Conception de systèmes à partir d’expériences numériques coûteuses

Emmanuel Vazquez

SUPELEC, Gif-sur-Yvette, France

AS PEN
Analyse de Sensibilité,
Propagation d’incertitudes et Exploration Numérique
de modèles en sciences de l’environnement

Les Houches
9 mai 2014
Outline

1. Computer models in engineering

2. How to construct a good estimation procedure?
   - Estimation from computer experiments
   - The problem with local optimization methods
   - Worst-case strategies
   - Lipschitzian optimization

3. Average-case approach to the problem of optimization
   - Main ideas
   - Expected Improvement

4. Estimation of a probability of failure
   - Main ideas
   - Optimal one-step lookahead strategy
   - Estimators of the probability of failure
   - One-dimensional illustration

5. Summing up
1. Computer models in engineering
Context overview

Model implemented under the form of a computer program (e.g., a finite element model).
A single run of the program may be time- and resource-consuming.
Computer models in engineering

- \( X \subseteq \mathbb{R}^d \) \( \leadsto \) factor/parameter space of the system
- \( f : X \to \mathbb{R} \) \( \leadsto \) performance or cost function (function of the outputs of the system)

Main classes of problems:

1. **Approximation** of the performance of a system, from expensive evaluations
   \[ x_i \mapsto f(x_i), \ 1 \leq i \leq N, \text{ on a domain of interest} \]

2. **Optimization** of the performance of a system, cost minimization...

   \[ x^* = \arg\max_{x \in X} f(x) \ \text{ or } \ x^* = \arg\min_{x \in X} f(x) \]

3. In presence of uncertain factors, estimation of a probability of failure

   \( \alpha^u(x) := P_{X}\{x \in X : f(x) > u\} \)

   where \( P_{X} \) is some probability distribution on \((X, B(X))\)

**NB:** this is a simplified view \( \rightarrow \) most real problems have several performance functions, and mix different objectives
Example 1/2 – Risk analysis

- Computer simulations to assess the probability of undesirable events

- A serious accident: loss of coolant in a pressurized water nuclear reactor

- Under these conditions, temperature of fuel rods can be described by ~ 50 dimensioning factors, which are not known accurately

- Peak temperature can be estimated using complex and time-consuming simulations

- $f : \mathbb{X} \to \mathbb{R}$ peak temp. as a function of dim. factors

- Objective: estimate a probability of exceeding a critical value

$$\alpha = P_X\{f \geq u\}$$

or a worst-case

$$M = \sup_{x \in X} f(x)$$

(Courtesy of CEA)
Example 2/2 – Design optimization

- Computer simulations to design a product or a process, in particular
  - to find the best feasible values for design parameters (optimization problem)
  - to minimize the probability of failure of a product

- To comply with European emissions standards, the design parameters of combustion engines have to be carefully optimized

- The shape of intake ports controls airflow characteristics, which have direct impact on
  - the performances of the engine
  - emissions of NO\textsubscript{X} and CO

- \( f : \mathbb{X} \subset \mathbb{R}^d \rightarrow \mathbb{R} \) performance as a function of design parameters \((d = 20 \sim 100)\)

- Computing \( f(x) \) takes 5 \sim 20 hours

- Objective: estimate \( x^* = \arg\max_x f(x) \), or \( x^* = \arg\max_x f(x) \) subject to \( \Pr\{\text{pollutant emissions} \leq \text{threshold}\} > \gamma \)

Simulation of an intake port (Navier-Stokes equations) (courtesy of Renault)
Distinct properties of computer experiments

- The performance/cost function \( f : X \subseteq \mathbb{R}^d \rightarrow \mathbb{R} \) is only known through pointwise evaluations.

- An evaluation of \( f \) is called a computer experiment. It consists in:
  - choosing an \( x \in X \)
  - running one or several deterministic computer programs to obtain the value \( f(x) \)

- \( \nabla f \) may also be known in rare cases.

- The factor space \( X \) may be high-dimensional (typically 10 ∼ 100).

- Evaluation of \( f \) may be expensive (e.g., several hours)
  - budget of experiments is limited (typically < 1000)
2. How to construct a good estimation procedure?
Estimation from computer experiments

Let \( f : \mathbb{X} \to \mathbb{R} \) be a continuous function defined on a compact domain \( \mathbb{X} \) with non-empty interior

(\( f \) corresponds to a computer program whose output is not a closed-form expression of the inputs.)

