Two advances in GP-based prediction and optimization for computer experiments

David Ginsbourger

Institut de Mathématiques & Centre d'Hydrogéologie, Université de Neuchâtel, and Ecole Nat. Sup. des Mines de Saint-Etienne (Ph.D. defense coming soon :)

Journées du GdR MASCOT-NUM, Day 1 (18/03/2009)
University Paris XIII - Galilée institute

Outline

General context

Amazing growth of computing power in numerical simulation

- Powerful processors, clustering.
- Scientific computing has reached maturity: FEM, Monte-Carlo, etc.
- Paradox: since one always want more accurate results, computation times are often stagnating and even sometimes increasing!

Examples of application domains

- Crash-test simulation: $\approx 20h$ per "run"
- Nuclear plant: $> 1hr$ to estimate neutronic criticality of a set of fuel rods
- Simulation of the behaviour of a CO_2 bubble stocked during 10000 years in a natural geological reservoir: several days.

General context

Amazing growth of computing power in numerical simulation

- Powerful processors, clustering.
- Scientific computing has reached maturity: FEM, Monte-Carlo, etc.
- Paradox: since one always want more accurate results, computation times are often stagnating and even sometimes increasing!

Examples of application domains

- Crash-test simulation: $\approx 20h$ per "run"
- Nuclear plant: $> 1hr$ to estimate neutronic criticality of a set of fuel rods
- Simulation of the behaviour of a CO_2 bubble stocked during 10000 years in a natural geological reservoir: several days.

Gaussian Processes (GP) and functional learning

Approximating deterministic functions using GP's (Kriging)

$$y : \mathbf{x} \in D \subset \mathbb{R}^d \longrightarrow y(\mathbf{x}) \in \mathbb{R}$$

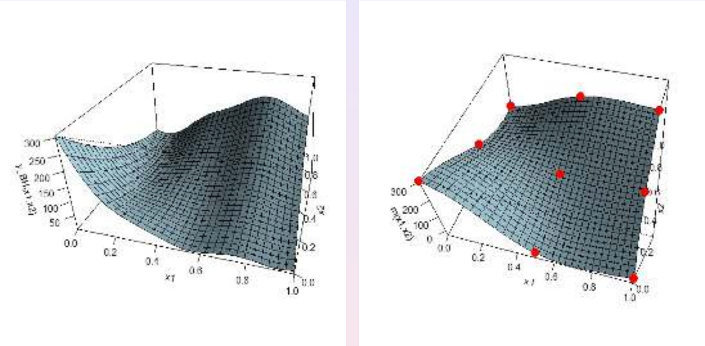
y is seen as one realization of a GP Y_x with mean μ and covariance kernel $k \in (\mathbf{x}, \mathbf{x}') \in D \times D \subset \mathbb{R}^d \times \mathbb{R}^d \longrightarrow k(\mathbf{x}, \mathbf{x}') \in \mathbb{R}$.

Basic assumptions and features of GP modeling

1. Kriging $\approx$ approximating y by conditioning $Y_\mathbf{x}$ on the observations at a learning set, both denoted by $\{\mathbf{X}, \mathbf{Y}\} = \{(\mathbf{x}^1, ..., \mathbf{x}^n), (y(\mathbf{x}^1), ..., y(\mathbf{x}^n))\}$
2. In practice, a parametric k is chosen beforhand (typically powered exponential) and the parameters are estimated based on $(\mathbf{X}, \mathbf{Y})$.

Industrial examples (DICE consortium): crash-test simulations, nuclear criticality studies, optimal conception of high-tech devices...

Example of Ordinary Kriging Interpolation



Interpolation (right) of Branin's function (left), known at 9 points (in red).

The kernel is $k(\mathbf{x}, \mathbf{x}') = \sigma^2 e^{-\sum_{j=1}^{2} \left(\frac{\mathbf{x}_j - \mathbf{x}'_j}{\psi_j} \right)^2}$ with (σ^2, ψ) estimated by ML

Ordinary Kriging Equations

A central property of OK, when k is known (and μ has a $\mathcal{U}(\mathbb{R})$ prior...)

$$[Y(\mathbf{x})|Y(\mathbf{X}) = \mathbf{Y}] \sim \mathcal{N}\left(m(\mathbf{x}), s^2(\mathbf{x})\right)$$

$$\begin{cases} m(\mathbf{x}) = \mathbb{E}[Y(\mathbf{x})|Y(\mathbf{X}) = \mathbf{Y}] = \widehat{\mu} + \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1}(\mathbf{Y} - \widehat{\mu}\mathbb{1}_n) \\ s^2(x) = Var[Y(\mathbf{x})|Y(\mathbf{X}) = \mathbf{Y}] = \sigma^2 - \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1}\mathbf{k}(\mathbf{x}) + \frac{\left(1 - \mathbb{1}_n^T \mathbf{K}^{-1}\mathbf{k}(\mathbf{x})\right)^2}{\left(\mathbb{1}_n^T \mathbf{K}^{-1}\mathbb{1}_n\right)} \end{cases}$$

$$\widehat{\mu} = \frac{\mathbb{1}^T \mathbf{K}^{-1}\mathbf{Y}}{(\mathbb{1}^T \mathbf{K}^{-1}\mathbb{1})}, \mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \dots & \dots & \dots. & \dots. \\ k(\mathbf{x}_n, \mathbf{x}_1) & \dots & \dots. & k(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix} \text{ and } \mathbf{k}(\mathbf{x}) = \begin{pmatrix} k(\mathbf{x}, \mathbf{x}_1) \\ k(\mathbf{x}, \mathbf{x}_2) \\ \dots \\ k(\mathbf{x}, \mathbf{x}_n) \end{pmatrix}$$

