



**HAL**  
open science

## Surrogates and (mono-objective) optimization: a long-term relationship

Rodolphe Le Riche

► **To cite this version:**

Rodolphe Le Riche. Surrogates and (mono-objective) optimization: a long-term relationship. DEA. France. 2016. cel-01281650

**HAL Id: cel-01281650**

**<https://hal.science/cel-01281650>**

Submitted on 2 Mar 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Surrogates and (mono-objective) optimization: a long-term relationship

Rodolphe Le Riche

CNRS LIMOS @ Ecole des Mines de St-Etienne

SAMCO workshop, Lorentz Center  
Leiden, 29 Feb.- 4 Mar. 2016

---

# Scope of the presentation

---

This talk is about : surrogates AND mono-objective optimization  
but not about : surrogates, optimization without surrogates

Pre-requisite : basics of optimization algorithms & surrogate modeling

Language elements :

- surrogates = metamodels = response surfaces = function approximation = proxy = emulator = specific surrogate name
- specific surrogate names : polynomial (e.g., quadratic ...)  
response surface, Gaussian process or kriging, artificial neural networks, radial basis functions, splines, support vector machines, high-dimensional model reduction (HDMR), generalized additive model (GAM), ...

Surrogates and (mono-objective) optimization have had a long-term relationship because most optimization methods can be seen as having a surrogate inside.

---

# Mono-objective optimization problem formulation

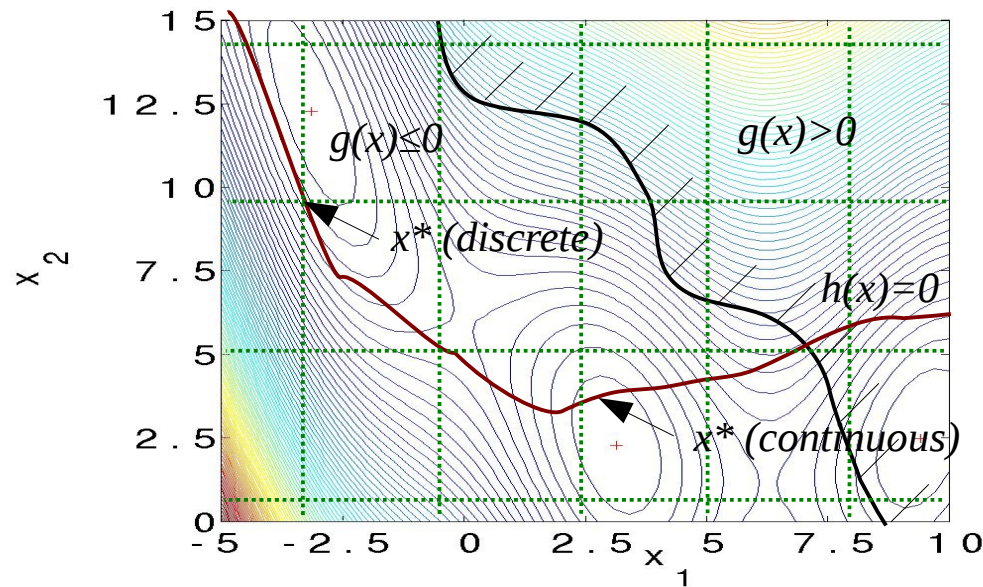
---

$$\min_{x \in S} f(x)$$

$x$  ,  $n$  optimization variables

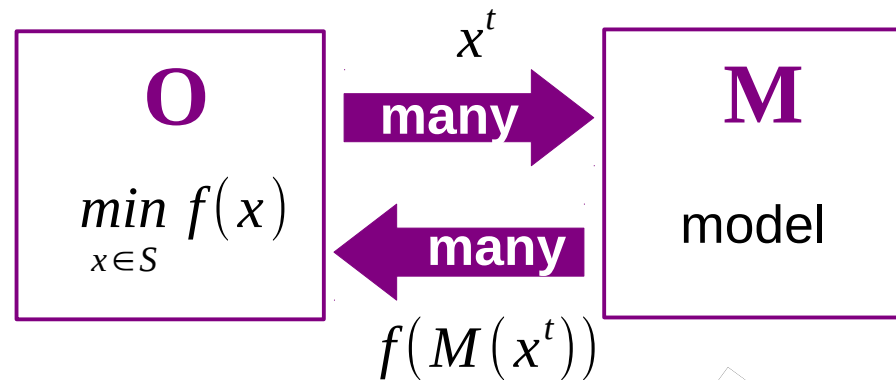
$S$  , search space ,  $x \in S$  , mainly a compact  $[x^{\text{LB}}, x^{\text{UB}}]$  in  $\mathbb{R}^n$   
but many concepts apply to  $\mathbb{I}^n$