Objective: from a set of computer experiments, obtain an approximation of

\[
    f : \mathbb{X} \to \mathbb{R}
\]

or

\[
    m(f) = \min_{x \in \mathbb{X}} f(x) = f(x^*)
\]

or

\[
    \alpha^u(f) = \mathbb{P}(f > u) = \int_{\mathbb{X}} 1_{f > u} \, d\mathbb{P}_{\mathbb{X}}
\]

... 

The result of a pointwise evaluation of \( f \) carries information about \( f \) and quantities depending on \( f \) (in particular, \( m(f) \), \( \alpha^u(f) \)...) 

Expensive computer experiments: the number of evaluations is limited \( \to m(f), \alpha^u(f) \), etc. must be estimated using a fixed number, say \( N \), of evaluations of \( f \).
The case of optimization

- In the context of rare events estimation and risk analysis, it is often desirable to assess the worst-case performance of a system, that is, to determine

\[ M = \sup_{x \in \mathcal{X}} f(x) \]

or

\[ m = \inf_{x \in \mathcal{X}} f(x) \]

- \( f \) may be non-convex
- this is a global optimization problem

- How to design a good optimization algorithm?

- In a context of risk analysis, and also in difficult economic environments, we want to use an optimization algorithm that will provide a robust estimation of the global optimum
Why local optimization methods may not be satisfactory in the domain of computer experiments?

- An illustrative example: consider
  \[ f : \mathbb{R}^2 \rightarrow \mathbb{R} \]
  \[ x \mapsto f(x) = \exp(1.8(x[1] + x[2])) + 5x[1] + 6x[2]^2 + 3\sin(4\pi x[1]) \]

- Objective: find an approximation of
  \[ x^* = \arg\min_{x \in [-1,1]^2} f(x). \]

  with a budget of \( N = 60 \) experiments
Illustrative example (continued)

Evaluations points using a Nelder-Mead algorithm

→ the algorithm converges to a local minimum (≈ 0.427)
This comes as no surprise (local search algorithm). But above all...

- after having spent the budget of (possibly expensive) evaluations, the behavior of the function is only known in a small region of the search domain
- the global behavior of the function is unknown
- potentially interesting regions have not been explored
How to construct a good estimation procedure?

This comes as no surprise (local search algorithm). But above all...

- after having spent the budget of (possibly expensive) evaluations, the behavior of the function is only known in a small region of the search domain

- the global behavior of the function is unknown
- potentially interesting regions have not been explored
How to construct a good estimation procedure?

The problem with local optimization methods

This comes as no surprise (local search algorithm). But above all...

- after having spent the budget of (possibly expensive) evaluations, the behavior of the function is only known in a small region of the search domain

- the global behavior of the function is unknown
- potentially interesting regions have not been explored
This comes as no surprise (local search algorithm). But above all...

- after having spent the budget of (possibly expensive) evaluations, the behavior of the function is only known in a small region of the search domain

- the global behavior of the function is unknown
- potentially interesting regions have not been explored
In a context of expensive-to-evaluate functions and a small budget of evaluations, it seems “safer” to achieve a balance between **local search** and **exploration** of the search domain.

Uniform random sampling:

→ minimum of evaluation results is $\approx -5.823$ (global minimum is $\approx -5.845$)

What is a **robust** optimization strategy? How to obtain such a strategy?
How to construct a good estimation procedure?

The problem with local optimization methods

- In a context of expensive-to-evaluate functions and a small budget of evaluations, it seems “safer” to achieve a balance between local search and exploration of the search domain

- Uniform random sampling:

  → minimum of evaluation results is \( \approx -5.823 \) (global minimum is \( \approx -5.845 \))

- What is a robust optimization strategy? How to obtain such a strategy?
In a context of expensive-to-evaluate functions and a small budget of evaluations, it seems “safer” to achieve a balance between local search and exploration of the search domain.

Uniform random sampling:

→ minimum of evaluation results is $\approx -5.823$ (global minimum is $\approx -5.845$)

What is a robust optimization strategy? How to obtain such a strategy?
In a context of expensive-to-evaluate functions and a small budget of evaluations, it seems “safer” to achieve a balance between local search and exploration of the search domain.

