Robust optimization : formulation, kriging and evolutionary approaches

Rodolphe Le Riche

Robust optimization : formulation, kriging and evolutionary approaches

Rodolphe Le Riche
CNRS and Ecole des Mines de Saint-Etienne

class given as part of the “Modeling and Numerical Methods for Uncertainty Quantification” French-German summer school, Porquerolles, Sept. 2014

Outline of the talk

1. Motivations for robust optimization

2. Formulations of optimization problems with uncertainties

3. Kriging-based approaches (costly functions)

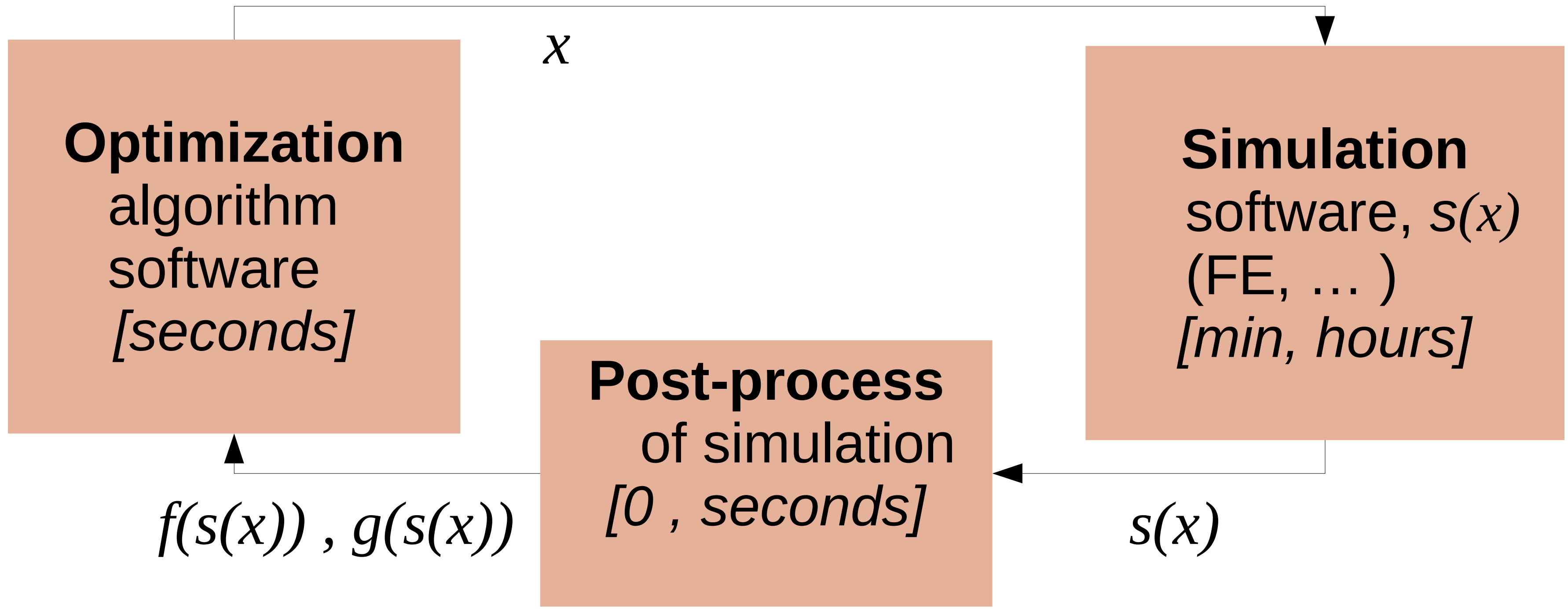
4. Evolutionary approaches (non costly functions)

Why do we optimize ?

Optimization as a mathematical formulation for decision

$$\min_{x \in S} f(x)$$
$$g(x) \leqslant 0$$

followed by a numerical, approximate, resolution,



Communication between programs by file, pipe, messages, ...

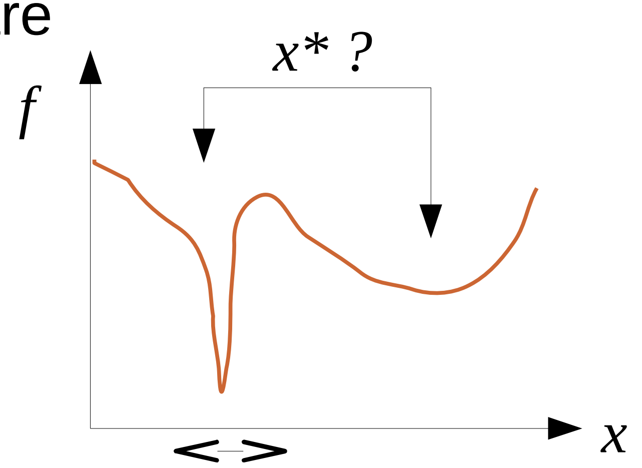
Motivations for robust optimization

What is the point – in practice – for deterministically solving an optimization problem when there are

• unstable optima

• aleatory model (*s*) parameters

• model uncertainties



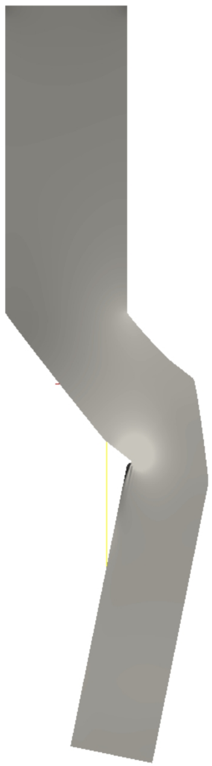
• dynamically changing model conditions (complex systems)

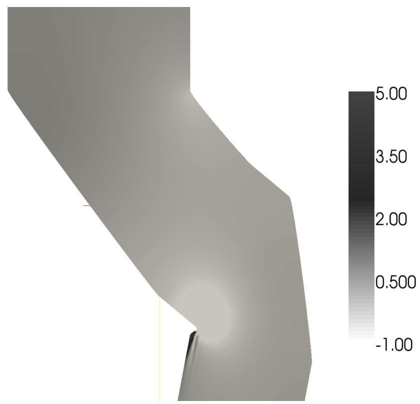
???

Unstable deterministic optimum. Expl of an air duct.

Minimize pressure loss by changing the (parameterized) shape.

deterministic design



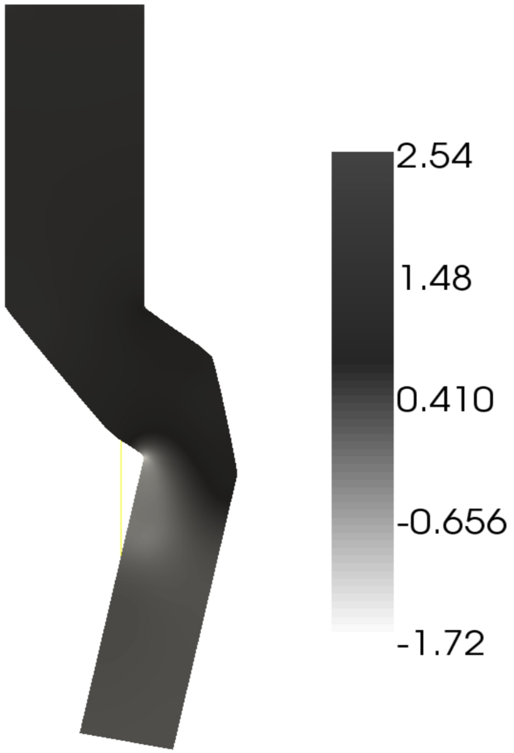


(zoom)

$$\Delta P = 0.604 \underset{\text{mesh}\times 2}{\rightarrow} 2.356$$

$$\Delta P \underset{\text{MC on } u}{=} 3.011 \pm 2.033$$

robust design

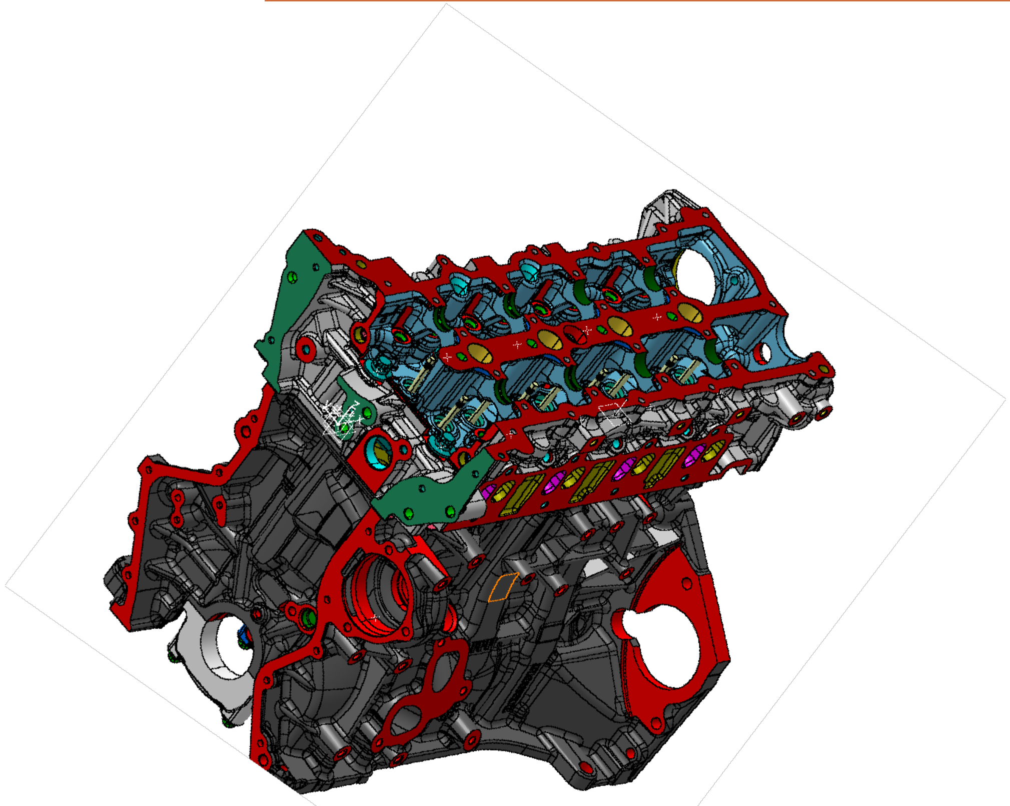


$$\Delta P \underset{\text{MC on } u}{=} 1.198 \pm 0.069$$

The optimization exploits meshing flaws.
The result is not stable w.r.t. mesh or boundary conditions changes.

Cf. J. Janusevskis and R. Le Riche, *Robust optimization of a 2D air conditioning duct using kriging*, technical report hal-00566285, feb. 2011.

Unstable deterministic optimum.
Expl of a combustion engine.



a +/- 1mm dispersion in the manufacturing of a car cylinder head can degrade its performance (g CO_2/km) by -20% (worst case)

Motivations for robust optimization

What is the point – in practice – for deterministically solving an optimization problem when there are

• unstable optima

• aleatory model (*s*) parameters

• model uncertainties

• dynamically changing model conditions (complex systems)

???

→ modify the problem statement, therefore also the optimization algorithms → robust optimization

Outline of the talk

1. Motivations for robust optimization

2. Formulations of optimization problems with uncertainties

3. Kriging-based approaches (costly functions)

4. Evolutionary approaches (non costly functions)

Formulations of optimization problems under uncertainties

H.G. Beyer, B. Sendhoff, Robust Optimization – A comprehensive survey, Comput. Methods Appl. Mech. Engrg, 196, pp. 3190-3218, 2007.

G. Pujol, R. Le Riche, O. Roustant and X. Bay, *L'incertitude en conception: formalisation, estimation,* Chapter 3 of the book *Optimisation Multidisciplinaire en Mécaniques : réduction de modèles, robustesse, fiabilité, réalisations logicielles*, Hermes, 2009 (in French ;-( )

Formulation of optimization under uncertainty
The double (x,U) parameterization

We introduce U, a vector of uncertain (random) parameters that affect the simulator s.
x is a vector of deterministic optimization (controlled) variables. x in S, the search space.

(cf. Taguchi, 80's)

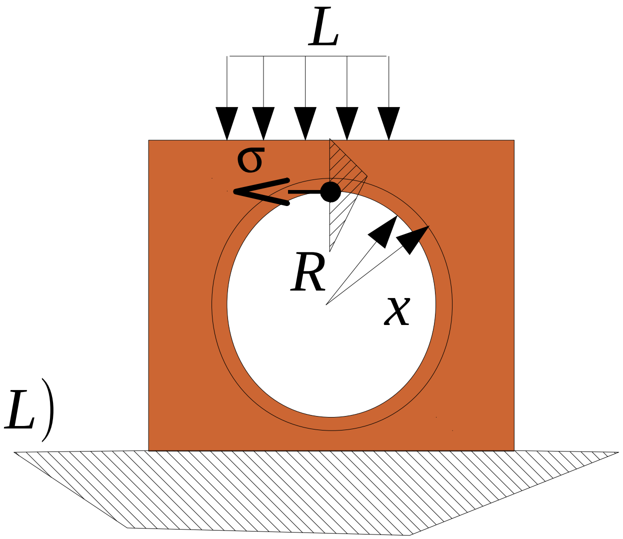
$s(x) \rightarrow s(x,U)$, therefore $f(x) \rightarrow f(s(x,U)) = f(x,U)$
and $g(x) \rightarrow g(s(x,U)) = g(x)$

U used to describe
- noise (as in identification with measurement noise)
- model error (epistemic uncertainty)
- uncertainties on the values of some parameters of s.

The (x,U) parameterization is general

Three cases (which can be combined)

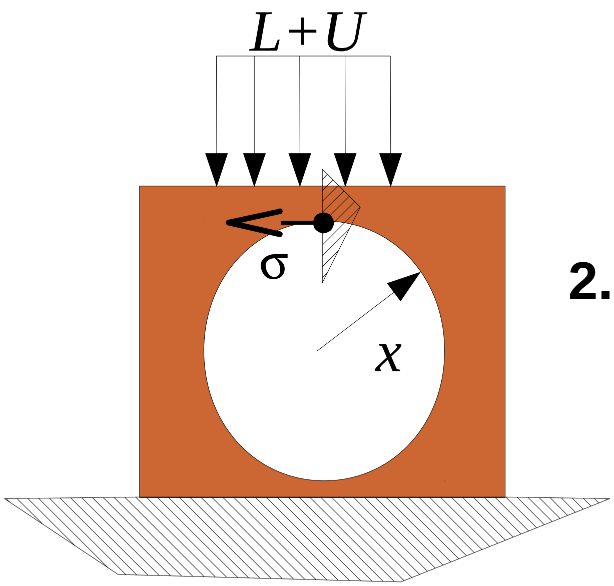
Expl : $f(.) \equiv s(.) \equiv \sigma(R,L)$ (radial stress)



1. Noisy controlled variables

Expl : manufacturing tolerance,

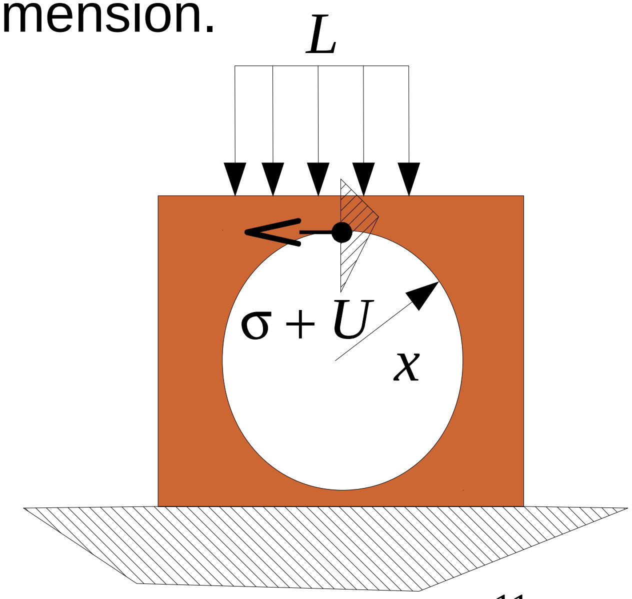
$R \equiv x+U \quad , \quad f(x,U) \equiv \sigma(x+U,L)$



2. Noise exogenous to the optimization variables

Expl : random load $L+U$, x is a dimension.

$f(x,U) \equiv \sigma(x,L+U)$



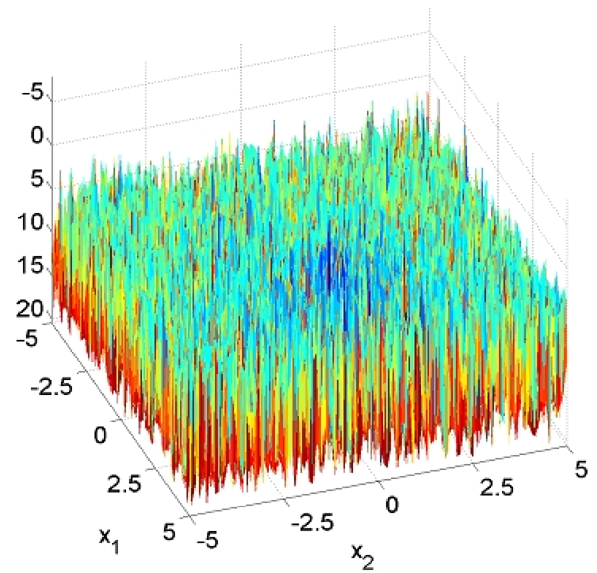
3. Noise as an error model for the simulation

Expl. : $f(x,U) \equiv \sigma(x,L)+U$

Formulation of optimization under uncertainties
(1) the noisy case

Let's not do anything about the uncertainties, i.e., try to solve

$$\min_{x \in S \subset \mathbb{R}^d} f(x, U)$$
$$g(x, U) \leqslant 0$$



It does not look good : gradients are not defined, what is the result of the optimization ?
But sometimes there is no other choice. Ex : *y* expensive with uncontrolled random numbers inside (like a Monte Carlo statistical estimation, numerical errors, measured input).