Ordinary Kriging Equations

A central property of OK, when k is known (and μ has a $\mathcal{U}(\mathbb{R})$ prior...)

$$[Y(\mathbf{x})|Y(\mathbf{X}) = \mathbf{Y}] \sim \mathcal{N}\left(m(\mathbf{x}), s^2(\mathbf{x})\right)$$

$$\begin{cases} m(\mathbf{x}) = \mathbb{E}[Y(\mathbf{x})|Y(\mathbf{X}) = \mathbf{Y}] = \widehat{\mu} + \mathbf{k}(\mathbf{x})^T\mathbf{K}^{-1}\left(\mathbf{Y} - \widehat{\mu}\mathbb{1}_n\right) \\ s^2(x) = Var[Y(\mathbf{x})|Y(\mathbf{X}) = \mathbf{Y}] = \sigma^2 - \mathbf{k}(\mathbf{x})^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}) + \frac{\left(1-\mathbb{1}_n^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x})\right)^2}{\left(\mathbb{1}_n^T\mathbf{K}^{-1}\mathbb{1}_n\right)} \end{cases}$$

$$\widehat{\mu} = \frac{\mathbb{1}^T\mathbf{K}^{-1}\mathbf{Y}}{(\mathbb{1}^T\mathbf{K}^{-1}\mathbb{1})}, \mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & k(\mathbf{x}_1,\mathbf{x}_2) & \dots & k(\mathbf{x}_1,\mathbf{x}_n) \\ k(\mathbf{x}_2,\mathbf{x}_1) & k(\mathbf{x}_2,\mathbf{x}_2) & \dots & k(\mathbf{x}_2,\mathbf{x}_n) \\ \dots & \dots & \dots. & \dots. \\ k(\mathbf{x}_n,\mathbf{x}_1) & \dots & \dots. & k(\mathbf{x}_n,\mathbf{x}_n) \end{pmatrix} \text{ and } \mathbf{k}(\mathbf{x}) = \begin{pmatrix} k(\mathbf{x},\mathbf{x}_1) \\ k(\mathbf{x},\mathbf{x}_2) \\ \dots \\ k(\mathbf{x},\mathbf{x}_n) \end{pmatrix}$$

Ordinary Kriging Equations

A central property of OK, when k is known (and μ has a $\mathcal{U}(\mathbb{R})$ prior...)

$$[Y(\mathbf{x})|Y(\mathbf{X})=\mathbf{Y}] \sim \mathcal{N}\left(m(\mathbf{x}), s^2(\mathbf{x})\right)$$

$$\begin{cases} m(\mathbf{x}) = \mathbb{E}[Y(\mathbf{x})|Y(\mathbf{X})=\mathbf{Y}] = \widehat{\mu} + \mathbf{k}(\mathbf{x})^T\mathbf{K}^{-1}(\mathbf{Y}-\widehat{\mu}\mathbb{1}_n) \\ s^2(x) = Var[Y(\mathbf{x})|Y(\mathbf{X})=\mathbf{Y}] = \sigma^2 - \mathbf{k}(\mathbf{x})^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}) + \frac{\left(1-\mathbb{1}_n^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x})\right)^2}{\left(\mathbb{1}_n^T\mathbf{K}^{-1}\mathbb{1}_n\right)} \end{cases}$$

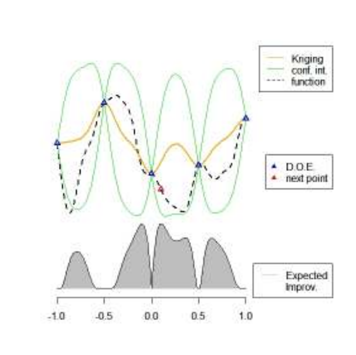
$$\widehat{\mu} = \frac{\mathbb{1}^T\mathbf{K}^{-1}\mathbf{Y}}{(\mathbb{1}^T\mathbf{K}^{-1}\mathbb{1})}, \mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & k(\mathbf{x}_1,\mathbf{x}_2) & \dots & k(\mathbf{x}_1,\mathbf{x}_n) \\ k(\mathbf{x}_2,\mathbf{x}_1) & k(\mathbf{x}_2,\mathbf{x}_2) & \dots & k(\mathbf{x}_2,\mathbf{x}_n) \\ \dots & \dots & \dots. & \dots. \\ k(\mathbf{x}_n,\mathbf{x}_1) & \dots & \dots. & k(\mathbf{x}_n,\mathbf{x}_n) \end{pmatrix} \text{ and } \mathbf{k}(\mathbf{x}) = \begin{pmatrix} k(\mathbf{x},\mathbf{x}_1) \\ k(\mathbf{x},\mathbf{x}_2) \\ \dots \\ k(\mathbf{x},\mathbf{x}_n) \end{pmatrix}$$

The *Expected improvement* (EI) criterion

Expected Improvement

$$EI(\mathbf{x}) = \mathbb{E}\left[(\min(Y(\mathbf{X})) - Y(\mathbf{x}))^{+} \,|\, Y(\mathbf{X}) = \mathbf{Y}\right]$$

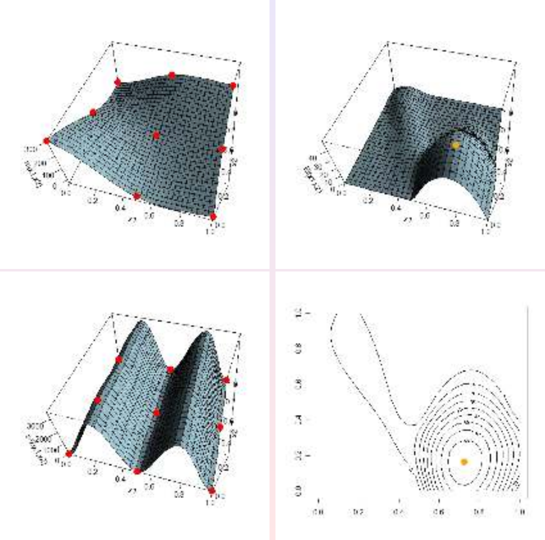
M. Schonlau, W.J. Welch and D.R. Jones.
Efficient Global Optimization of Expensive Black-box Functions
Journal of Global Optimization, 1998.