Uniform random sampling:

→ minimum of evaluation results is $\approx -5.823$ (global minimum is $\approx -5.845$)

What is a robust optimization strategy? How to obtain such a strategy?
The worst-case approach

- Let $\mathcal{A}_N$ be the class of all optimization strategies $X_N$ that query sequentially $N$ evaluations of $f$ at points $X_1, \ldots, X_N$.
- Define the error of approximation of a strategy $X_N \in \mathcal{A}_N$ on $f$ as

  $$\varepsilon(X_N, f) = \hat{m}_N(f) - m(f)$$

  with $\hat{m}_N(f) = f(X_1) \land \cdots \land f(X_N)$

- Assume that $f$ belongs to a class of functions $\mathcal{F} \rightarrow \text{prior information}$

→ A first idea to define a notion of a good strategy is to consider robustness with respect to a worst case
  - Define the minimax risk

  $$r_{\text{minimax}}(\mathcal{F}) = \inf_{X_N \in \mathcal{A}_N} \sup_{f \in \mathcal{F}} \varepsilon(X_N, f)$$

  - A strategy $X_N^*$ that attains $r_{\text{minimax}}(\mathcal{F})$ is called an (optimal) minimax strategy
  - $X_N^*$ has the best worst-case performance on $\mathcal{F}$
Example of a minimax strategy: case of Lipschitz functions

Definition

A function $f : X \to \mathbb{R}$ is called Lipschitz continuous if there exists a real constant $K \geq 0$ such that, for all $x_1$ and $x_2$ in $X$,

$$|f(x_1) - f(x_2)| \leq K\|x_1 - x_2\|.$$ 

Any such $K$ is referred to as a Lipschitz constant for the function $f$. 
Example of a minimax strategy: case of Lipschitz functions

- Let $\mathcal{F}$ be the class of all Lipschitz continuous functions $X \to \mathbb{R}$, with Lipschitz constant $K$
- Assume that $f \in \mathcal{F}$
- For any strategy $X_N$, define the fill distance as

$$h_N = \sup_{x \in X} \min_{i=1, \ldots, N} |x - X_i|$$

- For any $X_N \in A_N$ and any $f \in \mathcal{F}$,

$$\varepsilon(X_N, f) = f(X_1) \land \cdots \land f(X_N) - f(x^*) \leq f(X_i^*) - f(x^*) \leq Kh_N,$$

where $X_i^*$ is the nearest point to $x^*$
- Thus, for any $X_N \in A_N$, $\sup_{f \in \mathcal{F}} \varepsilon(X_N, f) \leq Kh_N$
- For any $X_N$, there exists a function $f \in \mathcal{F}$ such that

$$\varepsilon(X_N, f) = Kh_N$$

Thus,

$$\sup_{f \in \mathcal{F}} \varepsilon(X_N, f) = Kh_N$$
Example of a minimax strategy: case of Lipschitz functions

- Consequence: a minimax strategy minimizes $h_N$
  - sample points have to be uniformly distributed over the search domain

- In dimension one:
  - for any $X_N$, $h_N \geq \frac{|X|}{(N+1)}$
  - the optimal strategy is the uniform sampling: $r_{\text{minimax}}(\mathcal{F}) = K \frac{|X|}{(N+1)}$

- How to deal dimension $d > 1$?
  - using a uniform grid is not optimal (not mentioning the fact that the budget of evaluations must be at least $N = 2^d$)
  - sampling randomly with a uniform distribution over $X$ provides no guarantee that $h_n$ will be small
  - optimizing the design of experiments to yield a small $h_n$ is interesting but may numerically expensive
  - Maximin Latin Hypercube Sampling [McKay, Conover and Beckman (1979)] is an easy procedure that will generally provide good suboptimal designs
Example of a maximin Latin hypercube sampling of size $n = 100$ in dimension $d = 8$
The worst-case approach

- Consequence: for Lipschitz continuous functions, the minimax strategy consists in having sample points uniformly distributed over the search domain.
- Here, the optimal strategy is non-adaptive!
- It may be more satisfying to achieve a balance between exploration of the search domain and local search in promising regions (good performance on worst cases and good convergence rate).
- Worst-case setting: appropriate framework to assess the performance of an optimization algorithm?
- We need to know how an optimization algorithm performs for “typical” functions $f$ not corresponding to worst cases.
- A classical approach is to adopt an average-case point of view.
3. Average-case approach to the problem of optimization
Average-case approach