(2) an ideal series formulation

Replace the noisy optimization criteria by statistical measures

$G(x)$ is the random event "all constraints are satisfied" ,

$$G(x) = \bigcap_i \{ g_i(x, U) \leqslant 0 \}$$

$$\min_{x \in S} q_{\alpha}^{c}(x) \quad \text{(conditional quantile)}$$

such that $P(G(x)) \geqslant 1 - \varepsilon$

where the conditional quantile is defined by

$$P\left(f(x, U) \leqslant q_{\alpha}^{c}(x) \mid G(x)\right) = \alpha$$

(3) simplified formulations often seen in practice

For bad reasons (joint probabilities ignored) or good ones (simpler numerical methods – Gaussian pdf – , lack of data, organization issues), quantiles are often replaced by averages and variances, conditioning is neglected, constraints are handled independently :

$$\min_{x\in S} q_\alpha(x) \quad \text{or} \quad \min_{x\in S} E\left(f(x,U)\right) \quad \text{and / or} \quad \min_{x\in S} V\left(f(x,U)\right)$$

$$\text{or} \quad \min_{x\in S} E\left(f(x,U)\right)+r\sqrt{V\left(f(x,U)\right)}$$

$$\text{where} \quad P\left(f(x,U)\leqslant q_\alpha(x)\right)=\alpha \quad \text{and} \quad r>0$$

$$\text{such that} \quad P\left(G(x)\right)\geqslant 1-\varepsilon \quad \text{or} \quad P\left(g_i(x)\leqslant 0\right)\geqslant 1-\varepsilon_i$$

where ε is the series system risk
and ε_i is the i th failure mode risk

Direct approaches (1/4)

In practice, statistical performance measures are estimated

$$\hat{f}(x) = \left\{ \widehat{E_U}(f(x,U)) \text{ or } \widehat{E_U}(f(x,U)) + r\sqrt{\widehat{V_U}(f(x,U))} \text{ or } \widehat{E_U q_\alpha}(x) \right\}$$

Crude Monte Carlo expl :

$$\hat{f}(x) = \widehat{E_U}(f(x,U)) = \frac{1}{\mathrm{MC}} \sum_{i=1}^{\mathrm{MC}} f(x,u^i) \text{ , } u^i \text{ i.i.d. } \sim U$$

$$\hat{f}(x) = \widehat{E_U q_\alpha}(x) = \lfloor MC \times \alpha \rfloor\text{-th lowest among } f(x,u^1),\ldots,f(x,u^{\mathrm{MC}})$$

$\hat{g}(x) = \ldots$ (cf. reliability estimation classes)

Direct approaches (2/4)

Direct (naive) approaches to optimization with uncertainties have a **double loop** : propagate uncertainties on *U*, optimize on *x*.

OP times

Optimization : loop on *x*

MC times

Estimation of the performance (average, std dev, percentile of *f(x,U)* , prob. of g(x,U)) : loop on *u* , Monte Carlo

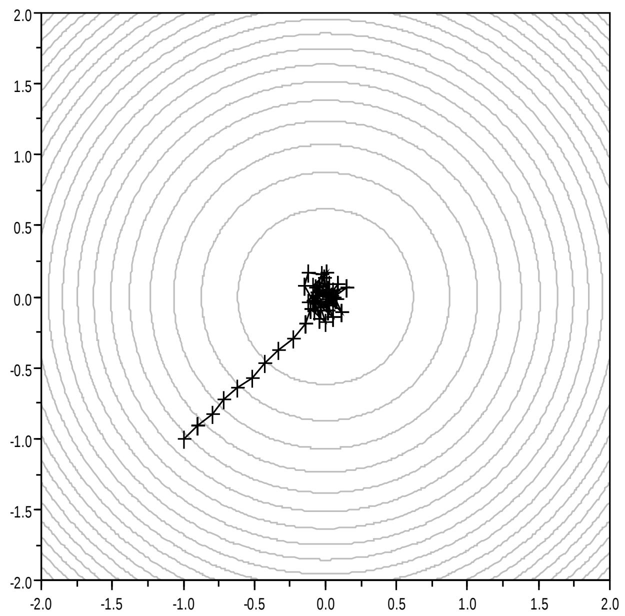
Such a double loop is **very costly** : Total cost = OP $\times$ MC (calls to s)

Direct approaches (3/4)

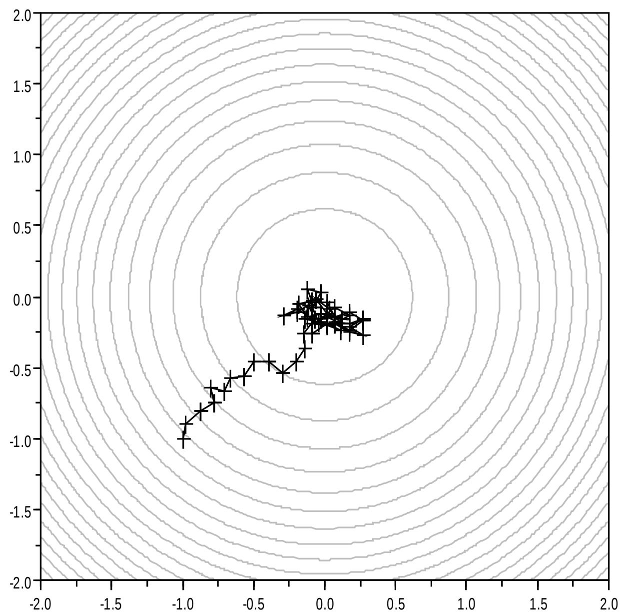
Most local (e.g., gradient based) optimizers will show poor convergence with noisy statistical estimators (e.g., crude Monte Carlo).

Ex : quasi-Newton method with finite differences

little noise



more noise



$$\hat{f}(x) = \frac{1}{100} \sum_{i=1}^{100} \left\| x + u_i \right\|^2 \qquad u_i \sim N(0, I_2) \qquad \hat{f}(x) = \left\| x + u_i \right\|^2$$

Direct approaches (4/4)

Avoid noisy statistical estimators with **common random numbers**

Expl : $\hat{f}(x) = \frac{1}{\text{MC}} \sum_{i=1}^{\text{MC}} f(x, u^i)$ has same regularity as f for given u^i's

see Bruno Tuffin's class on sampling (quasi Monte Carlo)

once for all before the loops

Sample $\{u^1, \ldots, u^{\text{MC}}\}$ according to U

for i=1:OP
    optimizer(past $x, \hat{f}(x)$) → new x
    for j=1:MC
        $\hat{f}(x) \leftarrow \hat{f}(x) + f(x, u^i)$
    end
    $\hat{f}(x) \leftarrow \hat{f}(x)/\text{MC}$
end

But,

- does not solve the cost issue
- (less critically) the estimates $\hat{f}(x^1), \ldots, \hat{f}(x^{\text{OP}})$ depend on the choice of $u^1, \ldots, u^{\text{MC}}$

Outline of the talk

1. Motivations for robust optimization

2. Formulations of optimization problems with uncertainties

3. Kriging-based approaches (costly functions)

4. Evolutionary approaches (non costly functions)

Unconstrained continuous optimization

$$\min_{x \in S \subset \mathbb{R}^d} f(x \text{ with uncertainties})$$

general,
no control on U

$$\min_{x \in S \subset \mathbb{R}^d} \hat{f}(x)$$

specific to E(f),
control on U

$$\min_{x \in S \subset \mathbb{R}^d} E_U f(x, U)$$

[Constraints, $g(x) \leq 0$, are not explicitely discussed in this talk. As a patch, you may assume that

$$\begin{array}{c} \min_{x \in S \subset \mathbb{R}^d} f(x) \\ g(x) \leq 0 \end{array} \rightarrow \min_{x \in S \subset \mathbb{R}^d} f(x) + p \times \max^2(0, g(x))$$

Constraints satisfaction problem : A. Chaudhuri, R. Le Riche and M. Meunier, *Estimating feasibility using multiple surrogates and ROC curves*, 54th AIAA SDM Conference, Boston, USA, 8-11 April 2013.]

Unconstrained continuous optimization **of costly functions**
~ 20 to 1000 calls possible, d = 1 to 20

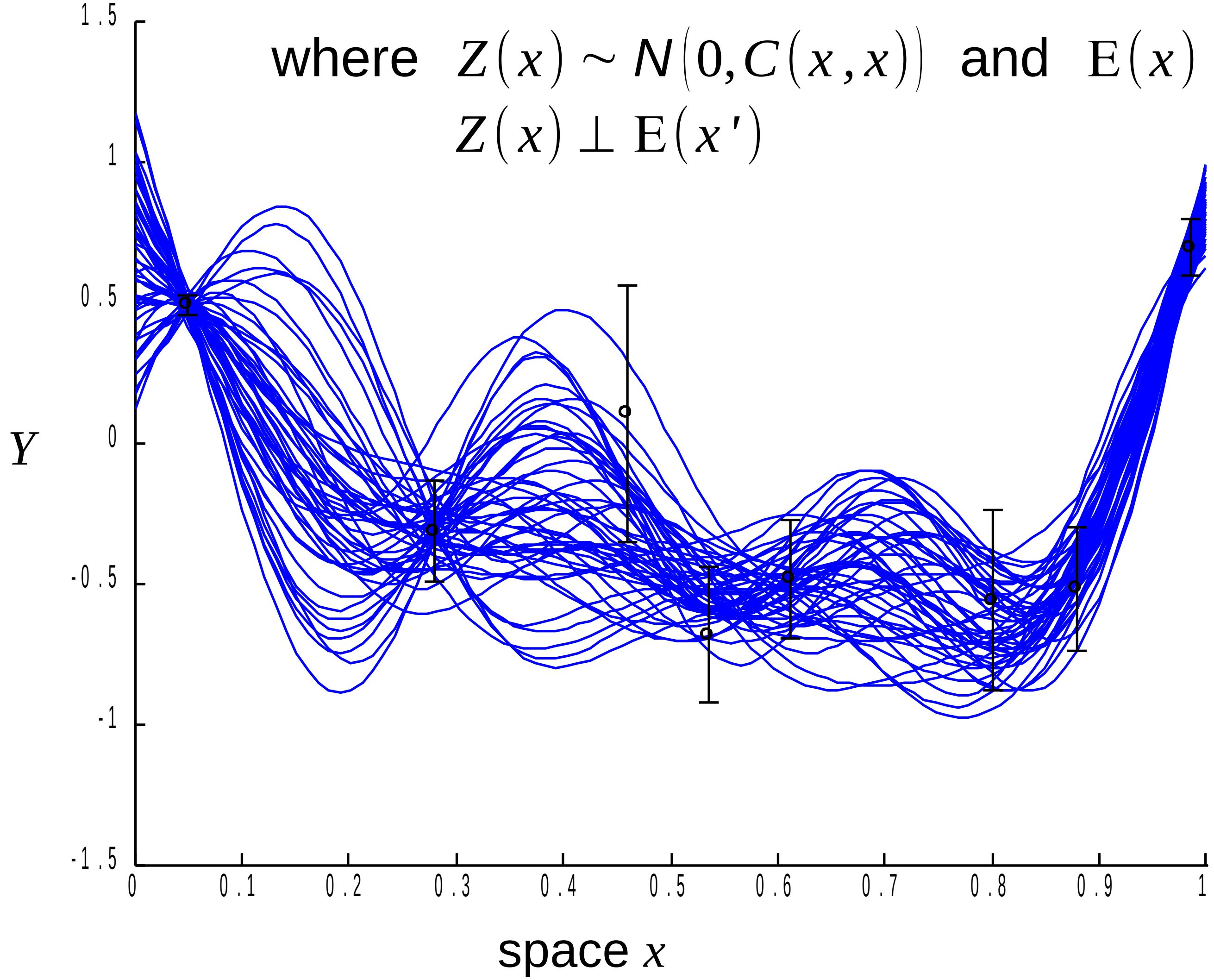
Kriging with noisy observations (1/5)

$$Y(x) = \mu(x) + Z(x) + \mathrm{E}(x)$$

$\hat{f}(x)$ here

where $Z(x) \sim N(0, C(x,x))$ and $\mathrm{E}(x) \sim N(0, \tau_x^2)$

$Z(x) \perp \mathrm{E}(x')$



Kriging with noisy observations (2/5)

Apply the conditioning result to the vector

$$\begin{bmatrix} Z(x^*)+\mu(x^*) \\ Y(x^1) \\ \cdots \\ Y(x^n) \end{bmatrix} = \begin{bmatrix} Z(x^*)+\mu(x^*) \\ \boldsymbol{\mu}+\mathbf{Z}+\mathbf{E} \end{bmatrix} \sim N\left(\begin{bmatrix} \mu(x^*) \\ \boldsymbol{\mu} \end{bmatrix}, \begin{bmatrix} \sigma^2 & C(x^*,\boldsymbol{X}) \\ C(x^*,\boldsymbol{X})^T & \boldsymbol{C}+\operatorname{diag}(\boldsymbol{\tau}^2) \end{bmatrix}\right)$$

because $\operatorname{Cov}(Y(x^i),Y(x^j)) = E\left((\mu^i+Z^i+\mathrm{E}^i-\mu^i)(\mu^j+Z^j+\mathrm{E}^j-\mu^j)\right) = C_{ij}+\delta_{ij}\tau_i^2$

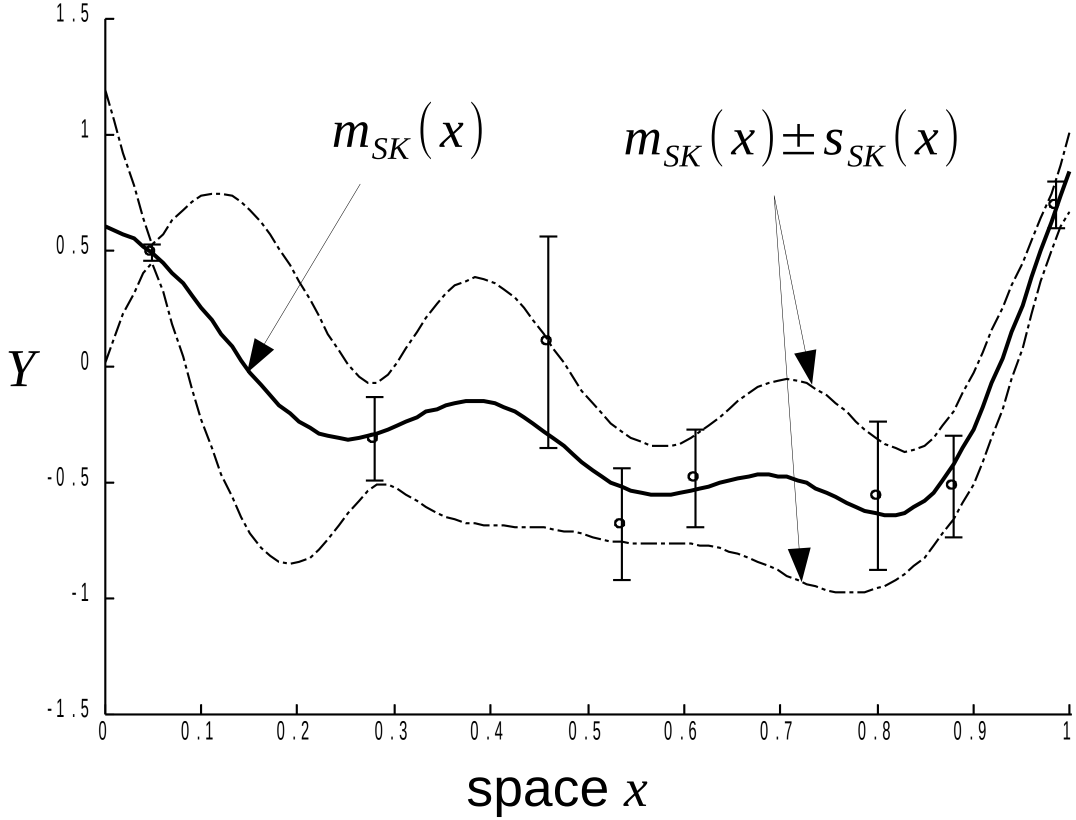
The only change w.r.t. the usual kriging formula is the addition of the observation variances on the covariance diagonal

$$(Z(x^*)+\mu(x^*))|\boldsymbol{Y}=\boldsymbol{y} \sim N\left(m_{SK}(x^*),\ v_{SK}(x^*)\right)$$

$$m_{SK}(x^*) = \mu(x^*)+C(x^*,\boldsymbol{X})(\boldsymbol{C}+\operatorname{diag}(\boldsymbol{\tau}^2))^{-1}(\boldsymbol{y}-\boldsymbol{\mu})$$

$$v_{SK}(x^*) = \sigma^2-C(x^*,\boldsymbol{X})(\boldsymbol{C}+\operatorname{diag}(\boldsymbol{\tau}^2))^{-1}C(x^*,\boldsymbol{X})^T$$

Kriging with noisy observations (3/5)



- **kriging no longer interpolating**
- **the kriging mean filters the noise**
- additive covariance diagonal terms called « nugget effect »
- often used as a regularization technique in non noisy situations

- kriging = our approach to link *x* and *U* spaces in optimization

Kriging with noisy observations (4/5)

In the context of robust optimization with MC estimators, the observation noise can be set as

$$\tau_i^2 = \text{variance of performance estimate, } \hat{f}(.) \text{ , at } x^i$$

Expl.: mean estimate has variance

$$\tau_i^2 = V\left(\hat{f}\left(x^i\right)\right) = \frac{1}{\text{MC}\left(\text{MC}-1\right)} \sum_{j=1}^{\text{MC}} \left(f\left(x, u^j\right) - \hat{f}\left(x\right)\right)^2$$

(Expl. with quantile estimate, cf. Le Riche et al., *Gears design with shape uncertainties using Monte Carlo simulations and kriging*, SDM, AIAA-2009-2257)

Kriging with noisy observations (5/5)

The hyperparameters can be tuned through max likelihood, 2 cases

Known (from context) **heterogeneous noise**

$$\boldsymbol{C}_{\boldsymbol{\tau}} \equiv \boldsymbol{C} + \mathrm{diag}\left(\tau_1^2, \dots, \tau_n^2\right) \quad , \quad \boldsymbol{C}_{\boldsymbol{\tau}} \equiv \sigma^2 \boldsymbol{R}_{\boldsymbol{\tau}}$$

and do the usual MLE estimation (cf. ``intro. to kriging class'')
replacing $\boldsymbol{R}$ by $\boldsymbol{R}_{\boldsymbol{\tau}}$ → $\hat{\sigma}$, $\hat{\theta}_1$, … , $\hat{\theta}_n$

Unknown homogeneous noise

$$\boldsymbol{C}_{\boldsymbol{\tau}} \equiv \boldsymbol{C} + \tau^2 \boldsymbol{I} \quad , \quad \boldsymbol{C}_{\boldsymbol{\tau}} \equiv \sigma^2 \boldsymbol{R}_{\boldsymbol{\tau}}$$

and do the usual MLE estimation replacing $\boldsymbol{R}$ by $\boldsymbol{R}_{\boldsymbol{\tau}}$
(1 additional parameter in the concentrated likelihood, τ)
→ $\hat{\tau}$, $\hat{\sigma}$, $\hat{\theta}_1$, … , $\hat{\theta}_n$

kriging-based approaches

We have seen

- how to formulate a robust optimization problem
- how to model noisy observations with kriging

- but how to optimize when a kriging metamodel is built ?

Kriging prediction minimization

The simplest (naive) approach.

For $t=1,t^{max}$ do,

Learn $Y^t(x)$ (m_K and s_K^2) from $f(x^1), \ldots , f(x^t)$

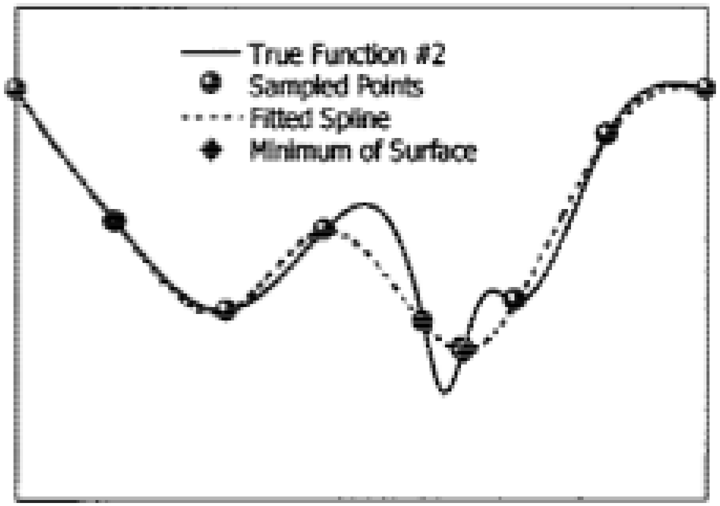
$x^{t+1} = \min_x m_K(x)$

Calculate $f(x^{t+1})$

$t = t+1$

End For

e.g., using CMA-ES*
if multimodal
* Hansen et al., 2003



But it may fail if $m_K(x^{t+1}) = f(x^{t+1})$:
the minimizer of m_K is at a data point
which may not even be a local optimum.