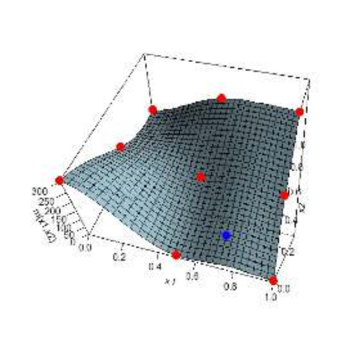
$$EI(\mathbf{x}) = (\min(y(\mathbf{X})) - m(\mathbf{x}))\,\Phi\left(\frac{\min(y(\mathbf{X})) - m(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{\min(y(\mathbf{X})) - m(\mathbf{x})}{s(\mathbf{x})}\right),$$

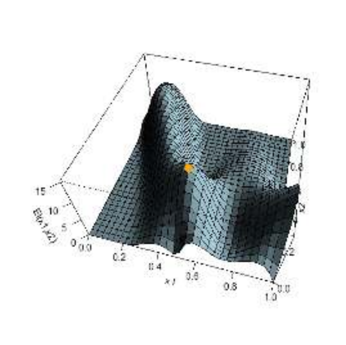
where Φ and ϕ are the cdf and pdf of the standard gaussian law, respectively.

Kriging-based optimization with EGO

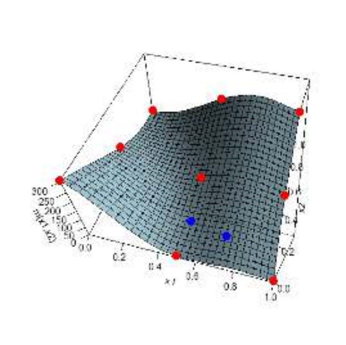


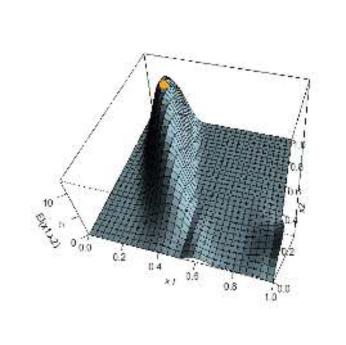
Kriging-based optimization with EGO



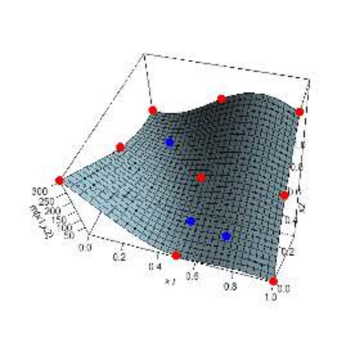


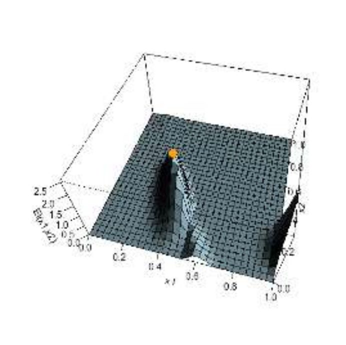
Kriging-based optimization with EGO



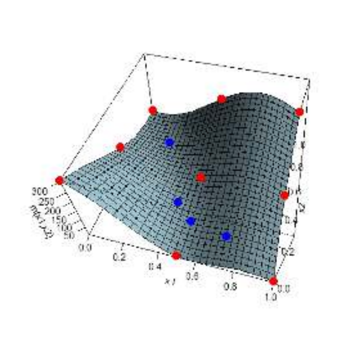


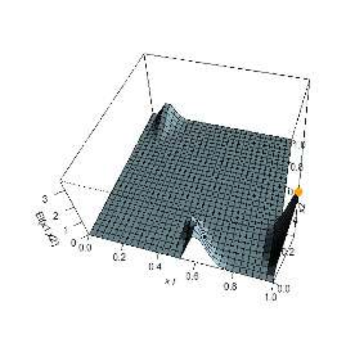
Kriging-based optimization with EGO



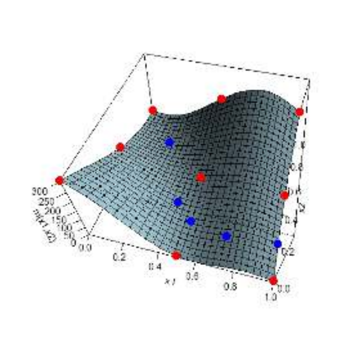


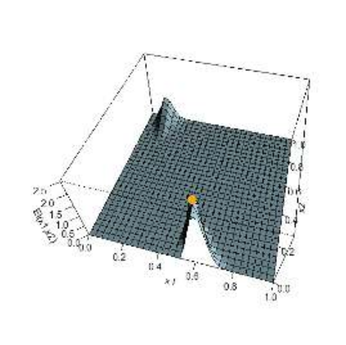
Kriging-based optimization with EGO



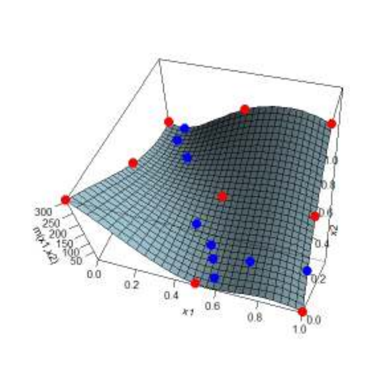


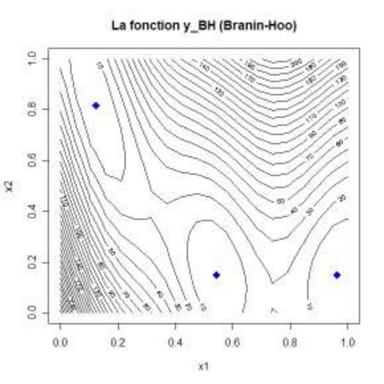
Kriging-based optimization with EGO





Kriging-based optimization with EGO: results





EGO: a sequential procedure

Sketch of the EGO Algorithm

function EGO($\mathbf{X}$, $\mathbf{Y}$, p)
  for $i \leftarrow 1, p$ **do**
    $\mathbf{x}_i^{new} = argmax_{\mathbf{x} \in D} \mathbb{E}I(\mathbf{x})$ ▷ with updated **X** and **Y**
    $\mathbf{X} = \mathbf{X} \bigcup \{\mathbf{x}_i^{new}\}$
    $\mathbf{Y} = \mathbf{Y} \bigcup \{y(\mathbf{x}_i^{new})\}$ ▷ with updated **X** and **Y**
    Update the covariance parameters and the kriging model
  end for
end function

Major issue with sequentiality:
In industrial context, the project duration is more crucial than computation time, provided that the latter can be distributed on multiple processors.

Algorithms such as EGO may be wasteful... they need to be parallelized!

Question: how to evaluate the added value of sampling q points?

EGO: a sequential procedure

Sketch of the EGO Algorithm

```
function EGO(X, Y, p)
    for i ← 1, p do
        x_i^new = argmax_{x∈D} 𝔼I(x)                ▷ with updated X and Y
        X = X ⋃ {x_i^new}
        Y = Y ⋃ {y(x_i^new)}                        ▷ with updated X and Y
        Update the covariance parameters and the kriging model
    end for
end function
```

Major issue with sequentiality:
In industrial context, the project duration is more crucial than computation time, provided that the latter can be distributed on multiple processors.

Algorithms such as EGO may be wasteful... they need to be parallelized!

Question: how to evaluate the added value of sampling q points?

EGO: a sequential procedure

Sketch of the EGO Algorithm

function EGO($\mathbf{X}$, $\mathbf{Y}$, p)