- Average-case → introduction of a probability space $(\Omega, \mathcal{B}, P_0)$
- We consider methods where $f$ is seen as a sample path of a real-valued random process $\xi$ defined on $(\Omega, \mathcal{B}, P_0)$ with parameter in $\mathbb{X}$
  $\implies$ there exists $\omega \in \Omega$ such that
  \[ f = \xi(\omega, \cdot) \]
- From a Bayesian decision-theoretic point of view, $\xi$ represents prior knowledge about $f$

- A good strategy is a strategy that achieves, or gets close to, the optimal average risk
  \[ r_{\text{average}} := \inf_{\mathcal{X}_N \in \mathcal{A}_N} E_0(\epsilon(\mathcal{X}_N, \xi)) \]
  where $E_0$ denotes the expectation with respect to $P_0$
Expected Improvement [Mockus et al. 78, Schonlau et al. 96, Jones et al. 98]

- The optimal Bayesian one-step lookahead strategy for the problem of optimization corresponds to choosing each new evaluation point according to

\[
X_{n+1} = \arg\min_{x \in \mathcal{X}} E_n (\hat{m}_{n+1} - m \mid X_{n+1} = x)
\]

\[
= \arg\min_{x \in \mathcal{X}} E_n (\hat{m}_{n+1} \mid X_{n+1} = x)
\]

\[
= \arg\min_{x \in \mathcal{X}} E_n (\hat{m}_{n} \land \xi(X_{n+1}) \mid X_{n+1} = x)
\]

\[
= \arg\min_{x \in \mathcal{X}} E_n (0 \land (\xi(X_{n+1}) - \hat{m}_{n}) \mid X_{n+1} = x)
\]

\[
= \arg\max_{x \in \mathcal{X}} \rho_n(x) := E_n ((\hat{m}_{n} - \xi(X_{n+1}))_{+} \mid X_{n+1} = x)
\]

with

- \(E_n\) conditional expectation wrt \(\xi(X_1), \ldots, \xi(X_n)\)
- \(\hat{m}_{n} = \xi(X_1) \land \cdots \land \xi(X_n)\)
- \(z_+ = \max(z, 0)\)

- The sampling criterion \(\rho_n\) is the expected improvement (EI)
  - average excursion of \(\xi(x)\) below the current minimum of past evaluation results

- A well-known Bayesian optimization algorithm
  - proposed by Mockus et al.
  - popularized by the EGO algorithm of Jones et al.
Expected Improvement [Mockus 78, Schonlau et al. 96, Jones et al. 98]

- Assume $\xi$ is a Gaussian process, with known mean and covariance functions
- Then, $\rho_n(x)$ has a closed-form expression:

$$
\rho_n(x) = \gamma \left(m_n - \hat{\xi}_n(x; X_n), \sigma_n^2(x)\right),
$$

where

$$
\gamma(z, s) = \begin{cases} 
\sqrt{s} \Phi' \left(\frac{z}{\sqrt{s}}\right) + z \Phi \left(\frac{z}{\sqrt{s}}\right) & \text{if } s > 0, \\
\max(z, 0) & \text{if } s = 0.
\end{cases}
$$

and $\hat{\xi}_n(x; X_n)$ and $\sigma_n^2(x)$ are the kriging predictor and the kriging variance of $\xi(x)$ (Matheron, 1960) → see illustrating figure below.

- The EI algorithm:

$$
\begin{cases} 
X_1 = x_{\text{init}}, \\
X_{n+1} = \arg\max_{x \in X} \rho_n(x), \quad n \geq 1,
\end{cases}
$$
Global optimization based on EI

\[
\log_{10} \rho_n(x)
\]
Global optimization based on EI
Global optimization based on EI
Global optimization based on EI

\[
\rho_n(x) = \log_{10}(\rho_n(x))
\]
Global optimization based on EI

\[ \log_{10} \rho_n(x) \]

\[ x \in [-1, 1] \]
Global optimization based on EI

\[ \log_{10} \rho_n(x) \]

\[ E. \text{Vazquez} \]