D. Jones, A taxonomy of global optimization methods based on response surfaces, JOGO, 2001.

Notation : this slide + the ones coming about EGO are general to any optimization, therefore $\hat{f} \rightarrow f$

Kriging and optimization

Search the volume in an efficient manner.

Balance **EXPLORATION** with **INTENSIFICATION**

- **We will deterministically fill the design space in an efficient order.**

- **Other global search principles**
  - **Stochastic searches** : (pseudo)-randomly sample the design space S, use probabilities to intensify search in known high performance regions and sometimes explore unknown regions.
  - (pseudo-)**Randomly restart** local searches.
  - (and mix the above principles)

A state-of-the-art global optimization algorithm using metamodels : EGO

(D.R. Jones et al., JOGO, 1998)

EGO = Efficient Global Optimization = use a « kriging » metamodel to define the Expected Improvement (EI) criterion. Maximize EI to creates new x's to simulate.

EGO deterministically creates a series of design points that ultimately would fill S.

Some opensource implementations :

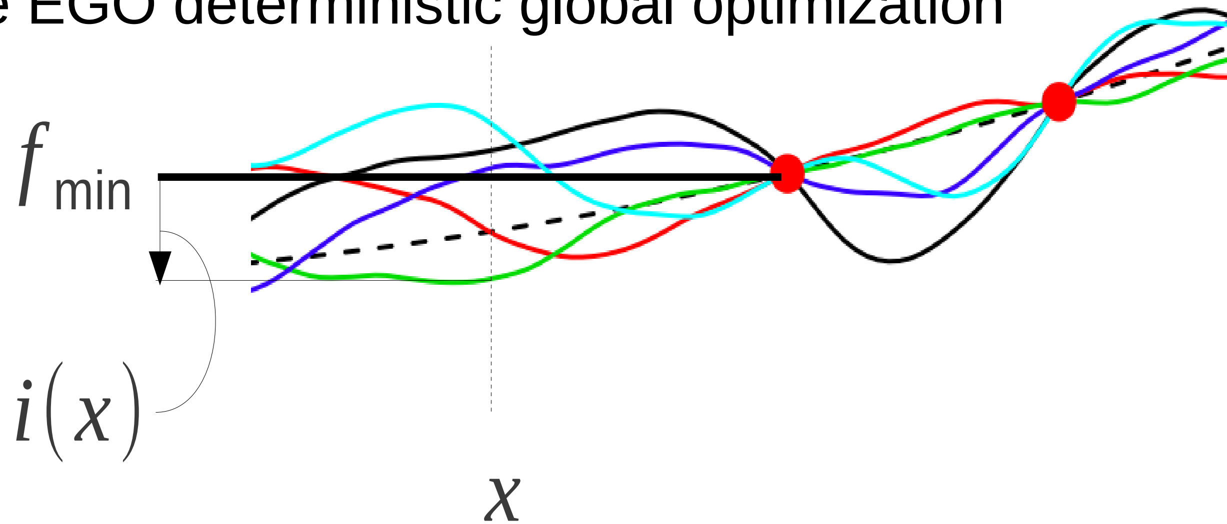
- DiceOptim in R (EMSE & Bern Univ.)
- Krisp in Scilab (Riga Techn. Univ & EMSE)
- STK: a Small (Matlab/GNU Octave) Toolbox for Kriging, (Supelec)

(one point-) Expected improvement

A natural measure of progress : the improvement,

$$I(x) = \left[f_{\min} - F(x)\right]^{+} \mid F(\mathbf{x}) = f(\mathbf{x}) \quad , \quad \text{where } [.]^{+} \equiv max(0,.)$$

- The expected improvement is known analytically.
- It is a parameter free measure of the exploration-intensification compromise.
- Its maximization defines the EGO deterministic global optimization algorithm.

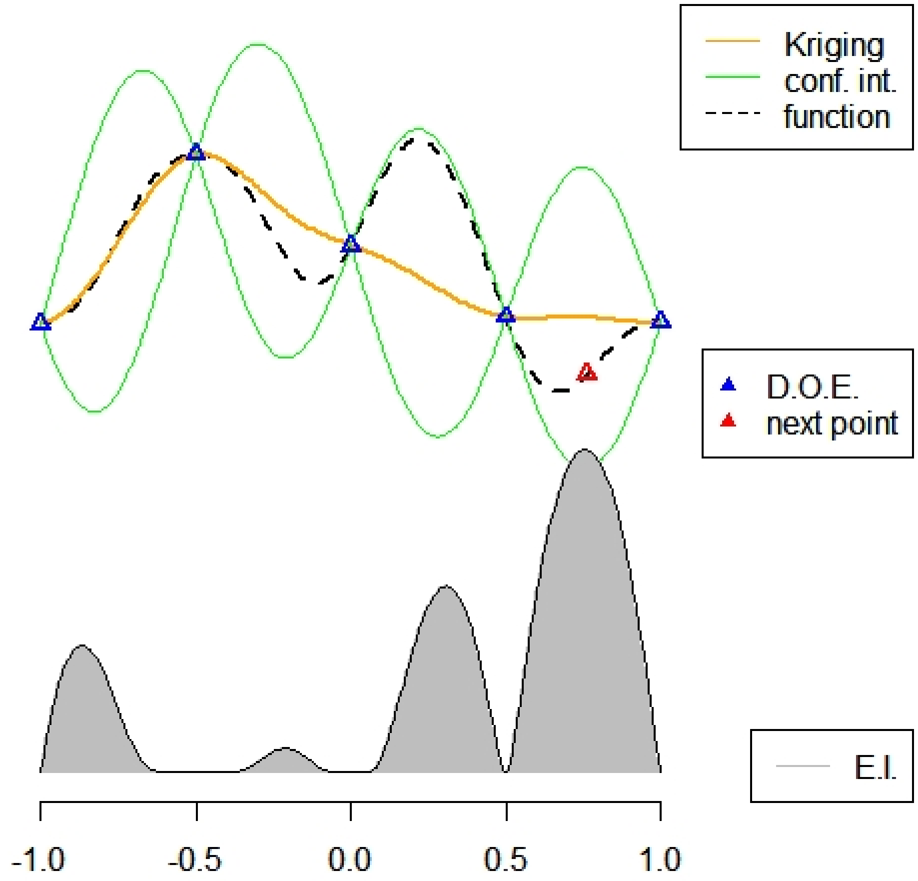


$$EI(x) = \sqrt{v_K(x)} \times \left(u(x)\Phi(u(x)) + \varphi(u(x)) \right) \quad , \quad \text{where } u(x) = \frac{f_{\min} - m_K(x)}{\sqrt{v_K(x)}}$$

One EGO iteration

At each iteration, EGO adds to the t known points the one that maximizes EI,

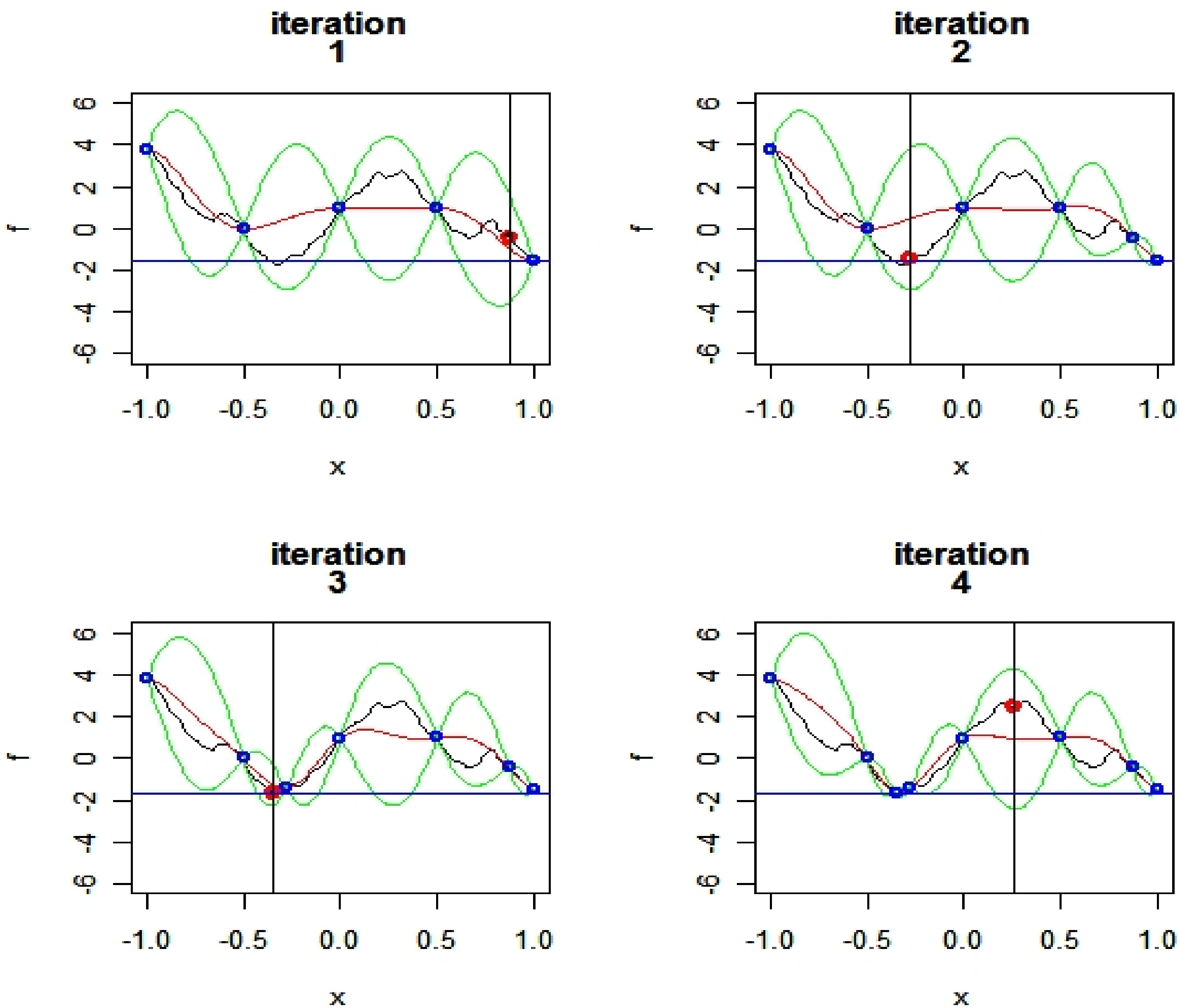
$$x^{t+1} = arg\,max_x\, EI(x)$$



then, the kriging model is updated ...

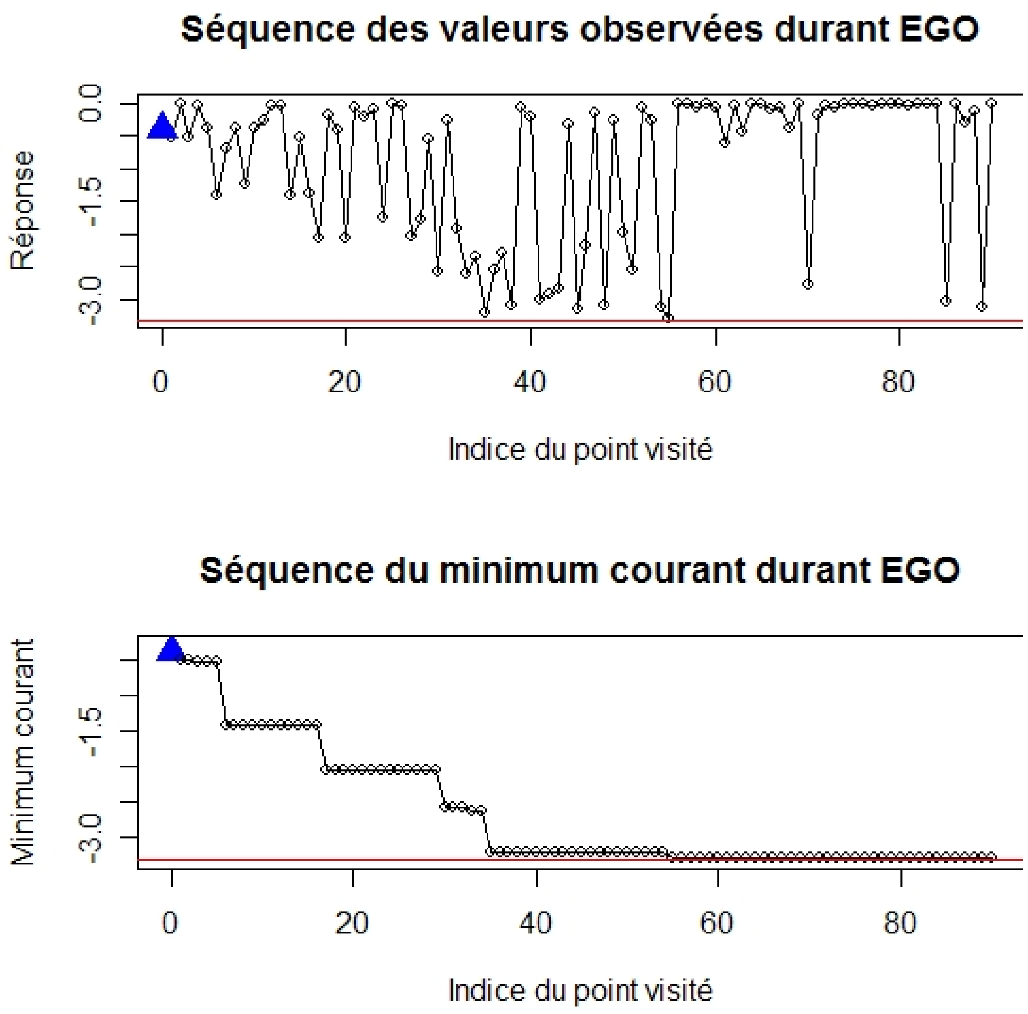
kriging-based approaches

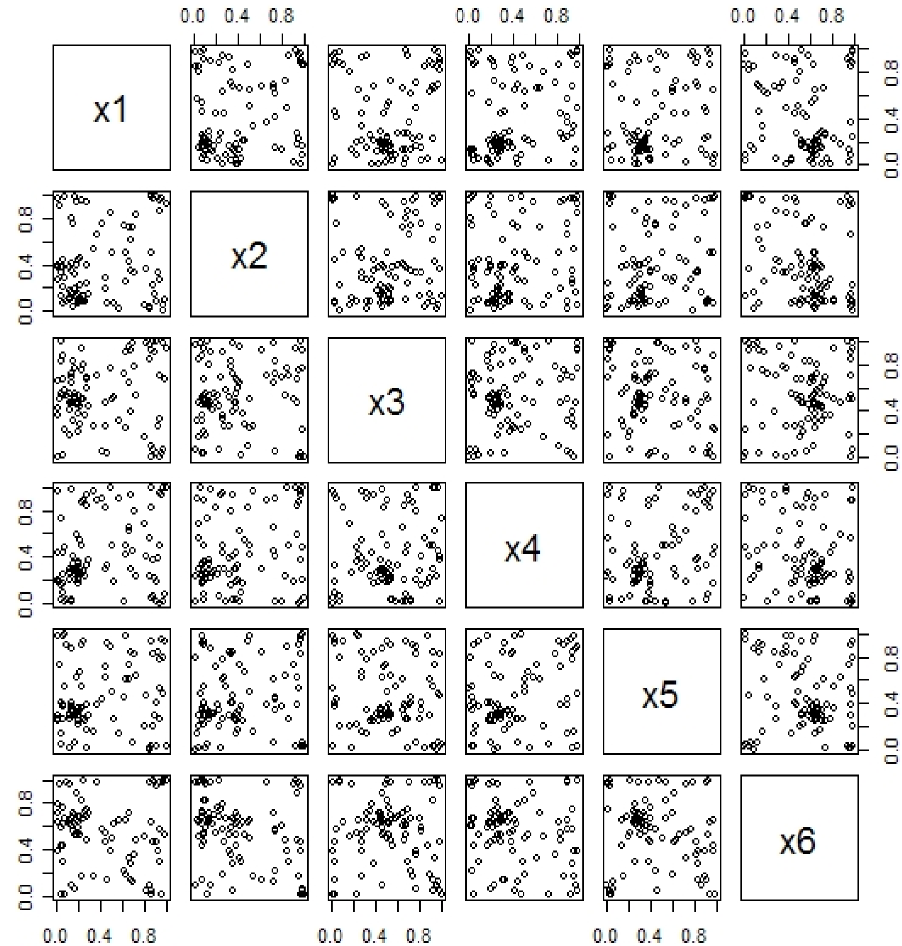
EGO : example



kriging-based approaches
EGO : 6D example

Hartman function, *f(x*)=-3.32* , 10 points in initial DoE





(DiceOptim, D. Ginsbourger, 2009)

Outline of the talk

1. Motivations for robust optimization

2. Formulations of optimization problems with uncertainties

3. Kriging-based approaches to robust optimization (costly functions)
   - Kriging noisy observations
   - Optimization and kriging
   - Robust optimization, no control on U
   - Robust optimization, control on U

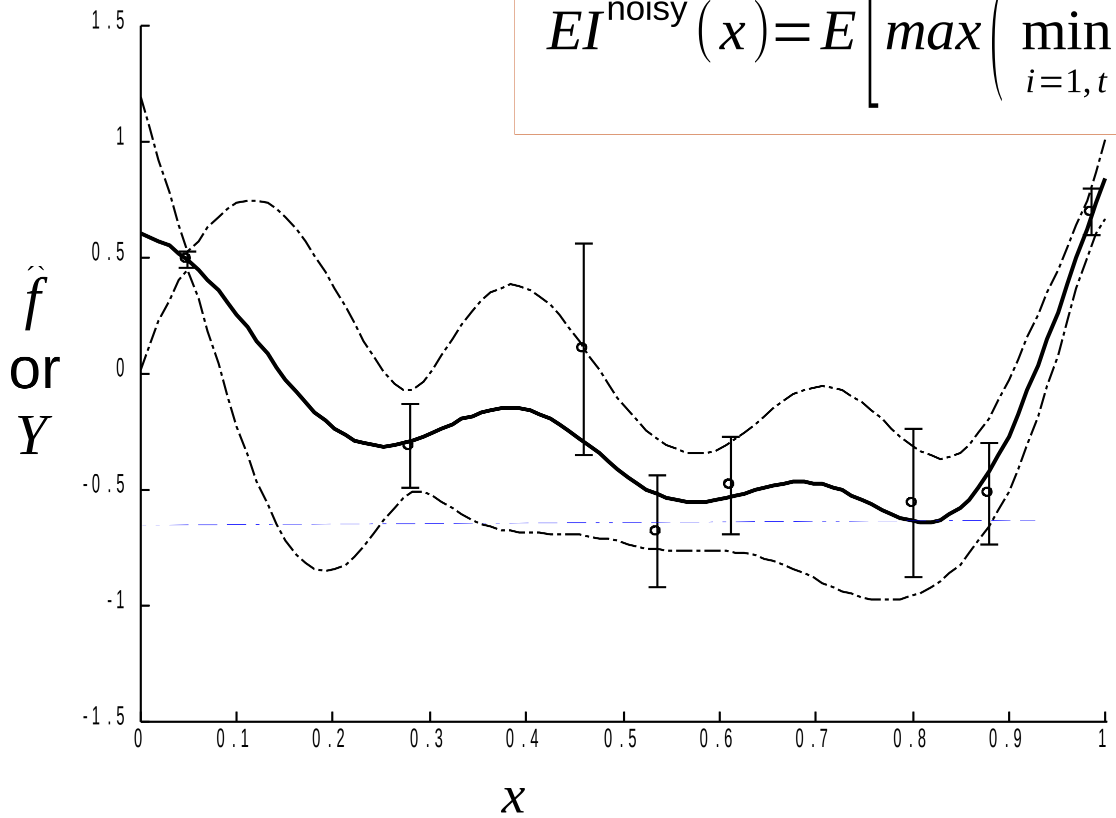
4. Evolutionary approaches (non costly functions)

EI for noisy functions

EI should not be used for noisy observations because $\hat{f}_{min} \equiv y_{min}$ is noisy ! (a low y_{min} would mislead EGO for a long time

Solution 1 : Add nugget effect and replace y_{min} by the best observed mean (filters out noise in already sampled regions) :

$$EI^{\text{noisy}}(x) = E\left[max\left(\min_{i=1,t} m_K(x^i) - (Z(x)+\mu(x)) \text{ , } 0 \right) \right]$$



known analytically
replace f_{min} by $\min_{i=1,t} m_K(x^i)$ in EI formula

Kriging-based robust optimization, no control on U

Expected Quantile Improvement

V. Picheny, D. Ginsbourger, Y. Richet, *Optimization of noisy computer experiments with tunable precision*, Technometrics, 2011.