    for $i \leftarrow 1, p$ **do**
        $\mathbf{x}_i^{new} = argmax_{\mathbf{x} \in D} \mathbb{E}I(\mathbf{x})$ ▷ with updated **X** and **Y**
        $\mathbf{X} = \mathbf{X} \bigcup \{\mathbf{x}_i^{new}\}$
        $\mathbf{Y} = \mathbf{Y} \bigcup \{y(\mathbf{x}_i^{new})\}$ ▷ with updated **X** and **Y**
        Update the covariance parameters and the kriging model
    end for
end function

Major issue with sequentiality:
In industrial context, the project duration is more crucial than computation time, provided that the latter can be distributed on multiple processors.

Algorithms such as EGO may be wasteful... they need to be parallelized!

Question: how to evaluate the added value of sampling q points?

Outline

An introduction to the q-points $\mathbb{E}$I

Background: $(\mathbf{x}_1^{new}, ..., \mathbf{x}_q^{new})$ has to be chosen such that the improvement brought after evaluating y at the q locations be as large as possible.
By definition, we wish them to (*a posteriori*) maximize:

$$i(\mathbf{x}_1^{new}, ..., \mathbf{x}_q^{new}) := \max\{[\min(y(\mathbf{X})) - y(\mathbf{x}_1^{new})]^+, ..., [\min(y(\mathbf{X})) - y(\mathbf{x}_q^{new})]^+\}$$
$$= \left[\min(y(\mathbf{X})) - \min\left(y(\mathbf{x}_1^{new}), ..., y(\mathbf{x}_q^{new})\right)\right]^+$$

This leads to the q-points expected improvement:

The q-$\mathbb{E}$I criterion

$$\mathbb{E}I(\mathbf{x}_1^{new}, ..., \mathbf{x}_q^{new}) := \mathbb{E}\left[\left(\min(y(\mathbf{X})) - \min(Y(\mathbf{x}_1^{new}), ..., Y(\mathbf{x}_q^{new}))\right)^+ |Y(\mathbf{X}) = \mathbf{Y}\right]$$

Remark: The $Y(\mathbf{x}_j^{new})'s$ are dependent random variables (also $|Y(\mathbf{X}) = \mathbf{Y}$).

An introduction to the q-points $\mathbb{E}$I

Background: $(\mathbf{x}_1^{new}, ..., \mathbf{x}_q^{new})$ has to be chosen such that the improvement brought after evaluating y at the q locations be as large as possible.
By definition, we wish them to (*a posteriori*) maximize:

$$i(\mathbf{x}_1^{new}, ..., \mathbf{x}_q^{new}) := \max\{[\min(y(\mathbf{X})) - y(\mathbf{x}_1^{new})]^+, ..., [\min(y(\mathbf{X})) - y(\mathbf{x}_q^{new})]^+\}$$
$$= \left[\min(y(\mathbf{X})) - \min\left(y(\mathbf{x}_1^{new}), ..., y(\mathbf{x}_q^{new})\right)\right]^+$$

This leads to the q-points expected improvement:

The q-$\mathbb{E}$I criterion

$$\mathbb{E}I(\mathbf{x}_1^{new}, ..., \mathbf{x}_q^{new}) := \mathbb{E}\left[\left(\min(y(\mathbf{X})) - \min(Y(\mathbf{x}_1^{new}), ..., Y(\mathbf{x}_q^{new}))\right)^+ | Y(\mathbf{X}) = \mathbf{Y}\right]$$

Remark: The $Y(\mathbf{x}_j^{new})'s$ are dependent random variables (also $|Y(\mathbf{X}) = \mathbf{Y}$).