$f$  , objective or cost function to minimize,  $f : S \rightarrow \mathbb{R}$

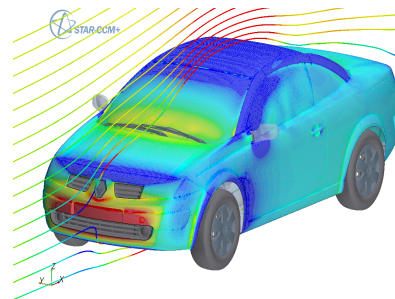


# Optimizing expensive functions

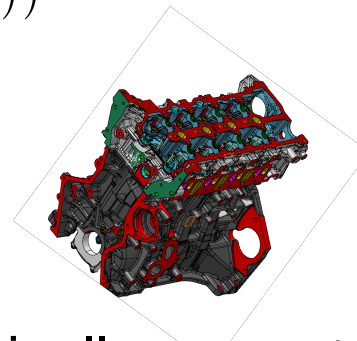
Optimization algorithms generate points  $x \in S$  in order to approximate the solution to  $\min_{x \in S} f(x)$



but  $M$  (e.g.,



,



)

is typically computationally intensive



computing sub-task

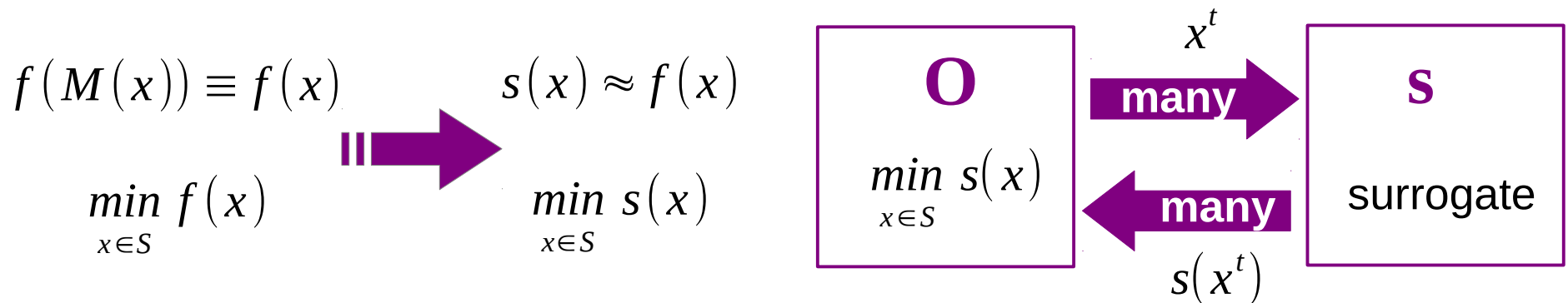


data exchange, flow  
proportional to line thickness

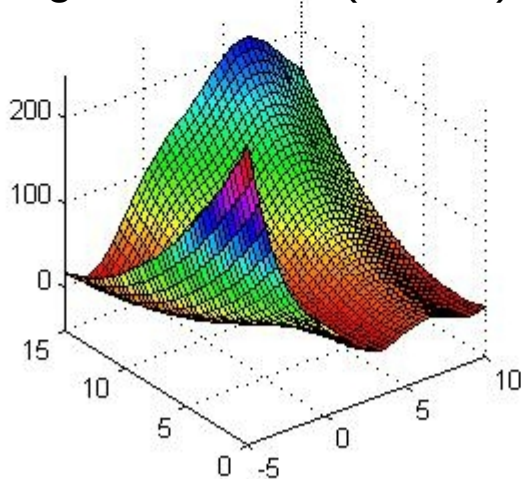
# Idea 1 : replace expensive functions by surrogates