Solution 2 : Add nugget effect and use the expected quantile improvement.

$$EQI(x) = E\left[max\left(q_{min} - Q^{t+1}(x)\,,\,0\right)\right]$$

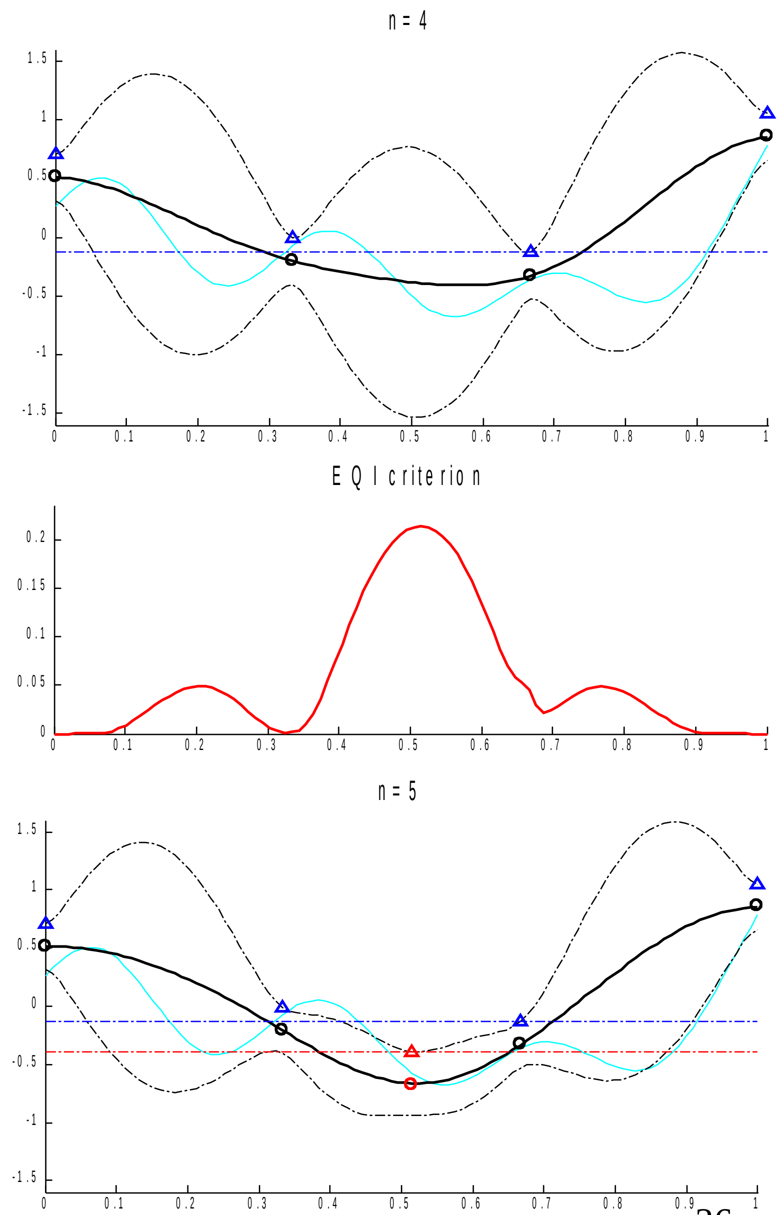
$$q_{min} = \min_{i=1,t}\; m_K(x^i) + \alpha\, s_K(x^i)$$

$$Q^{t+1}(x) = M_K^{t+1}(x) + \alpha\, s_K^{t+1}(x)$$

$M_K^{t+1}(x)$ is a linear function of $Y(x)$

$\Rightarrow EQI(x)$ is known analytically

A conservative criterion (noise and spatial uncertainties are seen as risk rather than opportunities).
Better for assigning ressources to reduce noise on a given DoE X^t obtained by Solution 1.



Outline of the talk

1. Motivations for robust optimization

2. Formulations of optimization problems with uncertainties

3. Kriging-based approaches to robust optimization (costly functions)
   - Kriging noisy observations
   - Optimization and kriging
   - Robust optimization, no control on U
   - Robust optimization, control on U

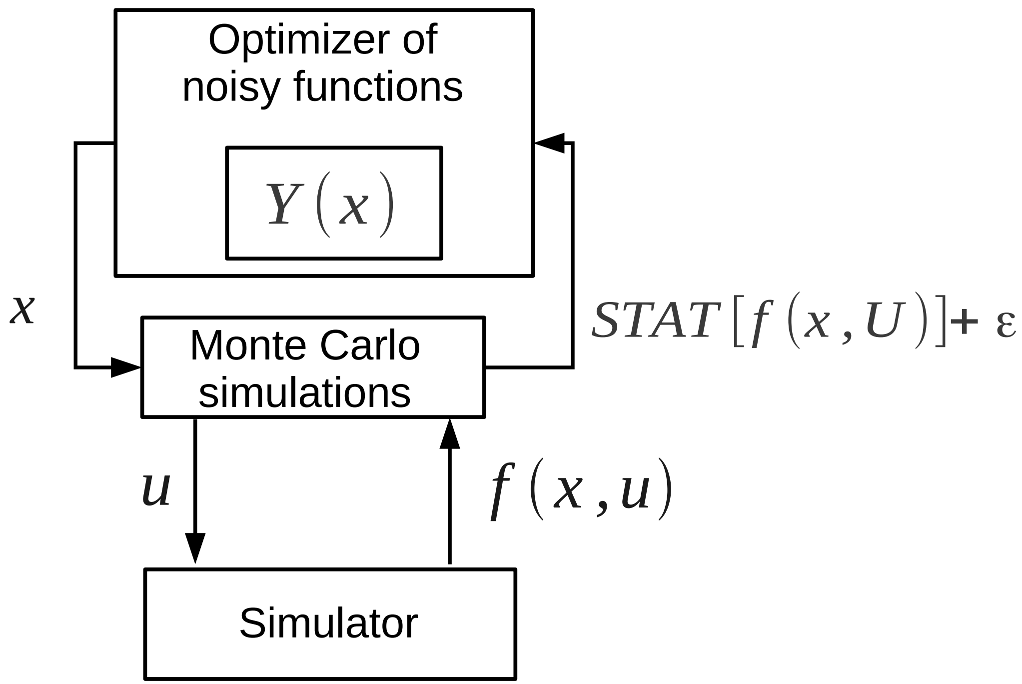
4. Evolutionary approaches (non costly functions)

(x,u) surrogate based approach

Assumptions : *x* and *U* controlled

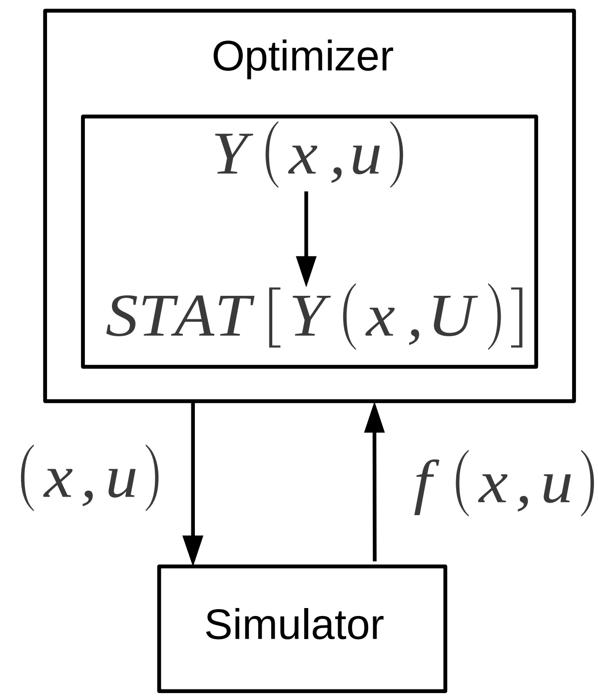
Y : surrogate model

Direct approach



Multiplicative cost of two loops involving *f*

(x,u) surrogate based approach



Only one loop of *f*

A general Monte Carlo - kriging algorithm

Hereafter is an example of a typical surrogate-based (here kriging) algorithm for optimizing any statistical measure of $f(x,u)$ (here the average).

MC – kriging algorithm

Create initial DOE (X^t,U^t) and evaluate f there ;
While stopping criterion is not met:

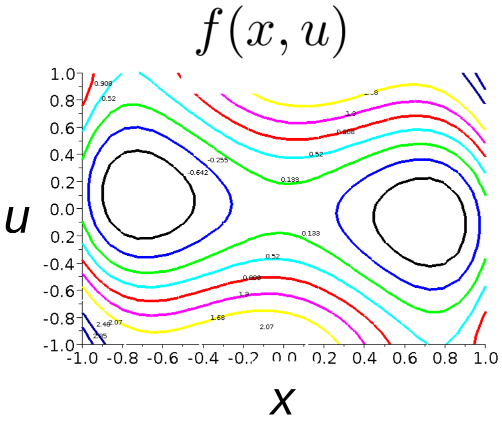
- Create kriging approximation Y^t in the joint (x,u) space from $f(X^t,U^t)$
- Estimate the value of the statistical objective function from Monte Carlo simulations on the kriging average m_Y^t.

  Expl : $\hat{f}(x^i) = \frac{1}{s}\sum_{k=1}^{s} m_K^t(x^i,u^k)$, where u^k i.i.d. from pdf of U
- Create kriging approximation Z^t in x space from $\left(x^i \, , \, \hat{f}(x^i)\right)_{i=1,t}$
- Maximize $EI_Z^{noisy}(x)$ to obtain the next simulation point $\rightarrow x^{t+1}$
  u^{t+1} sampled from pdf of U

  only call to f !
- Calculate simulator response at the next point, $f(x^{t+1},u^{t+1})$.
  Update DOE and t

Simultaneous optimization and sampling

Objective : $$\min_x \mathbb{E}_U[f(x,U)]$$



Principle : work in the joint (x,u) space.

Cf. J. Janusevskis and R. Le Riche, *Simultaneous kriging-based estimation and optimization of mean response*, Journal of Global Optimization, Springer, 2012

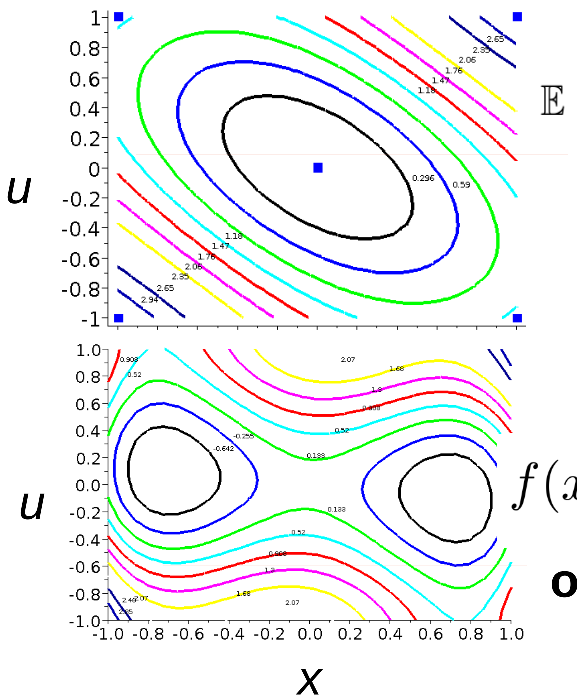
Integrated kriging (1)

$\min_x \mathbb{E}_U[f(x,U)]$ **: objective**

$Y^t_{(x,u)}(\omega)$ **: kriging approximation to deterministic** $f(x,u)$

$\boxed{Z^t_{(x)}(\omega) = \mathbb{E}_U[Y^t_{(x,U)}(\omega)]}$ **: integrated process approximation to** $\mathbb{E}_U[f(x,U)]$

integrate



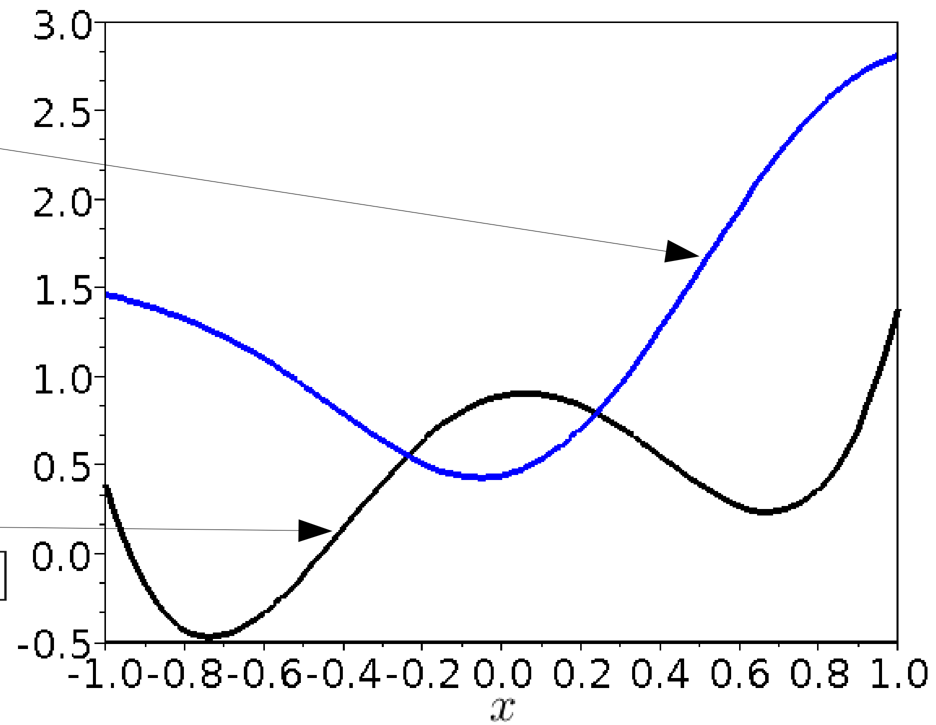
$\mathbb{E}\left(Y^t_{(x,u)}(\omega)\right)$

approximation $E_\Omega[Z_{(x)}(\omega)]$

f

$f(x,u)$

objective $E_U[f(x,U)]$



Integrated kriging (2)

The integrated process over *U* is defined as

$$\boxed{Z_{(x)}(\omega) = \mathbb{E}_U[Y^t_{(x,U)}(\omega)] = \int_{\mathbb{R}^m} Y^t_{(x,u)}(\omega) d_\mu(u)}$$

$d\mu(u)$ -probability measure on *U*

Because it is a linear transformation of a Gaussian process, it is Gaussian, and fully described by its mean and covariance

$$m_Z(x) = \int_{\mathbb{R}^m} m_Y(x,u) d\mu(u)$$

$$cov_Z(x;x') = \int_{\mathbb{R}^m} \int_{\mathbb{R}^m} cov_Y(x,u;x'u') d\mu(u) d\mu(u')$$

Analytical expressions of m_z and cov_z for Gaussian *U*'s are given in

J. Janusevskis and R. Le Riche, *Simultaneous kriging-based estimation and optimization of mean response*, Journal of Global Optimization, Springer, 2012

EI on the integrated process (1)

Z is a process approximating the objective function $\mathbb{E}_U[f(x,U)]$

Optimize with an Expected Improvement criterion,

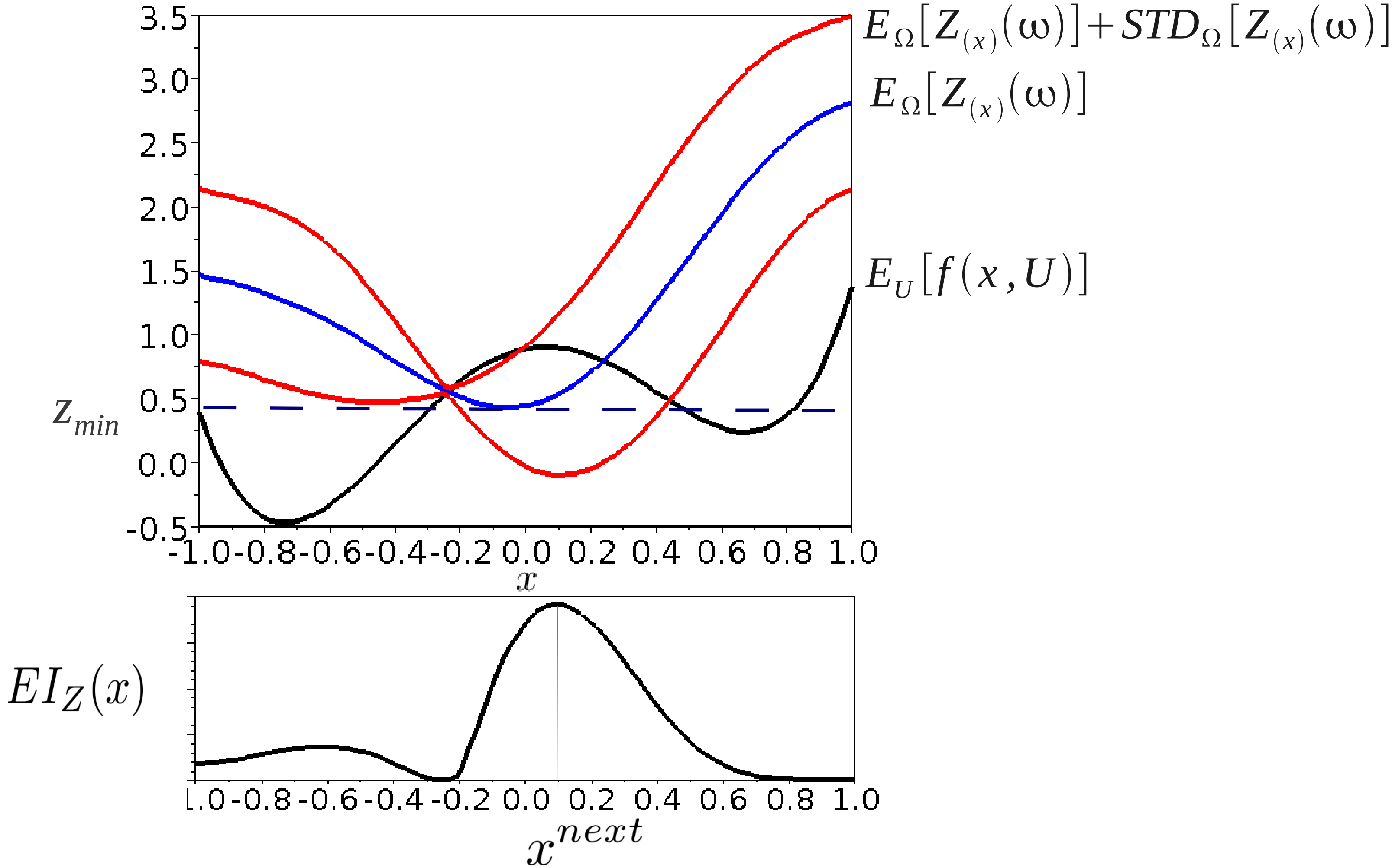
$$\boxed{x^{next} = \arg\max_x EI_Z(x)}$$