An illustration of the 2-points $\mathbb{E}$I

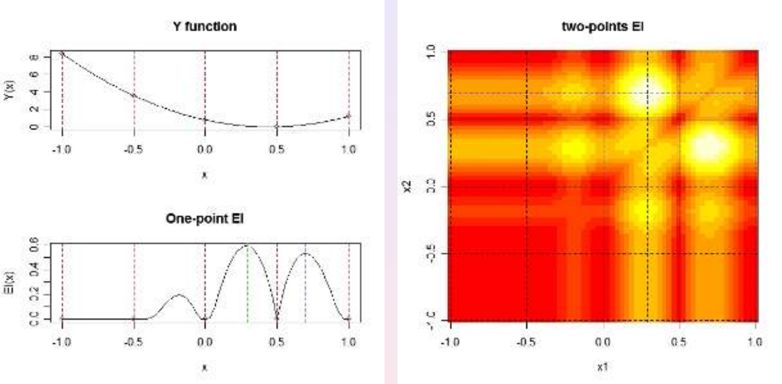


Figure: 1 and 2-$\mathbb{E}$I associated with a 2d degree polynom

The 2-$\mathbb{E}$I optimal couple is here made of 2 local maxima of the 1-$\mathbb{E}$I

An illustration of the 2-points $\mathbb{E}$I

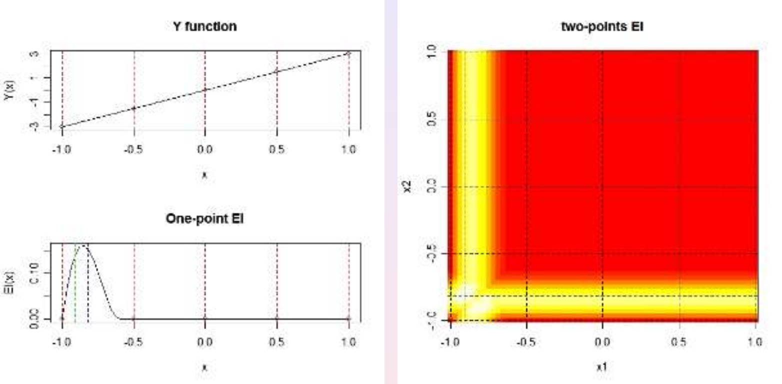


Figure: 1 and 2-$\mathbb{E}$I associated with a 1-dimensional linear function.

Now, the 2-$\mathbb{E}$I optimal couple is made of two points around the 1-$\mathbb{E}$I maximizer.

q-$\mathbb{E}$I optimal designs?

$$\mathbf{X}^{new*} = argmax_{\mathbf{x}_1^{new},\dots,\mathbf{x}_q^{new} \in S} \mathbb{E}I\left(\mathbf{x}_1^{new},\dots,\mathbf{x}_q^{new}\right)$$

This optimization is in dimension dq. Typically, $dq \geq 100$

Since the objective function is noisy (empirical $\mathbb{E}I$, computed by Monte-Carlo method) and derivative-free, the problem is not straightforward.

First proposed approach

Solve the problem in a greedy way, in feeding the Kriging model with arbitrary values (*Kriging Believer* and *Constant Liar* heuristic strategies).

One heuristic strategy for the cases where $q > 2$

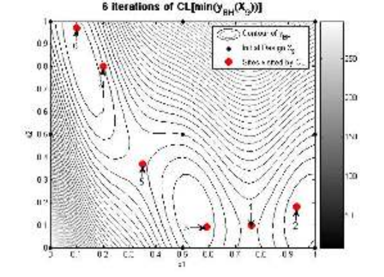
Constant Liar

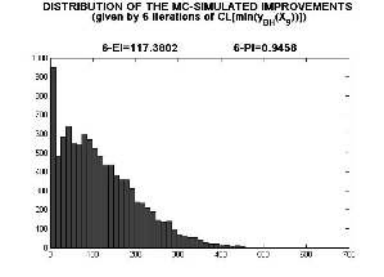
The model is sequentially updated in setting the unkwown $y(\mathbf{x}_i^{new})$ values equal to a fixed constant $L \in \mathbb{R}$:

function CL($\mathbf{X}$, $\mathbf{Y}$, L, q)
    for $i \leftarrow 1, q$ **do**
        $\mathbf{x}_i^{new} = argmax_{\mathbf{x} \in D} \mathbb{E}I(\mathbf{x})$ ▷ with updated **X** and **Y**
        $\mathbf{X} = \mathbf{X} \bigcup \{\mathbf{x}_i^{new}\}$
        $\mathbf{Y} = \mathbf{Y} \bigcup \{L\}$
    end for
end function

The constant L allows the user to control the repulsion created by the sequentially visited points ($L = max(\mathbf{Y})$ for a strong repulsion, $L = min(\mathbf{Y})$ for a smooth repulsion)

Using q-$\mathbb{E}$I to monitor heuristic strategies





Left: Branin-Hoo function with DOE $\mathbf{X}_9$ (small black points) and 6 first points given by the strategy CL[min($f_{BH}(\mathbf{X}_9)$)] (large bullets).