---

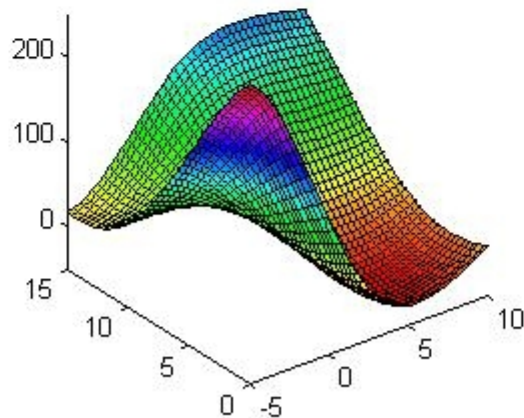
$f$  is too expensive (to do optimization, or more generally computer experiments). Let's replace it with a cheaper metamodel or surrogate = a statistical model of the physical model  $f(M(x))$ .



original function (Branin)



a surrogate (kriging)



The computing cost of solving this problem is considered negligible (compared to  $M$ ).

How to build  $s()$  ?  
To be partly discussed.

---

# Optimization has relied on surrogates for a long time

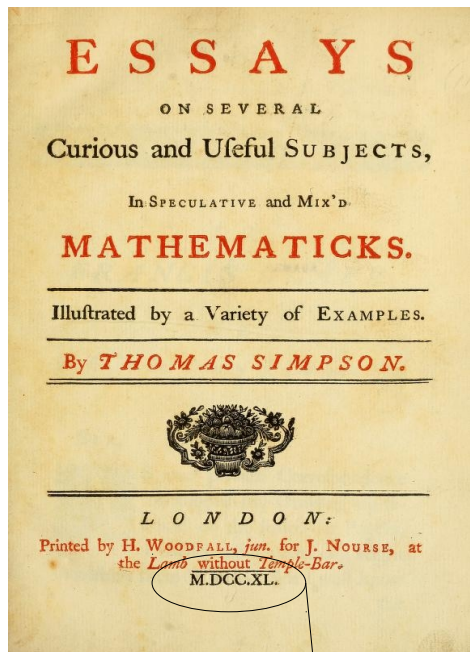
---

E.g., Newton's method

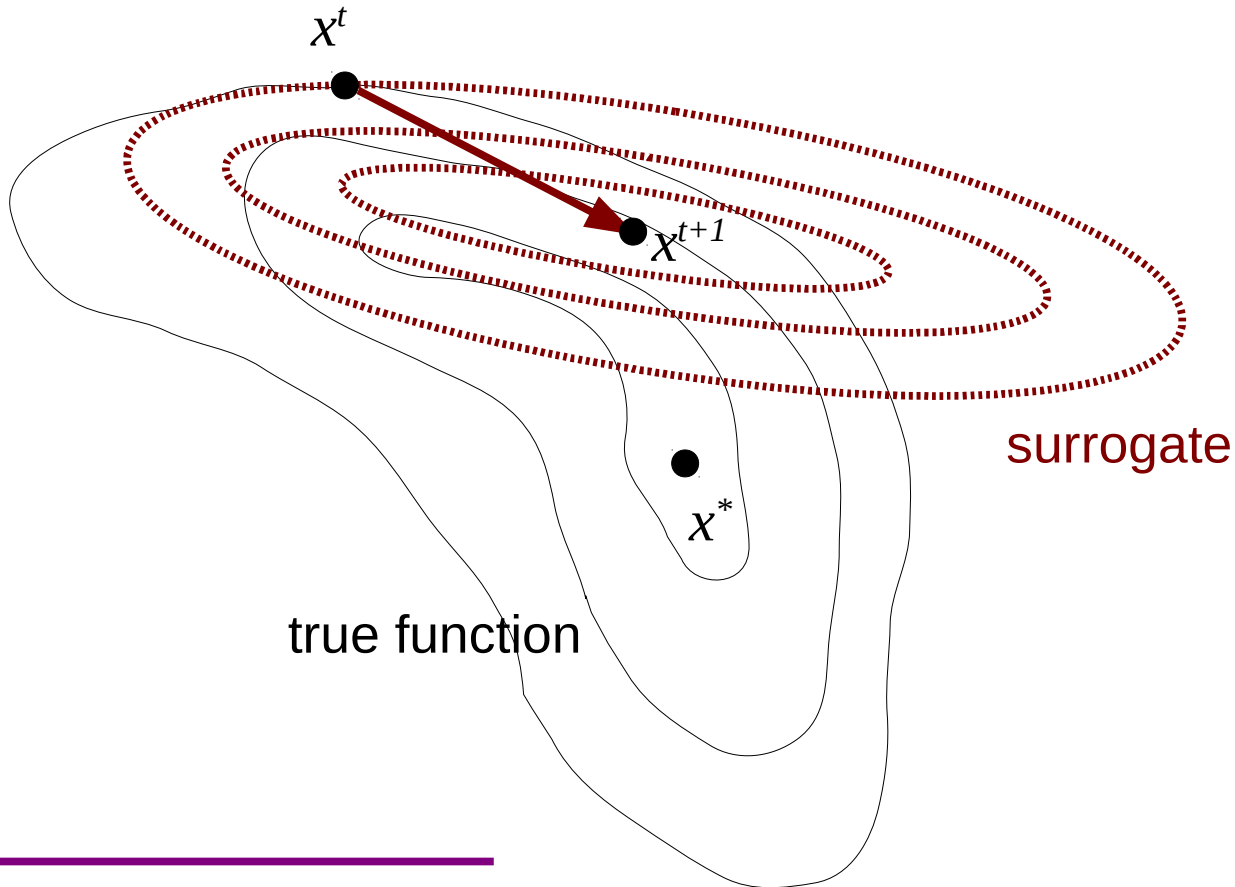
Step according to a local model (surrogate) of the function (quadratic on this example, corresponding to Newton method,