Optimize with an Expected Improvement criterion,

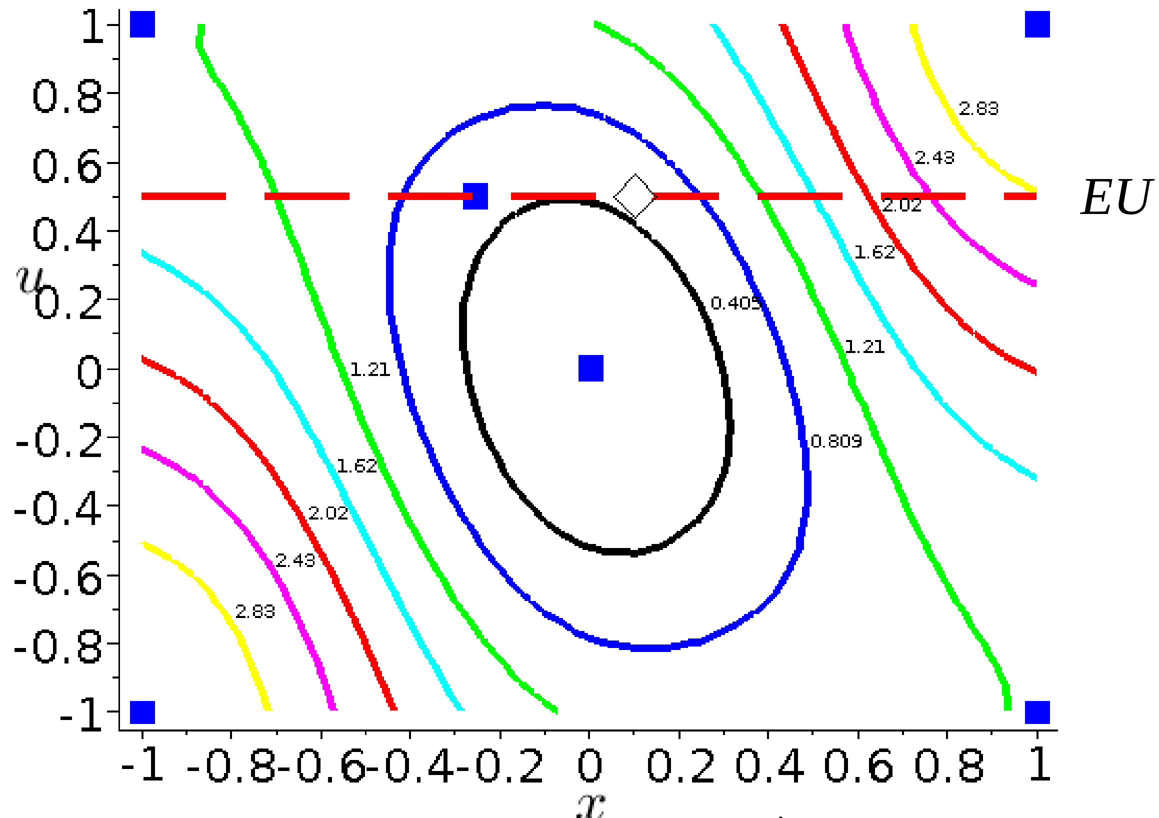
$I_Z(x) = max(z_{min} - Z(x), 0)$, but z_{min} not observed (in integrated space).

$\Rightarrow$ Define $z_{min} = \min_{x^{1,} \ldots, x^t} E(Z(x))$

EI on the integrated process (2)



EI on the integrated process (3)



$$x^{next} = \arg\max_{x} EI_Z^t(x)$$

x ok. What about *u* ? (which we need to call the simulator)

Simultaneous optimization and sampling : method

x^{next} gives a region of interest from an optimization of the expected f point of view.

One simulation will be run to improve our knowledge of this region of interest → one choice of (x,u).

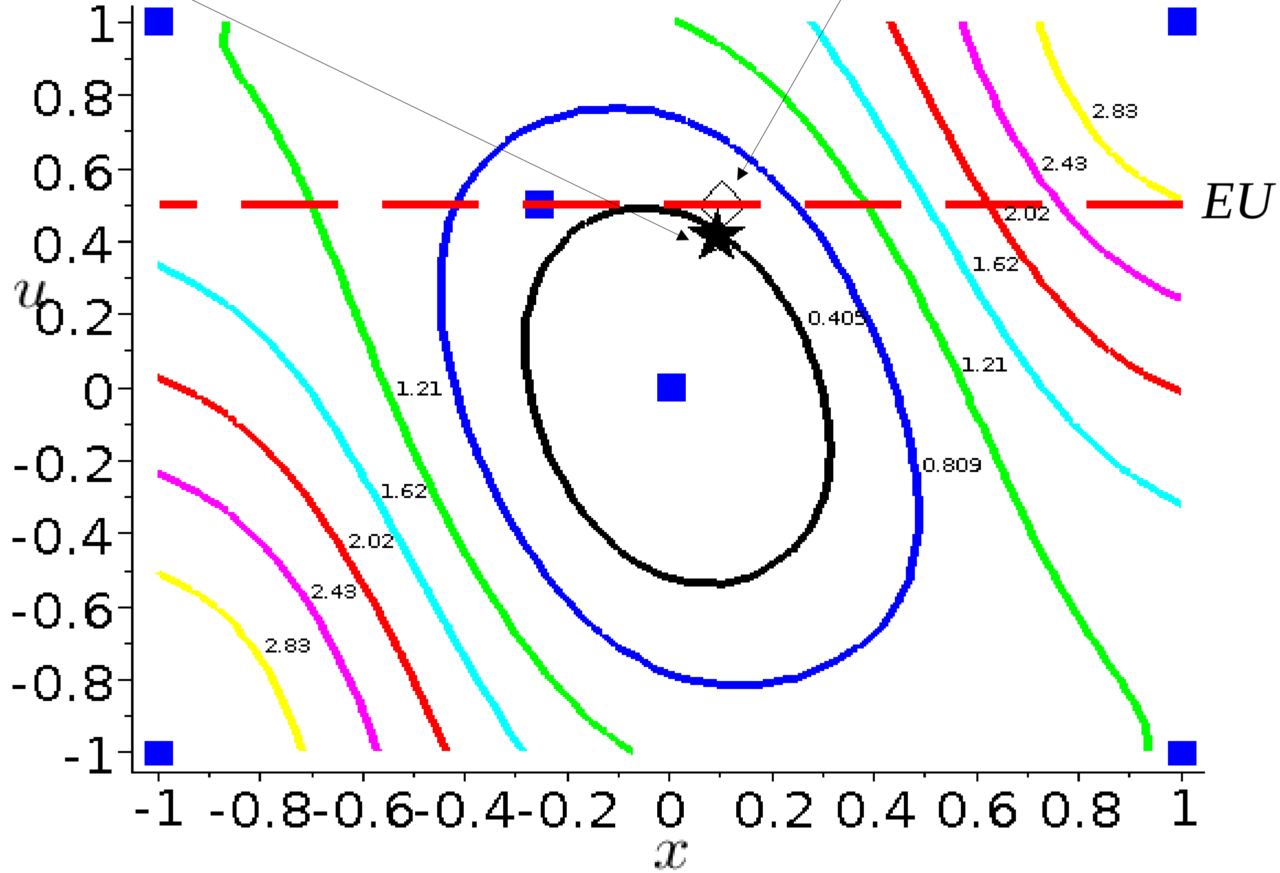
Choose (x^{t+1},u^{t+1}) that provides the most information, i.e., which minimizes the variance of the integrated process at x^{next}

$$\boxed{(x^{t+1}, u^{t+1}) = \arg\min_{x,u} \mathbb{VAR}\ Z^{t+1}_{(x^{next})}(\omega)}$$

Simultaneous optimization and sampling : expl.

$$x^{next} = \arg\max_{x} EI_Z^t(x)$$

$$(x^{t+1}, u^{t+1}) = \arg\min_{x,u} \mathbb{VAR}\ Z^{t+1}_{(x^{next})}(\omega)$$



Simultaneous optimization and sampling : algo

Create initial DOE in (x,u) space;

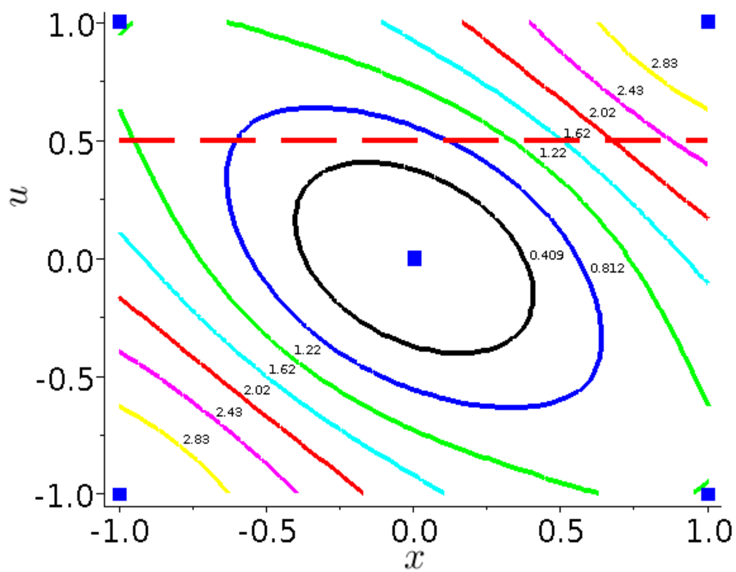
While stopping criterion is not met:

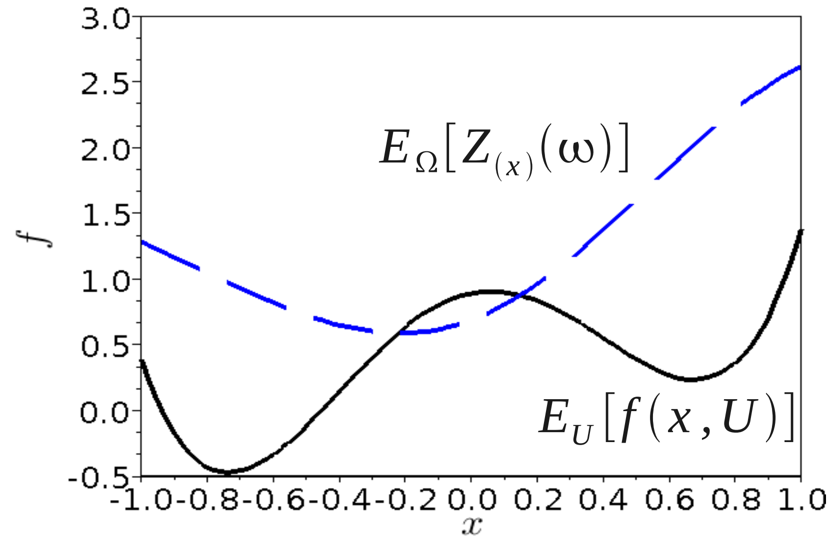
- **Create kriging approximation *Y* in the joint space** (x,u)
- **Using covariance information of *Y* to obtain approximation *Z* of the objective in the deterministic space** (x)
- **Use EI of Z to choose** (x^{next})
- **Minimize** $VAR(Z(x^{next}))$ **to obtain the next point** (x^{t+1},u^{t+1}) **for simulation**
- **Calculate simulator response at the next point** $f(x^{t+1},u^{t+1})$

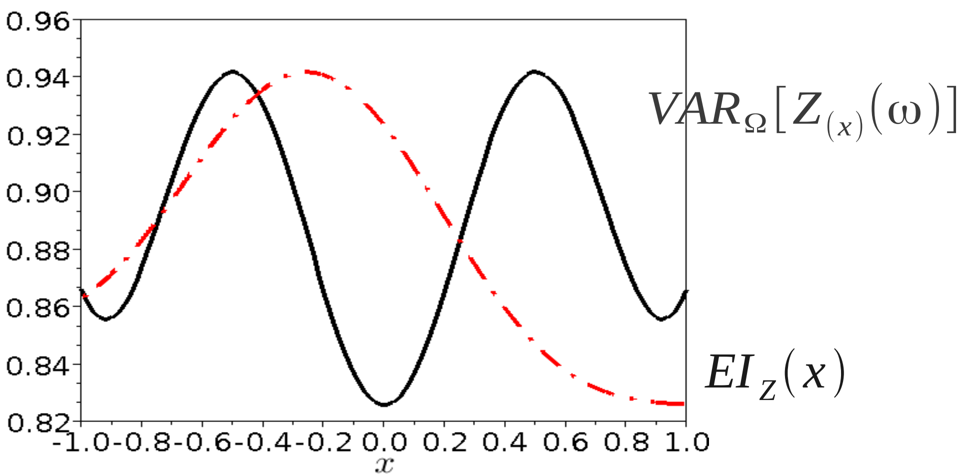
(4 sub-optimizations, solved with CMA-ES)