Right: Histogram of 10^3 Monte Carlo simulated improvements brought by the 6-points CL[min($f_{BH}(\mathbf{X}_9)$)] strategy. The corresponding 6-points PI and EI are given above.

A 6-dimensional case study: the "Hartmann" function

$$y(x) = -\sum_{j=1}^{4} c_j \times \exp\left(-\sum_{i=1}^{6} a_{i,j} \times (x_i - p_{i,j})^2\right)$$

$$a = \begin{pmatrix} 10.00 & 0.05 & 3.00 & 17.00 \\ 3.00 & 10.00 & 3.50 & 8.00 \\ 17.00 & 17.00 & 1.70 & 0.05 \\ 3.50 & 0.10 & 10.00 & 10.00 \\ 1.70 & 8.00 & 17.00 & 0.10 \\ 8.00 & 14.00 & 8.00 & 14.00 \end{pmatrix} \quad p = \begin{pmatrix} 0.1312 & 0.2329 & 0.2348 & 0.4047 \\ 0.1696 & 0.4135 & 0.1451 & 0.8828 \\ 0.5569 & 0.8307 & 0.3522 & 0.8732 \\ 0.0124 & 0.3736 & 0.2883 & 0.5743 \\ 0.8283 & 0.1004 & 0.3047 & 0.1091 \\ 0.5886 & 0.9991 & 0.6650 & 0.0381 \end{pmatrix} \quad c = \begin{pmatrix} 1.0 \\ 1.2 \\ 3.0 \\ 3.2 \end{pmatrix}$$

global minimum $= -\mathbf{3.32}$
global minimizer = $[0.202, 0.150, 0.477, 0.275, 0.312, 0.657]$

20 iterations of EGO starting from a 50-points design

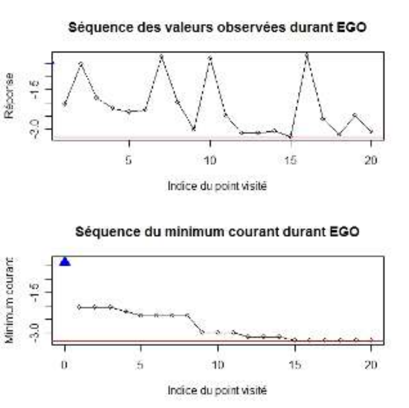


Figure: Starting from 50 points, the global optimum is reached in 15 iterations. EGO sequentially visits Hartmann's bassin of global minimum. It exploits the information given by the initial design.

90 iterations of EGO starting from a 10-points design

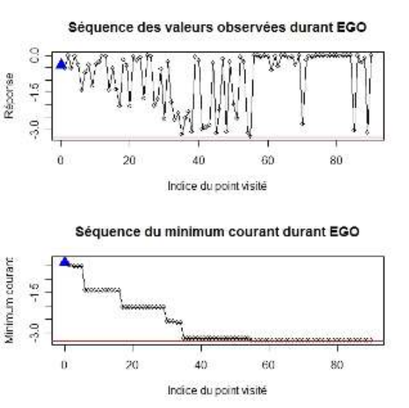


Figure: EGO find the minimum in 36 iterations: fast if we consider that **X** has only 10 points. The sequence is here more exploratory.

Optimizing by alternating CL and parallel evaluations

Idea: parallel synchronous optimization using CL at every iteration to get q explorations points. The lies of CL are corrected at the end of each iteration, after the parallel evaluations of the simulator.

Constant Liar with q points and n_{it} iterations

function CLMIN.STAGES($\mathbf{X}$, $\mathbf{Y}$, y, n_{proc}, n_{it})
  for $i \leftarrow 1, n_{it}$ **do**
    $L = min(\mathbf{Y})$
    for $j \leftarrow 1, n_{proc}$ **do**
      $\mathbf{x}_j^{new} = argmax_{\mathbf{x} \in D} \mathbb{E}I(\mathbf{x})$ ▷ with $\mathbf{X}$ and $\mathbf{Y}_{CL}$
      $\mathbf{X} = \mathbf{X} \bigcup \{\mathbf{x}_j^{new}\}$
      $\mathbf{Y}_{CL} = \mathbf{Y} \bigcup \{L\}$
    end for
    $\mathbf{Y} = \mathbf{Y} \bigcup y(\{\mathbf{x}_1^{new}, \ldots, \mathbf{x}_{n_{proc}}^{new}\})$ ▷ Parallel simulator evaluations
    Re-estimation of the kriging model
  end for
end function

CL with 10 proc. in parallel, 50 initial points

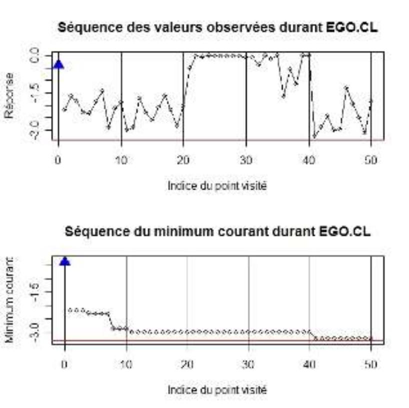


Figure: The minimum is reached in 5 time units!

