://gitlab.com/drti/lagun
  Lagun is a R/Shiny platform providing a user-friendly interface to methods and algorithms dedicated to data exploration, optimization and uncertainty quantification
- Partial funding by the french research agency anr SAMOURAI

Innovating for energy

Find us on:

www.ifpenergiesnouvelles.com

@IFPENinnovation

delphine.sinoquet@ifpen.fr

IFP Energies nouvelles