2D Expl, simultaneous optimization and sampling

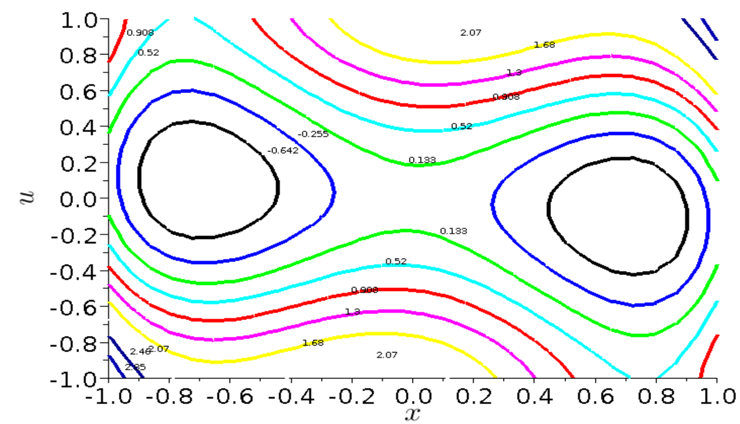
DOE and $E[Y(x,u)]$





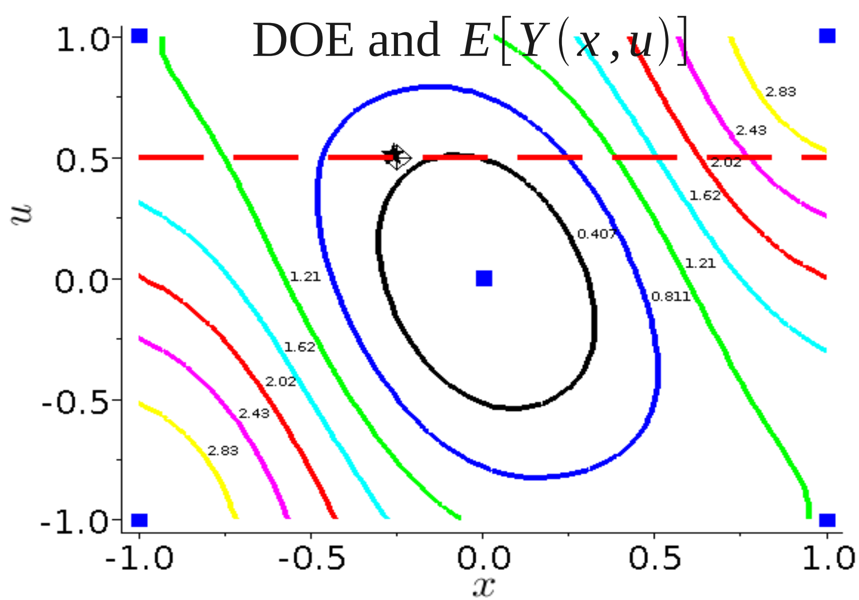


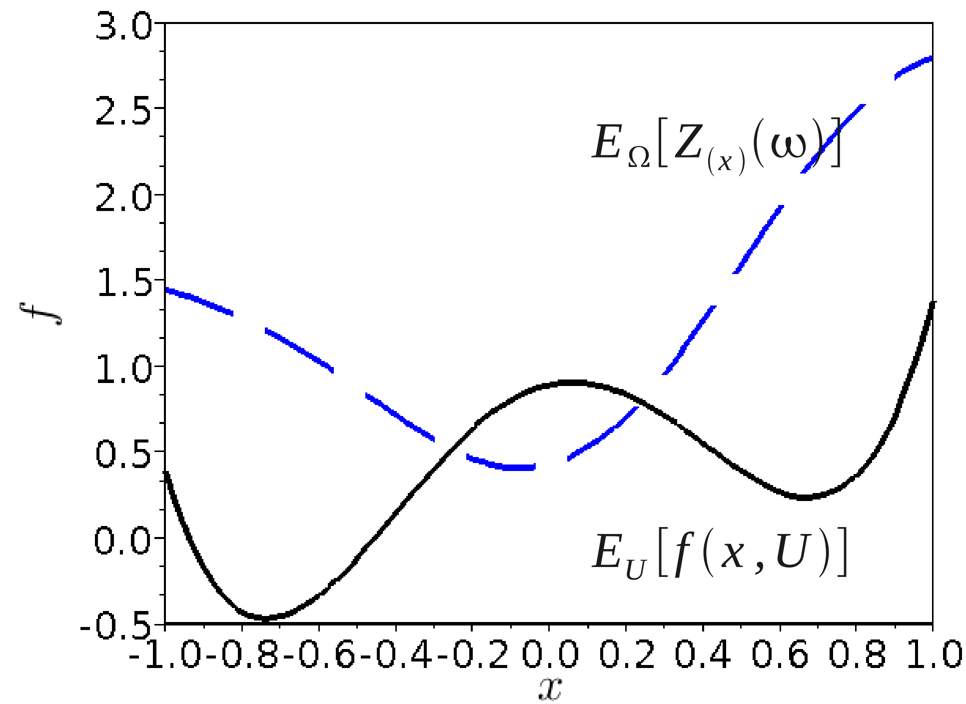
test function

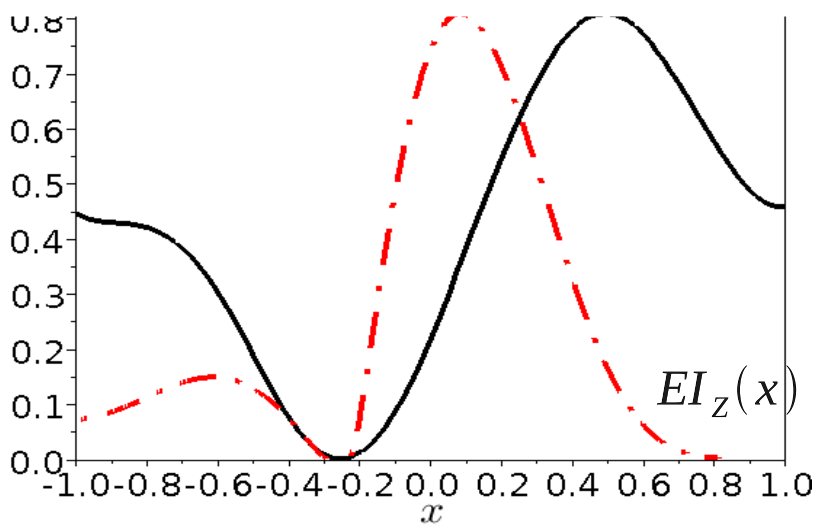


Kriging based optimization with uncertainties, U controlled

1st iteration







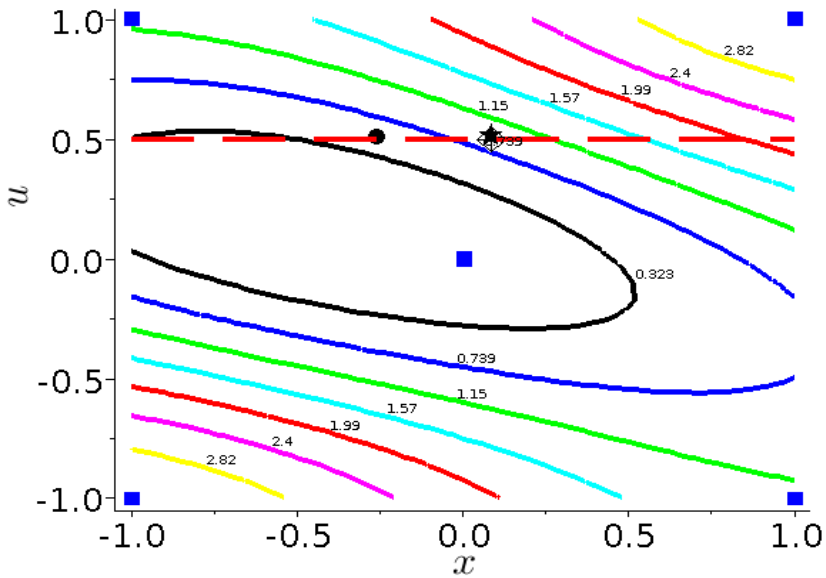
$VAR_\Omega[Z_{(x)}(\omega)]$

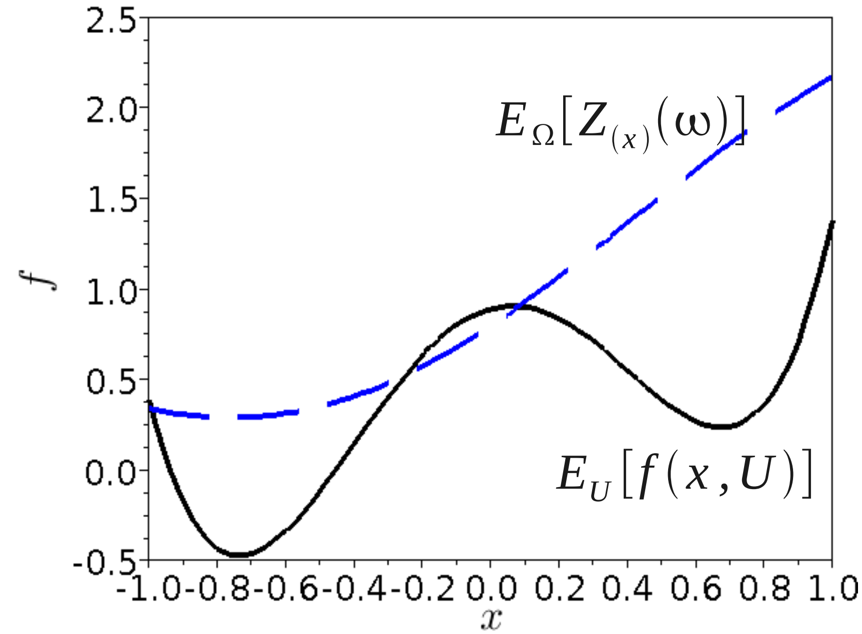
◇ $- \ (x^{next}, \mu)$

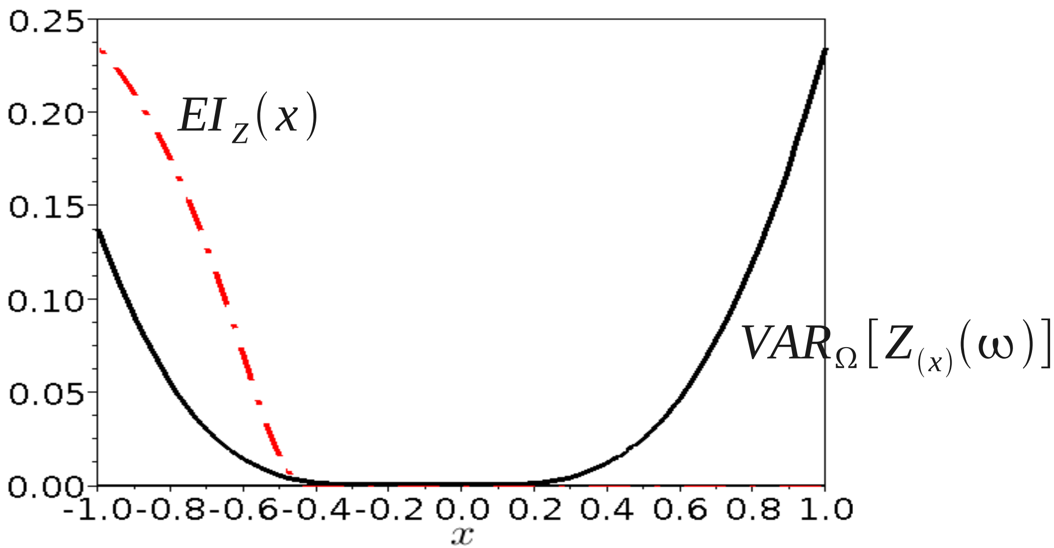
★ $- \ (x^{t+1}, u^{t+1})$

Kriging based optimization with uncertainties, U controlled
2nd iteration

DOE and $E[Y(x,u)]$





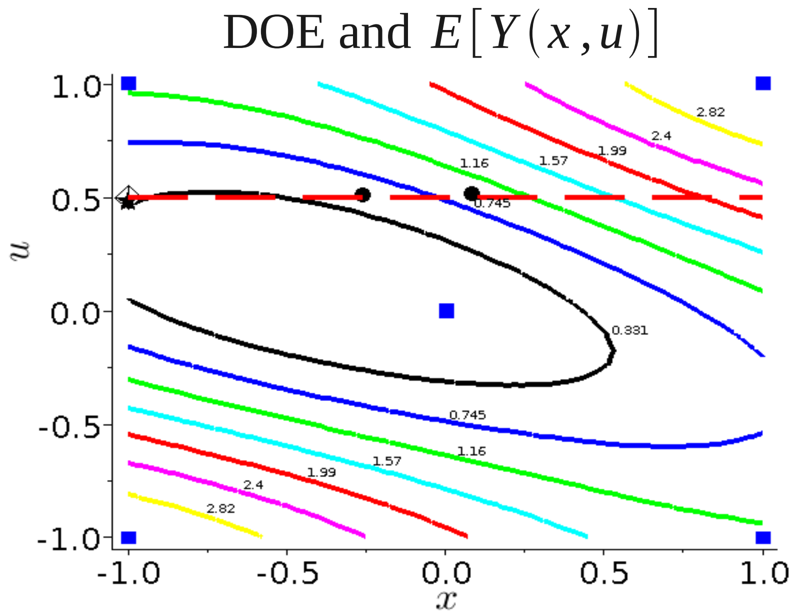


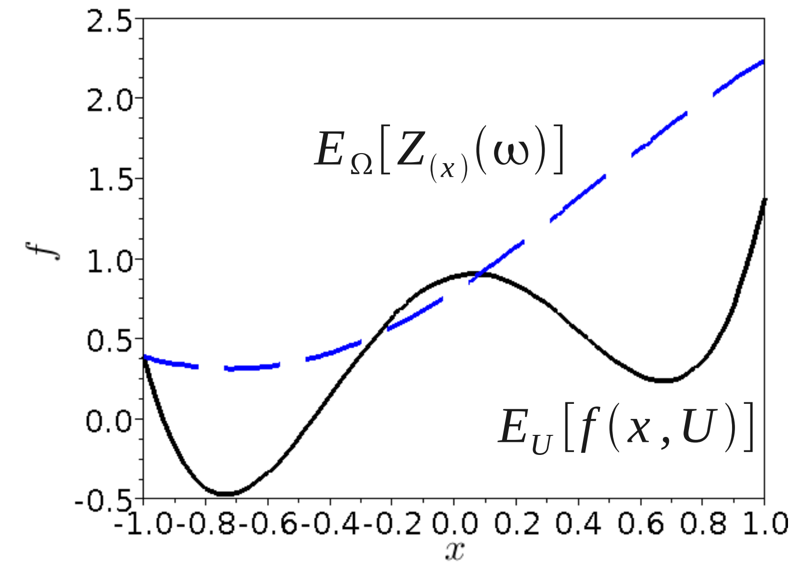
◇ – (x^{next}, μ)

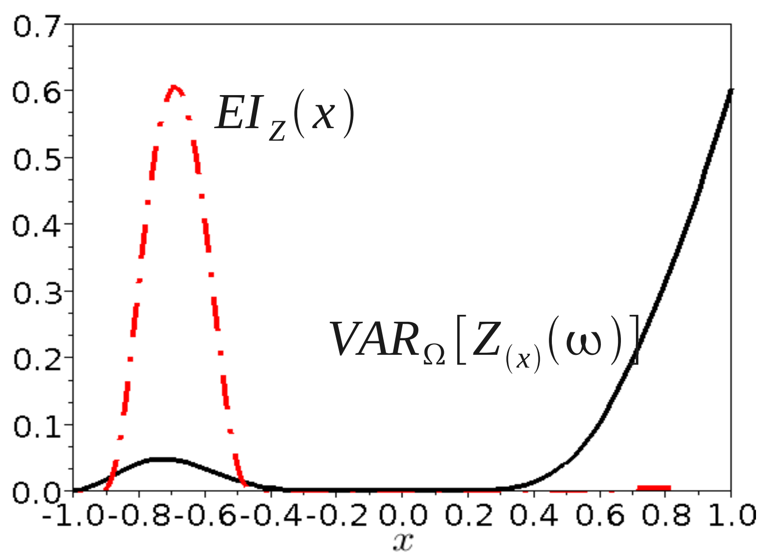
★ – (x^{t+1}, u^{t+1})

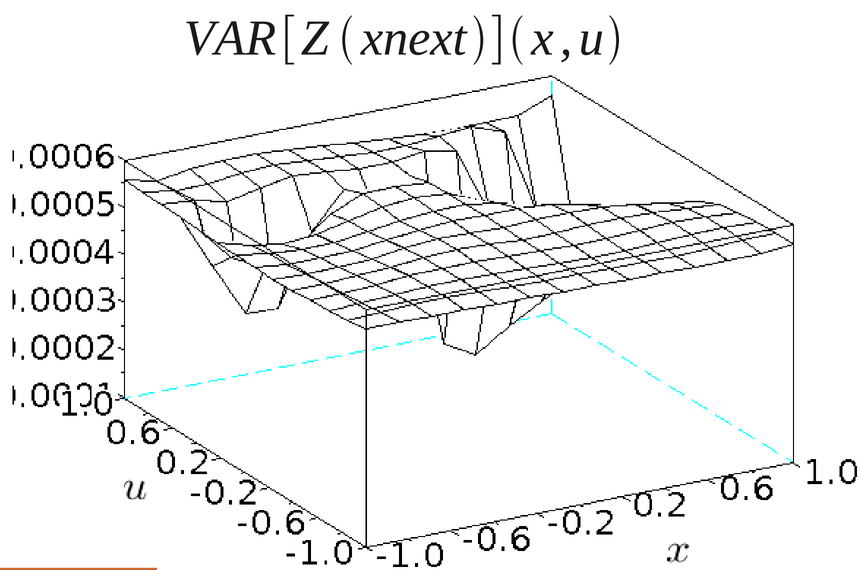
Kriging based optimization with uncertainties, U controlled

3rd iteration

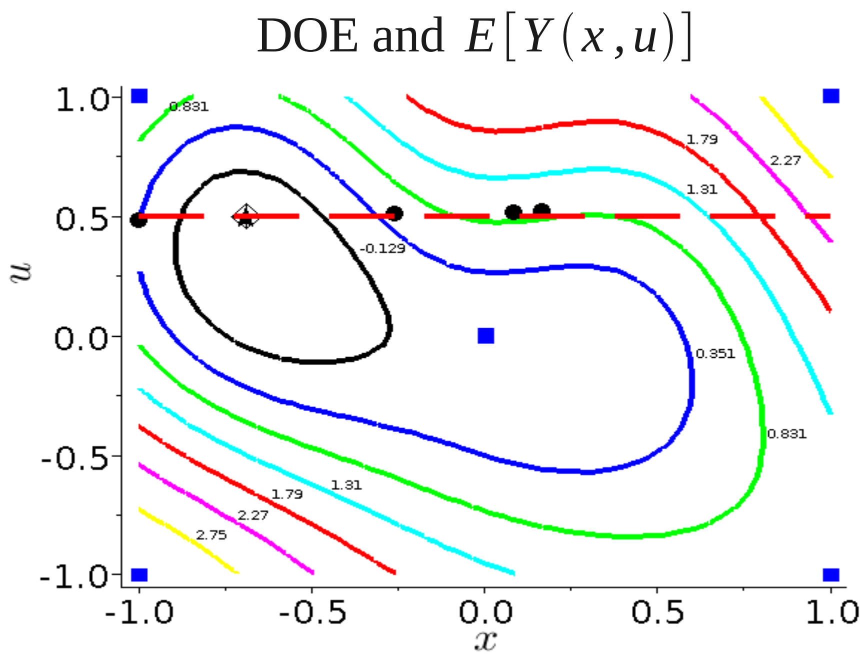


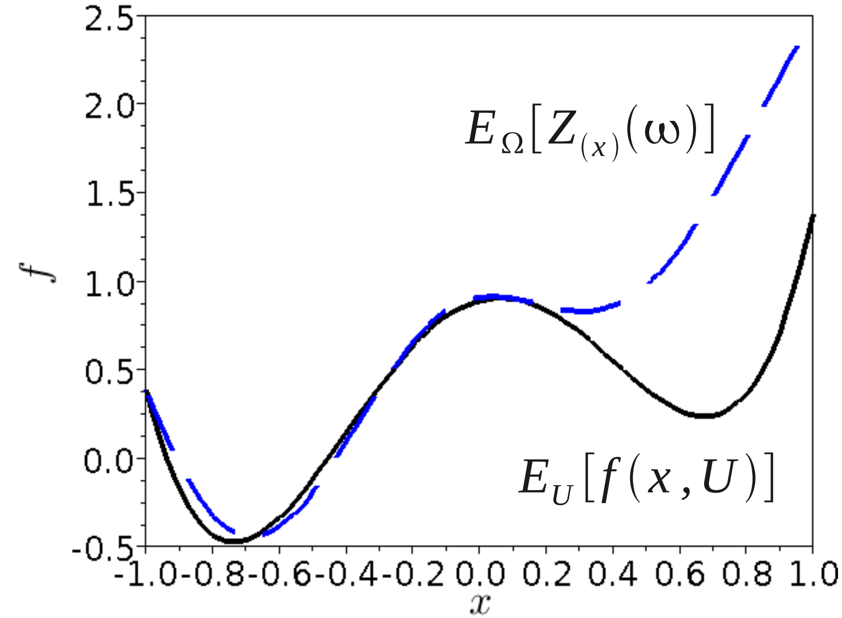


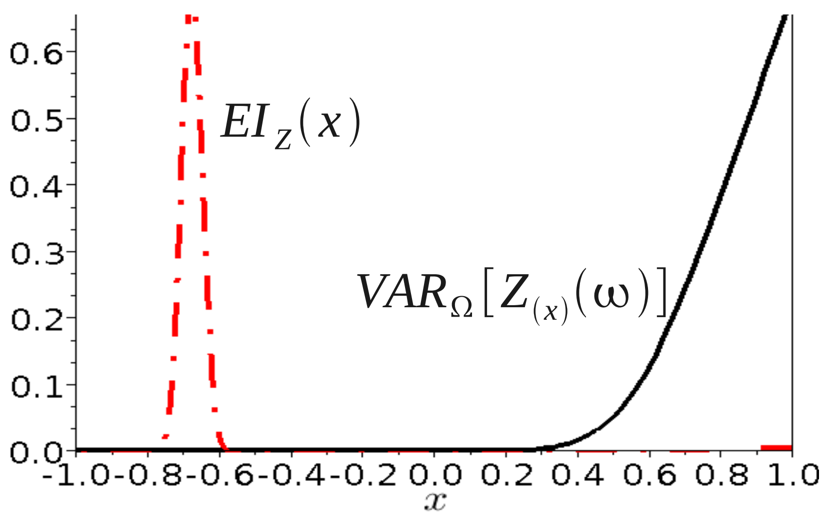




Kriging based optimization with uncertainties, U controlled
5th iteration





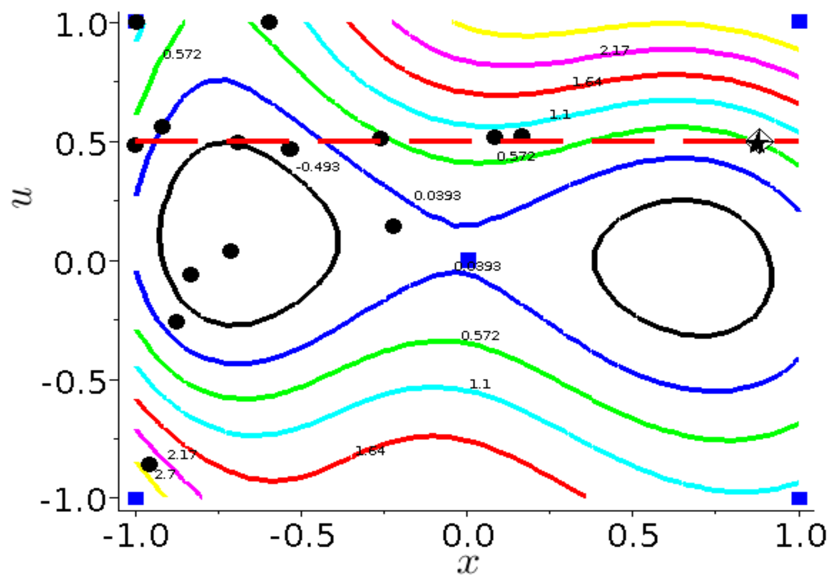


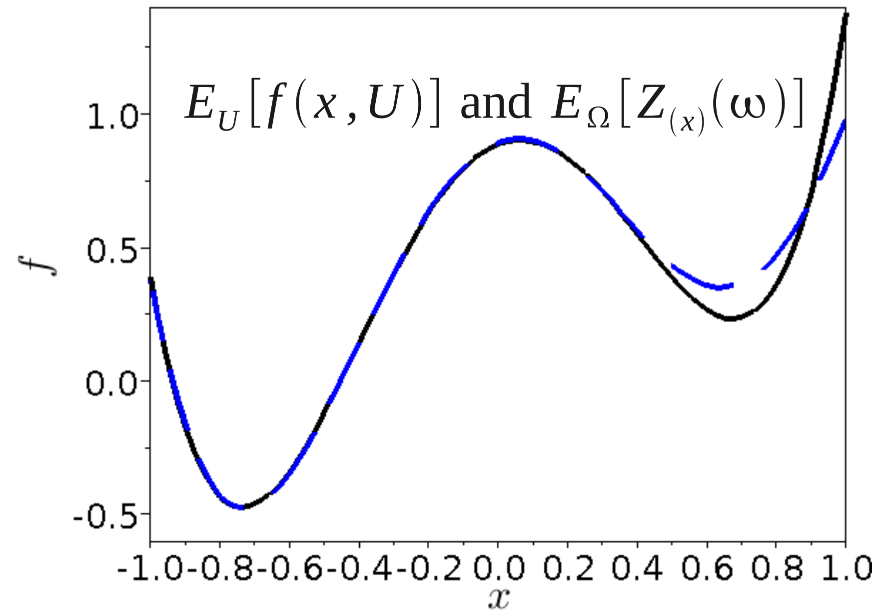
◈ – (x^{next}, μ)