CL with 10 proc. in parallel, 50 initial points

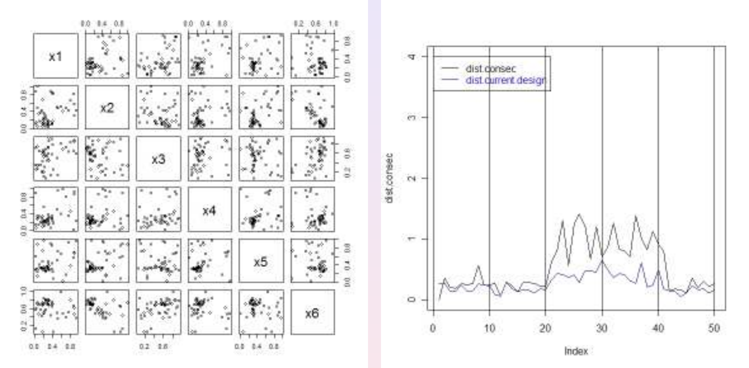


Figure: The algorithm alternates between a first exploration phase (two first time units), a more exploratory phase (3*rd* et 4*th* time units), and a final exploitation phase during which it finds the minimum.

CL with 10 proc. in parallel, 10 initial points

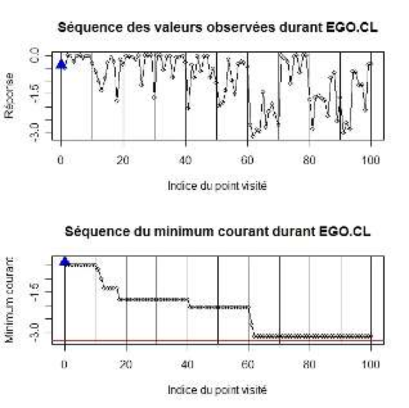


Figure: Starting from a 10-points design, CL_{min} with 10 processors finds the minimum in 7 time units.

CL with 10 proc. in parallel, 10 initial points

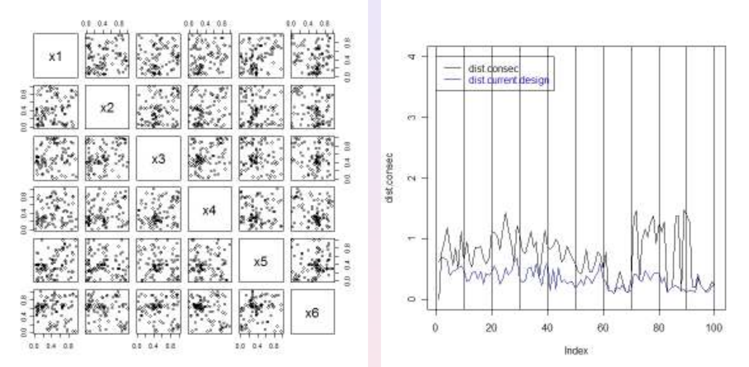


Figure: The algorithm starts by exploring and condamn less promizing zones, then finds the optimal zone and visits it until convergence. It finally leaves again to explorate further...

Conclusion and perspectives for the $q - \mathbb{E}I$

Experimental feed-back

- Computing kriging-based multipoints criteria by MC is affordable whatever the dimension of the space of inputs
- The Constant Liar heuristic strategy gave very promizing results on 1-, 2-, and 6-dimensional toy examples

Tracks for future works

- Apply these tools to real world problems (currently done with a nuclear safety application)
- Optimize the q-$\mathbb{E}$I (mutating good candidate designs ?)

Outline

Aim of this work

Industrial problem:

Let us assume that a set of physical symmetries leave y unchanged

...how can we take it into account within GP techniques?

Mathematical formulation:

if G is a finite groupe acting on D via

$$\Phi : (\mathbf{x}, g) \in D \times G \longrightarrow \Phi(\mathbf{x}, g) = g.\mathbf{x} \in D$$

what properties must k satisfy for $Y_{\mathbf{x}}$ to have its paths invariant by Φ?

Main property

Definition Y has its paths invariant under the action Φ if

$$\forall \omega \in \Omega,\ \forall \mathbf{x} \in D,\ \forall g \in G,\ \ Y_{\mathbf{x}}(\omega) = Y_{g.\mathbf{x}}(\omega)$$

Theorem: Let Y be a centered process (not necessarily gaussian!!!)

Y has invariant paths under Φ (up to a modification)

$$\Leftrightarrow$$

$\exists$ a d.p. kernel k_Z such that $k_Y(\mathbf{x}, \mathbf{x}') = \sum_{(g,g')\in G^2} k_Z(g.\mathbf{x}, g'.\mathbf{x}')$

Application: A smooth symmetrical 2-D GP

Idea: to build processes with paths invariant under Φ on the basis of a stationary process Y, by simply symmetrizing it:

$$\forall x \in D,\ Y_{\mathbf{x}}^{\Phi} = \frac{1}{2}(Y_{\mathbf{x}} + Y_{s(\mathbf{x})}) = \frac{1}{2}(Y_{(x_1,x_2)} + Y_{(x_2,x_1)})$$

The covariance kernel of the new process Y^{Φ} is given by:

$$\begin{aligned} k_{Y^{\Phi}}(x,x') &= \frac{1}{4}[k_Y(x-x') + k_Y(s(x)-x') + k_Y(x-s(x')) + k_Y(s(x)-s(x'))] \\ &= \frac{1}{4}\left[e^{-||(x_1-x_1',x_2-x_2')||^2} + e^{-||(x_1-x_1',x_2'-x_2)||^2} + e^{-||(x_1'-x_1,x_2-x_2')||^2} + e^{-||(x_1'-x_1,x_2'-x_2)||^2}\right] \end{aligned}$$

Note that Y^{Φ} inheritates from Y's smoothness, including on the axis of symmetry $\{\mathbf{x} \in \mathbb{R}^2 : s(\mathbf{x}) = \mathbf{x}\}$.