ASPEN, Les Houches

9 mai 2014 26 / 38
Global optimization based on EI
Global optimization based on EI
Global optimization based on EI

\[ \log_{10} \rho_n(x) \]

\[ x \]

E. Vazquez
ASPEN, Les Houches
9 mai 2014 26 / 38
Global optimization based on EI
Global optimization based on EI

- Average-case approach to the problem of optimization
- Expected Improvement

![Graph showing Expected Improvement](image-url)
Global optimization based on EI
Global optimization based on EI
EI/EGO: 2D illustration
(f defined on Slide 12)

NB: Global minimum found by the EI algorithm in only 31 evaluations (abs. tol. $1.10^{-4}$)
Global optimization based on EI

- Particularly interesting in the context of expensive-to-evaluate functions, very useful and effective in practical situations

- A great number of applications can be found in the literature (aeronautics, chemistry, energy...)

- Some theoretical results on the convergence of these algorithms: (Vazquez & Bect 2010, Bull 2011...)
4. Estimation of a probability of failure
Estimation of a probability of failure

Main ideas

- **Objective**: to obtain an approximation of

\[ \alpha_u(f) = P_X\{f > u\} = \int_X 1_{f > u} dP_X \]

from a small set of computer experiments

- An algorithm to estimate \( \alpha_u(f) \) using \( N \) evaluations is a pair \((X_N, \hat{\alpha}_N)\), where \( X_N \) is a strategy, such that

\[ X_N : f \mapsto X_N(f) = (X_1(f), X_2(f), \ldots, X_N(f)) \in X^N, \]

and \( \hat{\alpha}_N(f) \) is an estimator of \( \alpha_u(f) \).

- How to construct a good algorithm?
Main ideas (2/3)

- Assume that an estimator \( \hat{\alpha}_N \) has been chosen (see how later)
- How to construct the strategy \( X_N \)?
- Given a loss function

\[
L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}
\]

define the error of approximation of a strategy \( X_N \in \mathcal{A}_N \) on \( f \) as

\[
\epsilon(X_N, f) = L(\hat{\alpha}_N(f), \alpha(f))
\]

- Here, we shall consider the quadratic loss function, so that

\[
\epsilon(X_N, f) = (\hat{\alpha}_N(f) - \alpha(f))^2
\]
Main ideas (3/3)

- We adopt a **Bayesian approach**: $f$ is considered as a sample path of a real-valued random process $\xi$ defined on some probability space $(\Omega, \mathcal{B}, P_0)$ with parameter $x \in \mathbb{X}$.

- A **good strategy** is a strategy that gets close to the optimal **average risk**

$$r_{\text{average}} := \inf_{X_N} E_0 (\epsilon(X_N, \xi))$$

where $E_0$ denotes the expectation with respect to $P_0$.
For \( N = 1 \), given an estimator \( \hat{\alpha}_1 \) of \( \alpha \), the optimal Bayesian strategy is

\[
\begin{align*}
X_0 & = x_{\text{init}} \in \mathbb{X} \\
X_1 & = \text{argmin}_{x \in \mathbb{X}} E_0 ( (\hat{\alpha}_1 - \alpha)^2 \mid X_1 = x )
\end{align*}
\]

For \( N > 1 \)?

A natural and straightforward strategy consists in selecting greedily each evaluation as if it were the last one (at each step, we do as if \( N = 1 \))
Optimal one-step lookahead strategy (2/2)

- Let $E_n$, $n = 1, 2, \ldots$, denote the conditional expectation with respect to the past evaluations.

- **One-step lookahead strategy**: given a sequence of estimators $(\hat{\alpha}_n)_{n \geq 1}$, at step $n$, choose the next evaluation point according to

  $$X_{n+1} = \arg\min_{x \in X} J_n(x)$$

  where $J_n$ is a sampling criterion such that

  $$J_n(x) = E_n \left( (\hat{\alpha}_{n+1} - \alpha)^2 \mid X_{n+1} = x \right).$$

- We call such a strategy a stepwise uncertainty reduction (SUR) strategy


- Restricting $\xi$ to be a Gaussian process makes it possible to derive estimators for $\alpha$ and to compute $J_n$ with moderate computational efforts.
Estimators of the probability of failure under a Gaussian prior