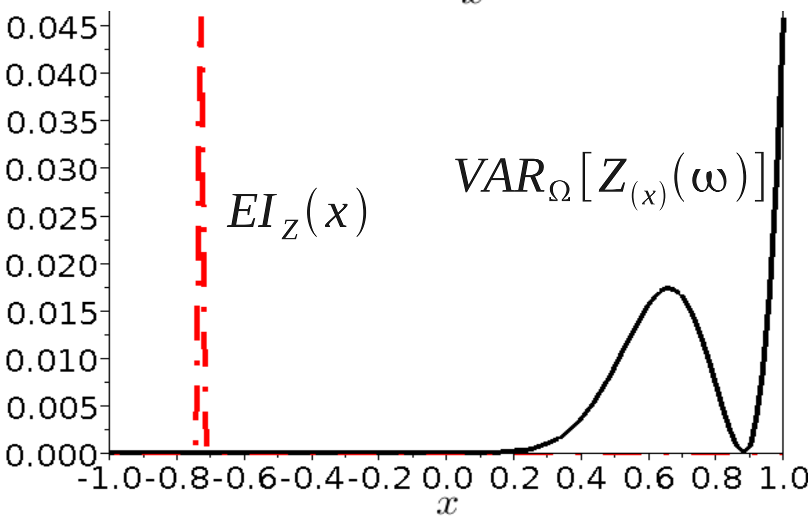
★ – (x^{t+1}, u^{t+1})

Kriging based optimization with uncertainties, U controlled
17th iteration

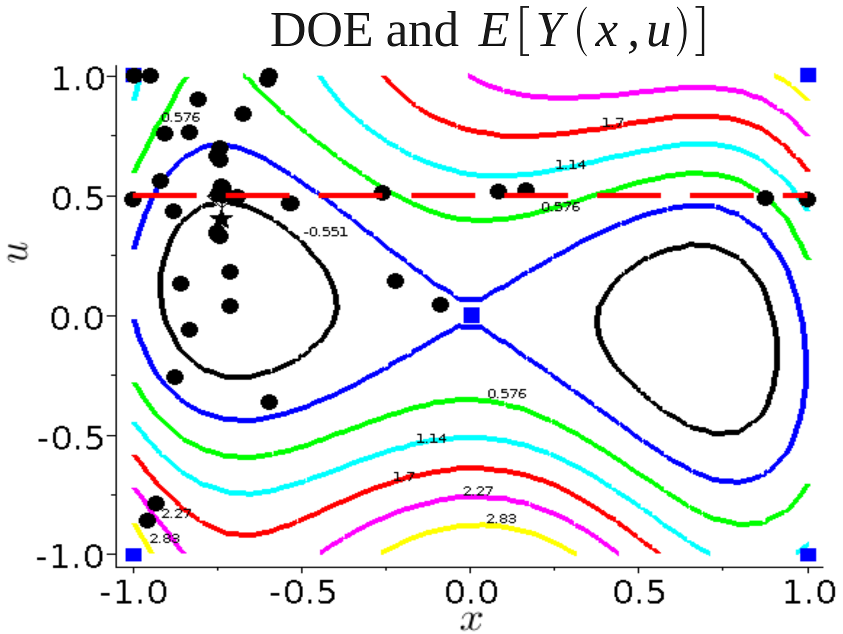
DOE and $E[Y(x,u)]$

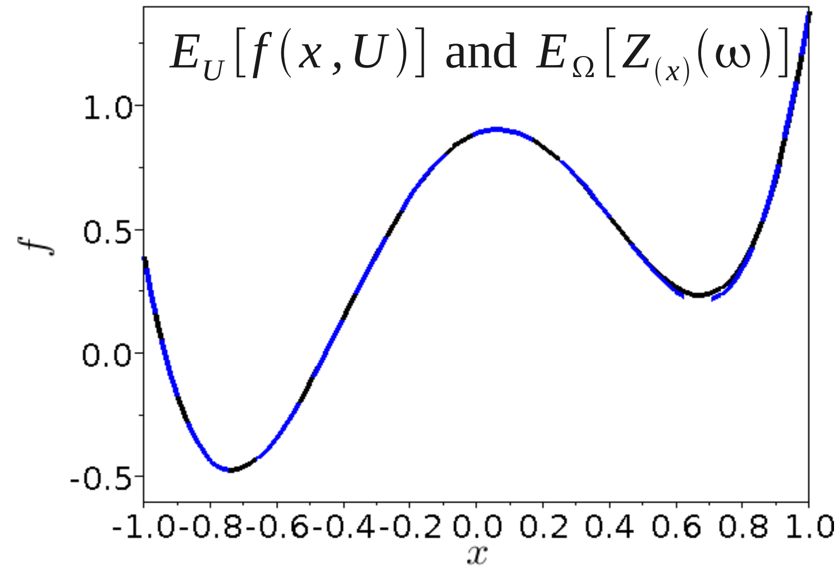


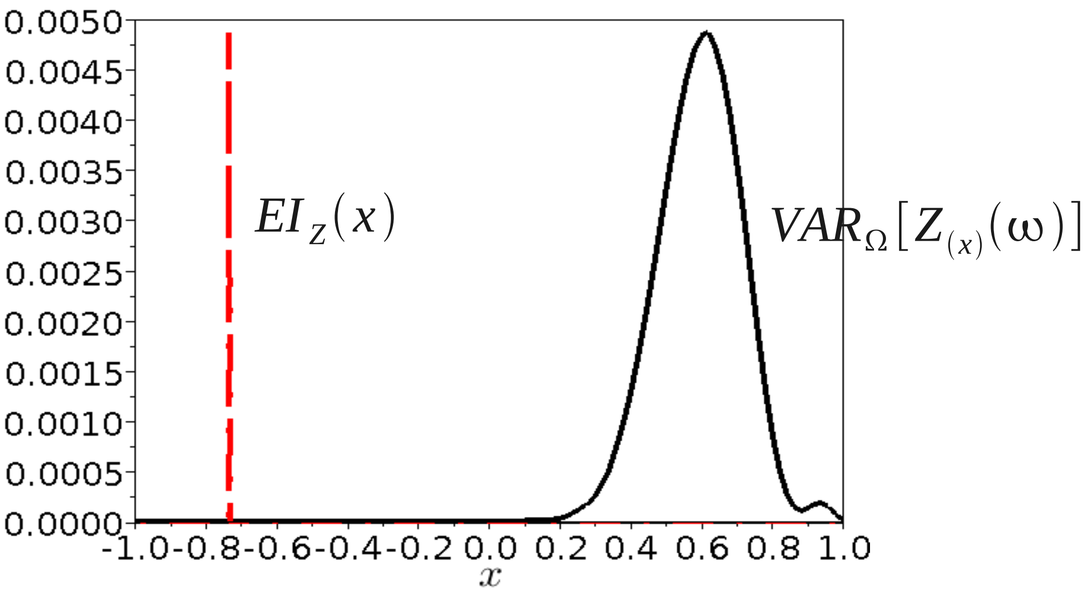




Kriging based optimization with uncertainties, U controlled
50th iteration







Comparison tests

Compare « simultaneous opt and sampling » method to

1. A direct MC based approach :
   EGO based on MC simulations in *f* with fixed number of runs, *s*.
   Kriging with homogenous nugget to filter noise.

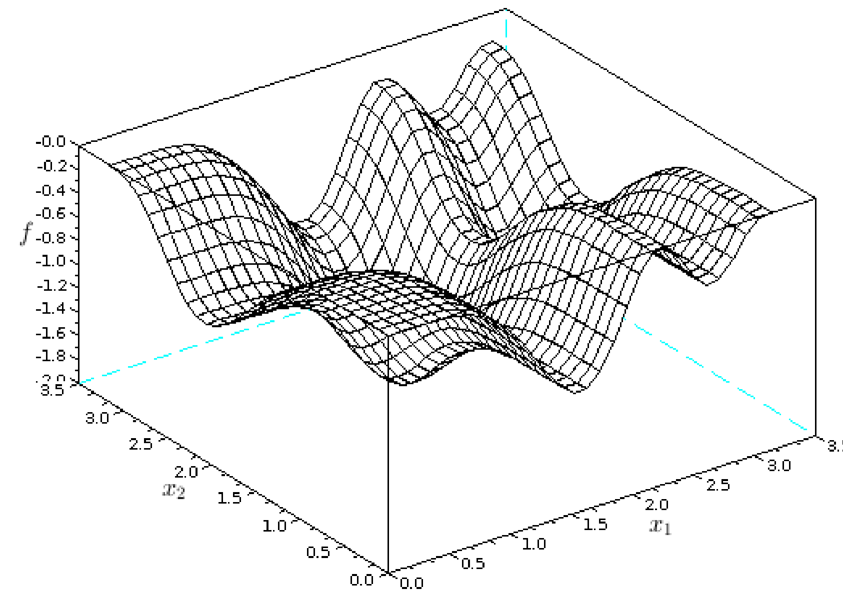
2. An MC-surrogate based approach :
   the MC-kriging algorithm.

Test functions

Test cases based on Michalewicz function

$$f(x) = -\sum_{i=1}^{n} \sin(x_i)\left[\sin(i x_i^2/\pi)\right]^2$$

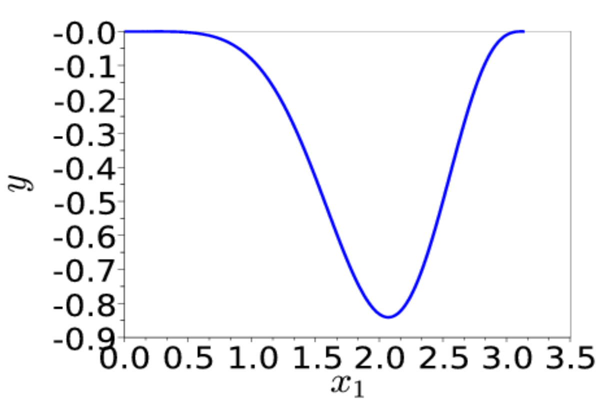
$$f(x, u) = f(x) + f(u)$$

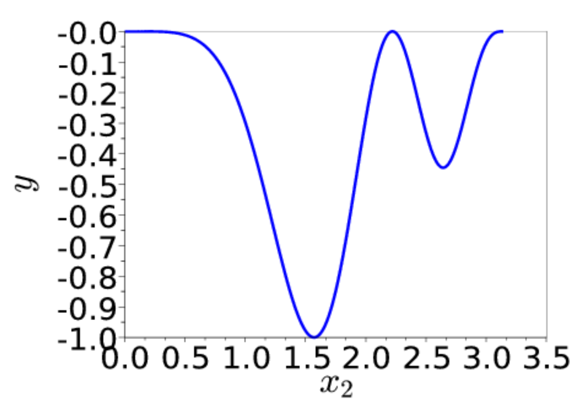


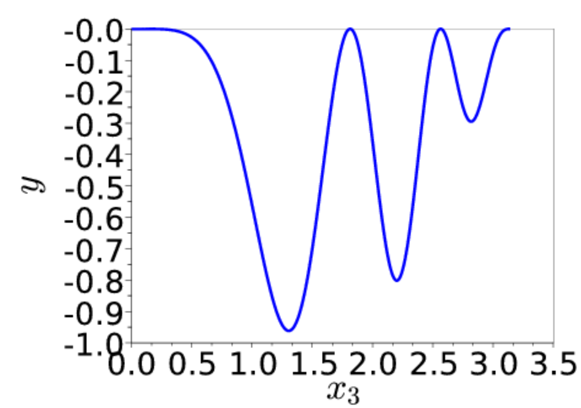
2D: $n_x=1 \quad n_u=1 \quad \mu=1.5 \quad \sigma=0.2$

4D: $n_x=2 \quad n_u=2 \quad \mu=[1.5,\ 2.1] \quad \sigma=[0.2,\ 0.2]$

6D: $n_x=3 \quad n_u=3 \quad \mu=[1.5,\ 2.1,\ 2] \quad \sigma=[0.2,\ 0.2,\ 0.3]$

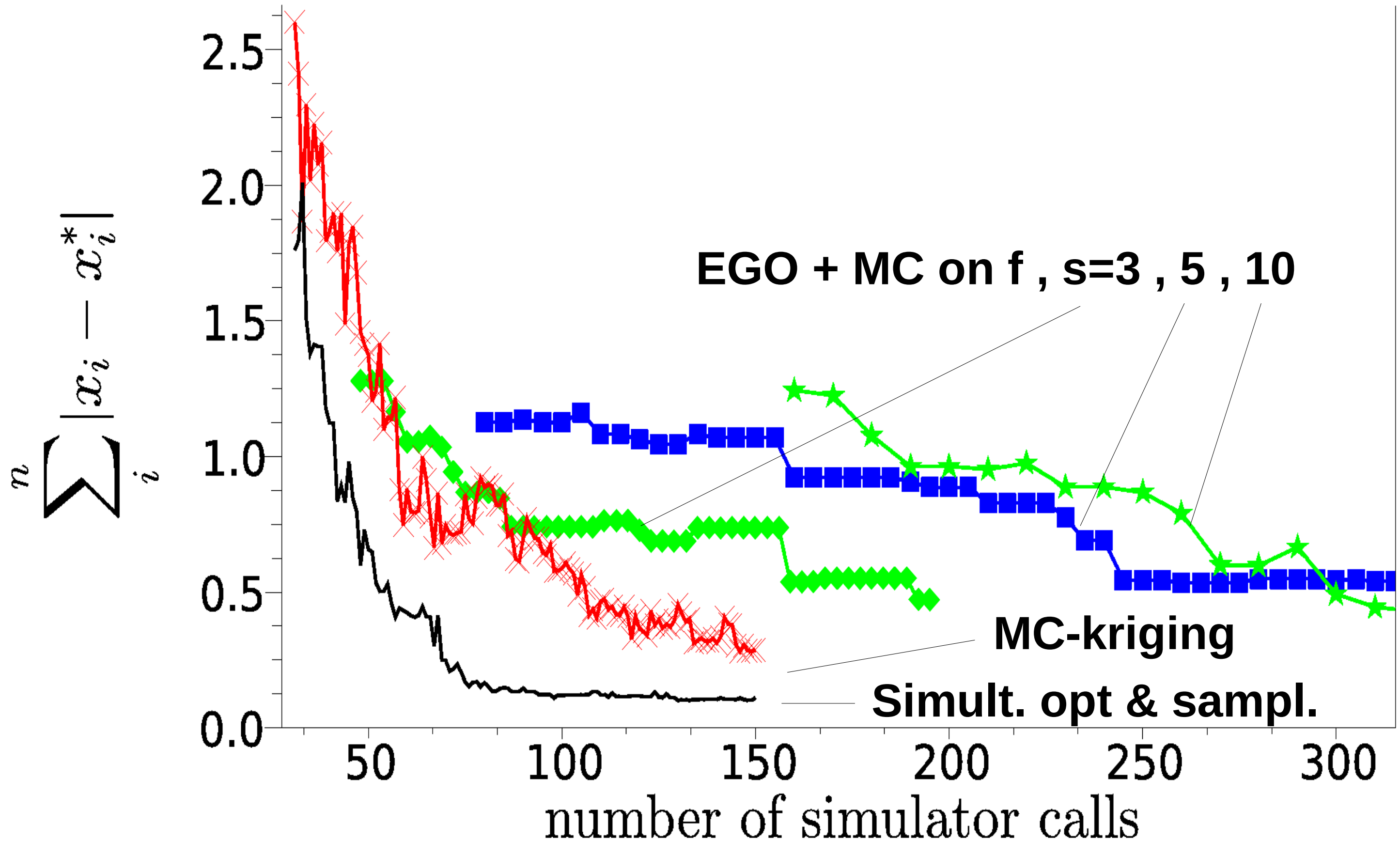






Test results

6D Michalewicz test case, $n_{x=3}$ =3 , n_U =3 .
Initial DOE: RLHS , $m=(n_x+n_U)*5 = (3+3)*5 = 30$;
10 runs for every method.



Partial conclusion

- min_x *f(x,U)*

We have discussed spatial statistics to filter the noise → kriging based approaches.

- Limitation : number of dimensions , dim(x) + dim(U) < 20

Beyond this, if the function f is not too costly, use stochastic evolutionary optimizers, which can be relatively robust to noise if properly tuned.

- Useful for optimizing statistical estimators which are noisy.

- No control over the U's
- No spatial statistics (i.e. in S or $S \times U$ spaces), pointwise approaches only.

Outline of the talk

1. Motivations for robust optimization

2. Formulations of optimization problems with uncertainties

3. Kriging-based approaches (costly functions)
   No control on U
   With control on U

4. Evolutionary approaches (non costly functions)
   The general CMA-ES
   Improvements for noisy functions :
      Mirrored sampling and sequential selection
      ~~Adding confidence to an ES~~

Evolutionary algorithms

Taking search decisions in probability is a way to handle the noise corrupting observed *f* values

→ use a stochastic optimizer, an evolution strategy (ES).

« elitism »

A simple (1+1)-ES

Initializations : x, $f(x)$, m, C, t_{max}.