Simulation of a smooth symmetrical 2-D GP

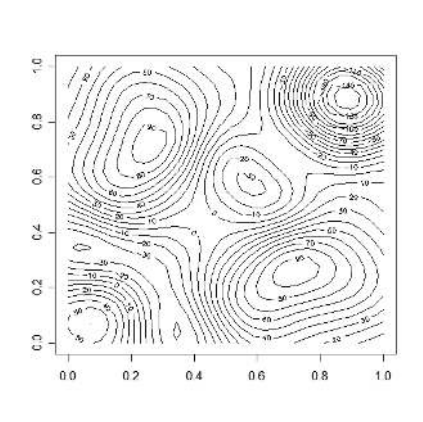


Figure: One path of GP with symmetrized Gaussian kernel

Kriging with a symmetrized kernel

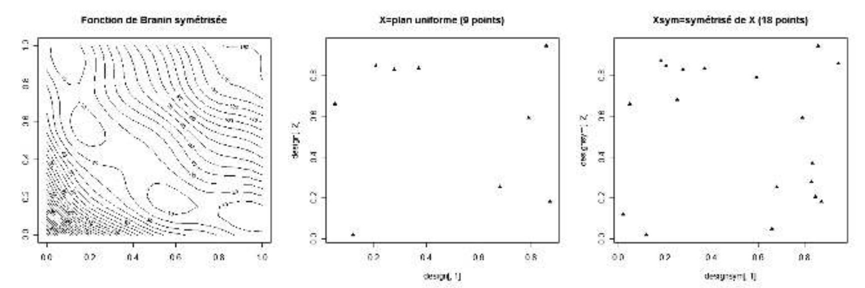


Figure: From left to right:

- symmetrized Branin function (f),
- a 9-points DOE X obtained by i.i.d. uniform drawings on the square,
- the DOE symmetrized from X with respect to f's axis of symmetry.

Kriging with a symmetrized kernel

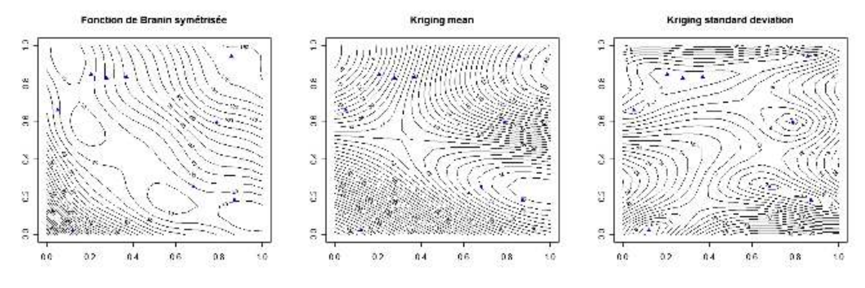


Figure: Kriging *f* on the basis of X (9 points), with Gaussian covariance

ISE on a 21 $\times$ 21-elements test grid: **820.93**

Kriging with a symmetrized kernel

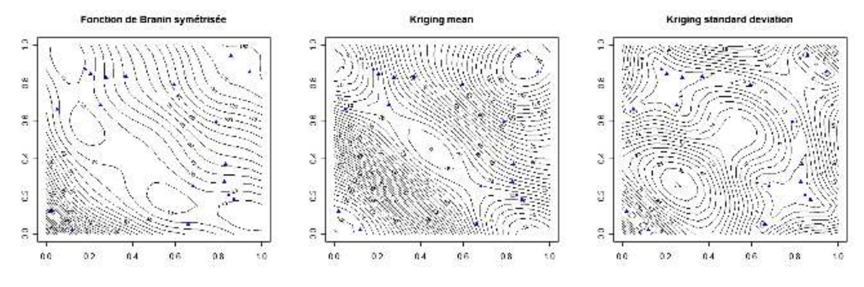


Figure: Kriging f on the basis of Xsym (18 points), with Gaussian covariance

ISE on a 21×21-elements test grid: **694.11**

Kriging with a symmetrized kernel

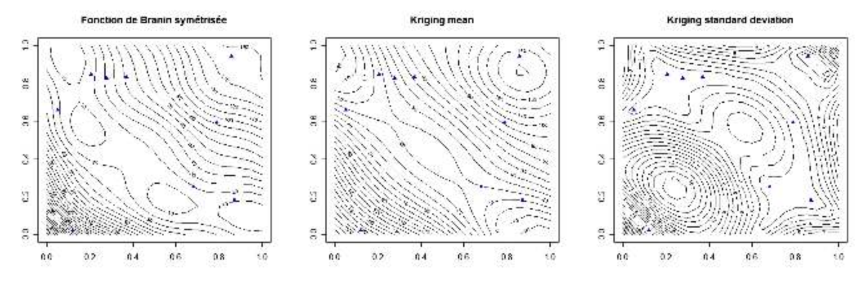


Figure: Kriging on the basis of X, with symmetrized Gaussian covariance

ISE on a 21 $\times$ 21-elements test grid: **330.26**

Conclusion and perspectives

It is not reasonable to make predictions using classical covariance kernels when invariances under some group actions are known *a priori*.

Symmetrizing stationary kernels provides a convenient way of getting invariant GPs with nice smoothness properties.

Future issues to be addressed include

- investigating broader classes of invariant kernels
- applying symmetrical Kriging to higher-dimensional industrial cases
- learning symmetries from data

Thank you for your Attention :)

Aknowledgments

- L. Carraro (Telecom Saint-Etienne), A. Antoniadis (UJF), R. Le Riche (CNRS), O. Roustant (Mines), R. Haftka (Florida), Y. Richet (IRSN), A. Journel (Stanford), V. Picheny (Mines / Florida), P. Renard (UNINE).
- This work was conducted within the frame of the DICE (Deep Inside Computer Experiments) Consortium between ARMINES, Renault, EDF, IRSN, ONERA, and Total S.A.

Any questions?