- Given a random process $\xi$ and a strategy $X_N$, the optimal estimator that minimizes $\mathbb{E}_0 \left( (\alpha - \hat{\alpha}_n)^2 \right)$ among all $\mathcal{F}_n$-measurable estimators $\hat{\alpha}_n$, $1 \leq n \leq N$, is

$$
\hat{\alpha}_n = \mathbb{E}_n (\alpha) = \mathbb{E}_n \left( \int_X 1_{\xi > u} \, dP_X \right) = \int_X p_n \, dP_X, \quad (1)
$$

where

$$
p_n : x \in X \mapsto P_n \{ \xi(x) > u \}.
$$

- When $\xi$ is a Gaussian process, the probability $p_n(x)$ of exceeding $u$ at $x \in X$, given $\mathcal{I}_n$, has a simple closed-form expression:

$$
p_n(x) = 1 - \Phi \left( \frac{u - \hat{\xi}_n(x)}{\sigma_n(x)} \right) = \Phi \left( \frac{\hat{\xi}_n(x) - u}{\sigma_n(x)} \right),
$$

with $\Phi$ the cdf of the normal distribution.

- Thus, in the Gaussian case, the estimator (1) is amenable to a numerical approximation, by integrating the excess probability $p_n$ over $X$.

- Other choices can be made.
Illustration in dimension one

- Unknown \( f \), threshold \( u \), and pdf. \( \frac{dP_X}{dx} \) over \( X \)
- Initial design
- Construction of \( f_n \), confidence intervals, probability of excursion \( P\{f(x) \geq u\}, x \in X \)
- Computation and minimization of \( J_n \)
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $\frac{dP_X}{dx}$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown \( f \), threshold \( u \), and pdf. \( dP_x/dx \) over \( X \)
- Initial design
- Construction of \( f_n \), confidence intervals, probability of excursion \( P\{f(x) \geq u\}, x \in X \)
- Computation and minimization of \( J_n \)
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown \( f \), threshold \( u \), and pdf. \( dP_x/dx \) over \( X \)
- Initial design
- Construction of \( f_n \), confidence intervals, probability of excursion \( P\{f(x) \geq u\}, x \in X \)
- Computation and minimization of \( J_n \)
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_{X}/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown \( f \), threshold \( u \), and pdf. \( dP_X/dx \) over \( X \)
- Initial design
- Construction of \( f_n \), confidence intervals, probability of excursion \( P\{f(x) \geq u\}, x \in X \)
- Computation and minimization of \( J_n \)
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown \( f \), threshold \( u \), and pdf. \( dP_X/dx \) over \( X \)
- Initial design
- Construction of \( f_n \), confidence intervals, probability of excursion \( P\{f(x) \geq u\} \), \( x \in X \)
- Computation and minimization of \( J_n \)
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_x/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $\frac{dP_X}{dx}$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_x/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation

E. Vazquez
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_x/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
**Illustration in dimension one**

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$.
- Initial design.
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$.
- Computation and minimization of $J_n$.
- Position of next evaluation.
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_x/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}$, $x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and pdf. $dP_x/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Illustration in dimension one

- Unknown $f$, threshold $u$, and PDF $dP_X/dx$ over $X$
- Initial design
- Construction of $f_n$, confidence intervals, probability of excursion $P\{f(x) \geq u\}, x \in X$
- Computation and minimization of $J_n$
- Position of next evaluation
Concluding remarks

- In the context of rare events estimation and risk analysis, Bayesian strategies show very good performances with respect to alternative approaches.
- Bayesian strategies can be used for global optimization, estimation of probabilities of failure, quantile estimation...
- In our work, we study both practical and theoretical questions regarding these strategies.
- We try to address effectively industrial needs by taking into account the constraints that appear in real problems.
This talk is based on the following papers:

- Emmanuel Vazquez, Julien Bect, *A sequential Bayesian algorithm to estimate a probability of failure*, 15th IFAC Symposium on System Identification (SYSID 2009), 2009, Saint-Malo, France [clickme]

- Romain Benassi, Julien Bect, Emmanuel Vazquez, *Robust Gaussian process-based global optimization using a fully Bayesian expected improvement criterion*, Learning and Intelligent Optimization, 5th International Conference, LION 5, 2011, Rome, Italy [clickme]