While $t < t_{max}$ do,

Sample $N(m,C)$ --> x'

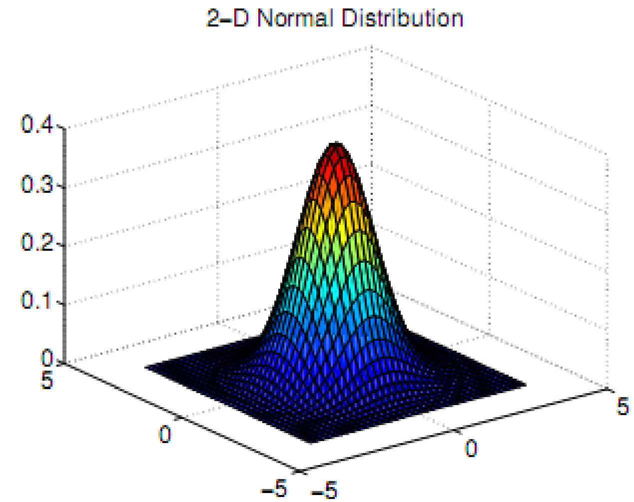
Calculate $f(x')$, $t = t+1$

If $f(x')<f(x)$, $x = x'$, $f(x) = f(x')$ Endif

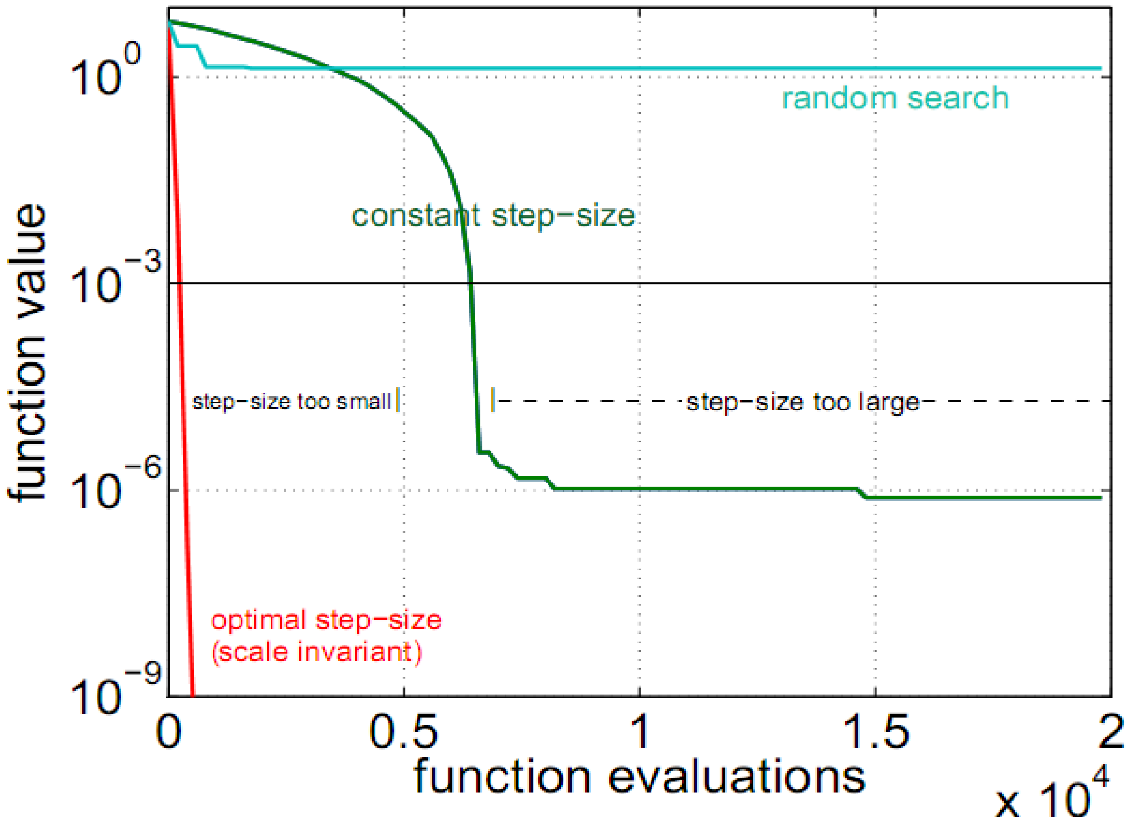
Update m (e.g., $m=x$) and C

End while

```
%(Scilab code)
x = m + grand(1,'mn',0,C)
```



Adapting the step size (C^2) is important



$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$$

in $[-0.2, 0.8]^n$
for $n = 10$

(A. Auger et N. Hansen, 2008)

Above isotropic ES(1+1) : $\boldsymbol{C} = \sigma^2 \mathrm{I}$, σ is the step size.
With an optimal step size ($\approx \|x\| / d$) on the sphere function, log linear speed that degrades only in $O(d)$.

The population based CMA-ES

(N. Hansen et al., since 1996, now with A. Auger)

CMA-ES = *Covariance Matrix Adaptation Evolution Strategy* = optimization through sampling and updating of a multi-normal distribution.

A fully populated covariance matrix is build : pairwise variables interactions learned. Can adapt the step in any direction.

The state-of-the-art evolutionary / genetic optimizer for continuous variables.

flow-chart of CMA-ES

CMA-ES is an evolution strategy *ES*-(μ,λ) :

Initializations : m, C, t_{max}, μ , λ

While $t < t_{max}$ do,

  Sample $N(m,C)$ --> $x^1,\ldots,x^\lambda$

  Calculate $f(x^1),\ldots,f(x^\lambda)$, $t = t+\lambda$

  Rank : $f(x^{1:\lambda}),\ldots,f(x^{\lambda:\lambda})$

  Update m and C *with the* μ bests, $x^{1:\lambda},\ldots,x^{\mu:\lambda}$

End while

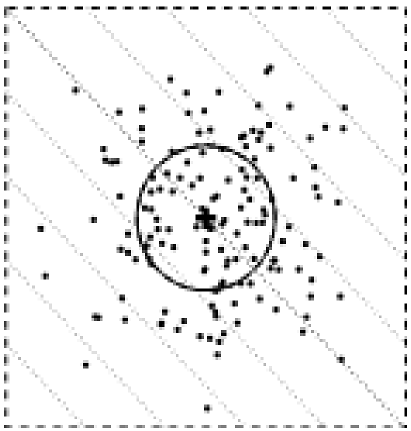
m *et* C are updated with

- the best **steps** (as opposed to points),
- a **time cumulation** of these best steps.

CMA-ES : adapting C^2 with good steps

(A. Auger et N. Hansen, 2008)

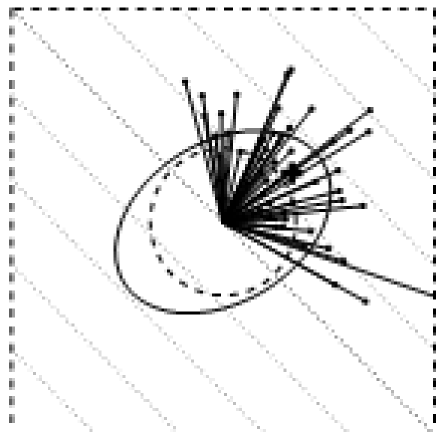
Initialization : $m \in S \;,\; C = I \;,\; c_{cov} \approx 2/n^2$



sampling

$$x^i = m + y^i$$
$$y^i \propto N(0, C)$$
$$i = 1, \ldots, \lambda$$

calculate f

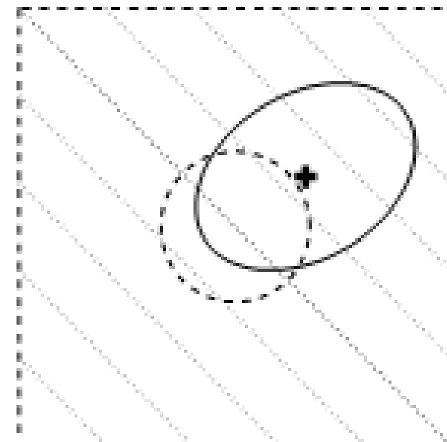


selection

$$y_w = \frac{1}{\mu} \sum_{i=1}^{\mu} y^{i:\lambda}$$

rank 1 C update

$$C \leftarrow (1 - c_{cov}) C + c_{cov} \mu \, y_w \, y_w^T$$



update m

$$m \leftarrow m + y_w$$

The state-of-the-art CMA-ES

(A. Auger and N. Hansen, *A restart CMA evolution strategy with increasing population size*, 2005)

Additional features :

- Steps weighting, $y_w = \sum_{i=1}^{\mu} w_i \, y^{i:\lambda}$
- Time cumulation of the steps.
- Simultaneous rank 1 and μ covariance adaptations.
- Use of a global scale factor, $C \rightarrow \sigma^2 C$.
- Restarts with increasing population sizes (unless it is the 2010 version with mirrored sampling and sequential selection, see later)

Has been used up to d ~ 1000 continuous variables.

Noisy optimization

- The general CMA-ES
- Improvements for noisy functions :
  Mirrored sampling and sequential selection

Resampling, noise and evolutionary algorithms

CMA-ES(μ,λ) can optimize many noisy functions because
1. it is not elitist
2. the choice of the next iteration average

$$m^{t+1} = m^t + \frac{1}{\mu}\sum_{i=1}^{\mu} y^{i:\lambda}$$

averages out errors (spatial sampling as a proxy for U sampling)

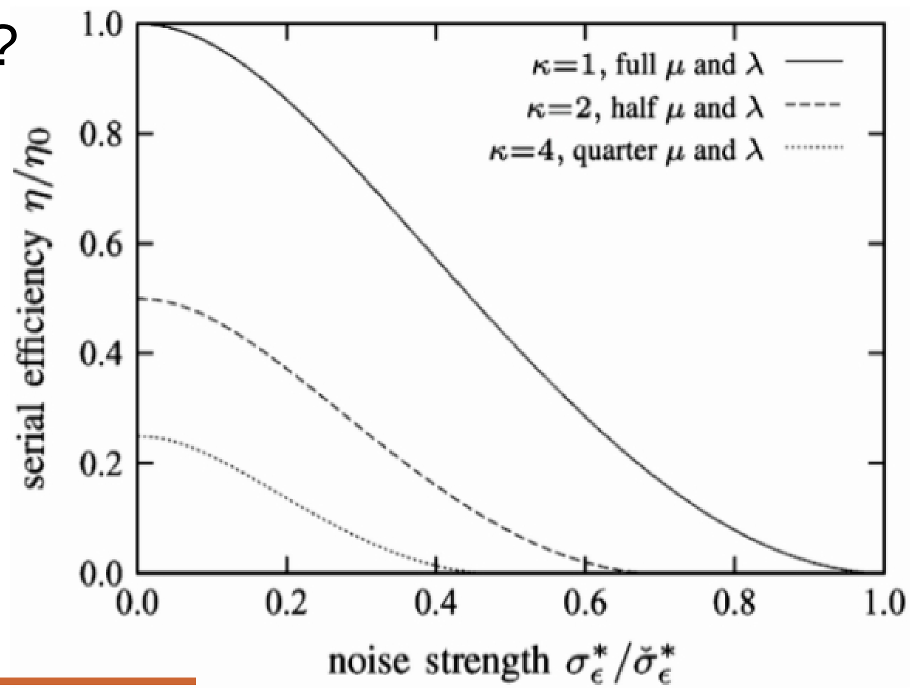
To improve convergence on noisy function, is it preferable

1. to resample $\hat{\hat{f}}(x) = \frac{1}{\kappa}\sum_{i=1}^{\kappa}\hat{f}^{(i)}(x)$

2. or to increase the population size ?

(for an equivalent increase in computation)

→ it is better to increase the population size. *[Beyer and Sendhoff 2007, Arnold and Beyer 2006]*



But one can still do better ...

Mirrored sampling and sequential selection (1)

D. Brockhoff, A. Auger, N. Hansen, D. V. Arnold, and T. Hohm. *Mirrored Sampling and Sequential Selection for Evolution Strategies,* PPSN XI, 2010

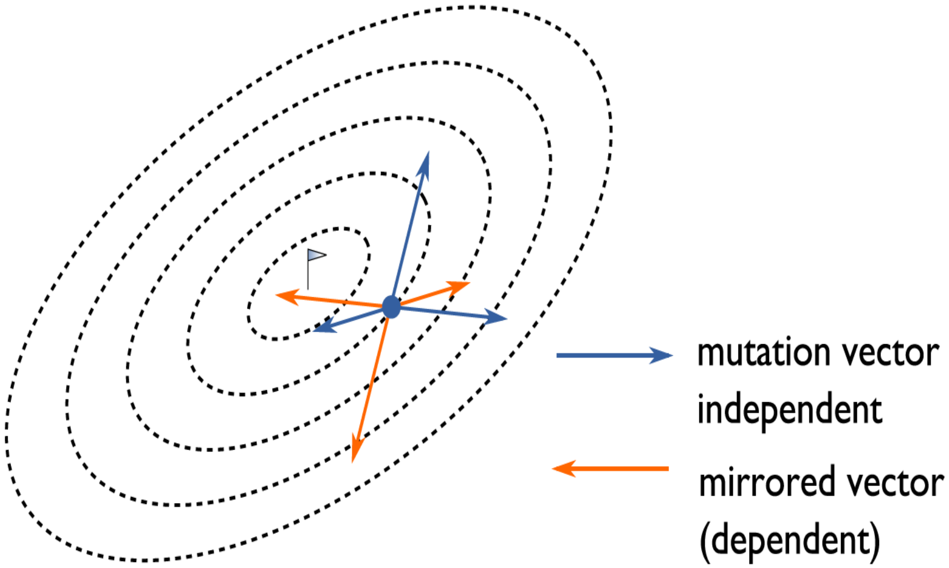
A. Auger, D. Brockhoff, N. Hansen, *Analysing the impact of mirrored sampling and sequential selection in elitist Evolution Strategies,* FOGA 2011

(1+1)-CMA-ES with restarts surprisingly good on some functions (including multimodal functions with local optima) ← small population advantage.

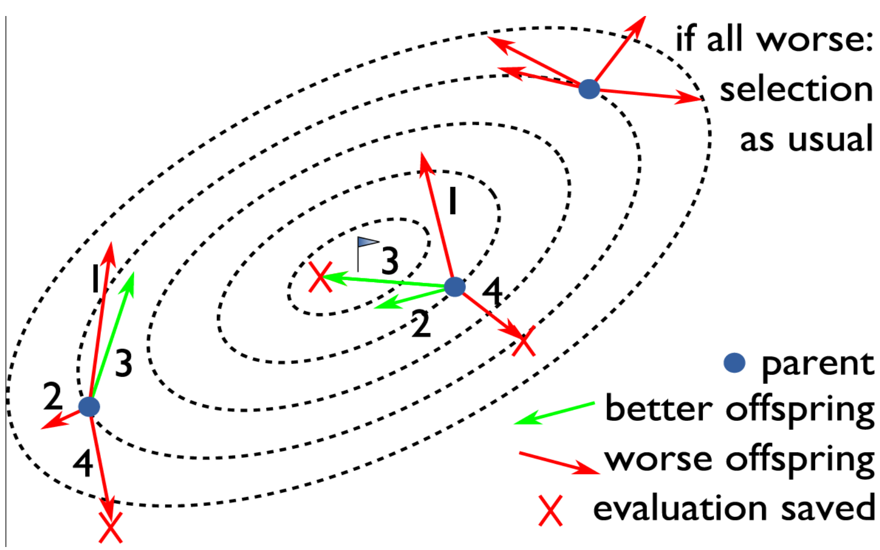
But « elitism » of (1+1)-ES bad for noisy functions : a lucky sample attracts the optimizer in a non-optimal region of the search space.

Question : how to design a fast local non-elistist ES ?

Mirrored sampling and sequential selection (2)



Derandomization via mirrored sampling : one random vector generates two offsprings.
Often good and bad in opposite directions.



Sequential selection : stop evaluation of new offsprings as soon as a solution better than the parent is found.

Combine the two ideas : when an offspring is better than its parent, its symmetrical is worse (on convex level sets), and vice versa → evaluate in order $m+y^1$, $m-y^1$, $m+y^2$, $m-y^2$, … .

Mirrored sampling and sequential selection (3)

Results :

small population, no elitism

(1,4)-ES with mirroring and sequential selection faster than (1+1)-ES on sphere function.
Theoretical result: Convergence Rate* ES (1+1)=0.202 ,
Convergence Rate (1,4ms)=0.223 .

[Brockhoff et al., Mirrored sampling and sequential selection for evolution strategies, 2010.]

Implementation within CMA-ES, tested in BBOB'2010** (Black Box Optimization Benchmarking)
Best performance among all algorithms tested so far on some functions of noisy testbed

* convergence rate $\equiv -\lim_{t \to \infty} \frac{\ln(\text{distance to optimum})}{t}$,
cf. slope line of $(\log(f), \text{time})$ earlier

** http://coco.gforge.inria.fr/bbob2010-downloads

Concluding remarks (1)

Today's story was :

• Optimization → difficult in the presence of noise → formulation of optimization in the presence of uncertainties → noisy functions

• → do spatial stats (kriging) [optimizer without U control → optimizer with U control]

•→stochastic optimizers directly applied to noisy functions.

Each method has its application domain :

• Stochastic optimizers robust to noise cannot be directly applied to an expensive (simulation based) objective function. An intermediate surrogate is needed.

• Vice versa, kriging based method involve large side calculations : they are interesting only for expensive *f*'s.

Concluding remarks (2)

Many (most) methods were not discussed :

- **Method of moments (Taylor expansions of the opt. criteria),**
- **FORM/SORM (local constraints approximations about probable points) ,**
- **Chance constraints and convex programming (worst U cases) … .**

A lot still to be done :

- **effect of the a priori uncertainty model (law of random parameters),**
- **optimize quantiles,**
- **statistically joined criteria,**
- **kriging like approaches (spatial stats) in high dimension,**
- **…**

Bibliography

Related books :

• A. Ben-Tal, L. El Ghaoui, A. Nemirowski, *Robust Optimization*, Princeton Univ. Press, 2009.
• R. E. Melchers, *Structural Reliability Analysis and Prediction*, Wiley, 1999.
• M. Lemaire, A. Chateauneuf, J.-C. Mitteau, *Structural Reliability*, Wiley, 2009.
• J. C. Spall, *Introduction to Stochastic Search and Optimization*, Wiley, 2003.
• *Multidisciplinary Design Optimization in Computational Mechanics*, P. Breitkopf and R. Filomeno Coehlo Eds., Wiley/ISTE, 2010.

Articles :

• H.G. Beyer, B. Sendhoff, *Robust Optimization – A comprehensive survey*, Comput. Methods Appl. Mech. Engrg, 196, pp. 3190-3218, 2007.

• J. Janusevskis and R. Le Riche, *Simultaneous kriging-based estimation and optimization of mean response*, Journal of Global Optimization, Springer, 2012

• G. Pujol, R. Le Riche, O. Roustant and X. Bay, *L'incertitude en conception: formalisation, estimation,* Chapter 3 of the book *Optimisation Multidisciplinaire en Mécaniques : réduction de modèles, robustesse, fiabilité, réalisations logicielles*, Hermes, 2009 (in French)

• D. Salazar, R. Le Riche, G. Pujol and X. Bay, *An empirical study of the use of confidence levels in RBDO with Monte Carlo simulations*, in Multidisciplinary Design Optimization in Computational Mechanics, Wiley/ISTE Pub., 2010.