$$\nabla^2 f(x^t)(x^{t+1} - x^t) = -\nabla f(x^t)$$

May fail : cf. trust region methods



1740

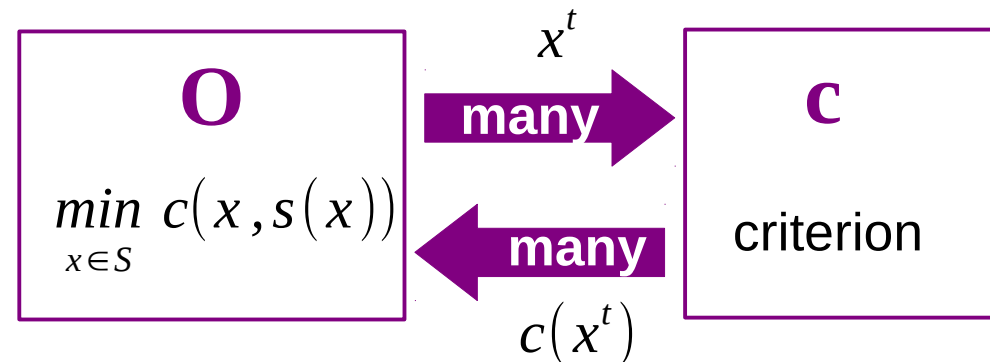


## Idea 2 : surrogate criterion

---

$f$  is too expensive (to do optimization, or more generally computer experiments). Let's replace the original problem (the function) with a problem that leads to the same solution. It includes idea 1.

$$f(M(x)) \equiv f(x) \quad \Rightarrow \quad c(x) \text{ not necessarily } \approx f(x)$$
$$\min_{x \in S} f(x) \quad \Rightarrow \quad \min_{x \in S} c(x, s(x))$$



ok as long as the best of the iterates  $x^t$  leads to  $\arg \min_{x \in S} f(x)$

The cost of calculating  $c(\cdot)$  is negligible w.r.t.  $f(\cdot)$ .

$c(\cdot)$  often based on  $s(\cdot)$

To be discussed : how to build  $c(\cdot)$  ?

---



# A first (naive) algorithm

---

- ▶ Use a quadratic polynomial surrogate

$$\begin{aligned} s(x; \theta) &= \theta_1 + \theta_1 x_1 + \dots + \theta_{n+1} x_n + \theta_{n+2} x_1 x_2 + \dots + \theta_{(n+1)(n+2)/2} x_n^2 \\ &= \sum_{i=1}^{(n+1)(n+2)/2} \Phi_i(x) \theta_i = \Phi(x) \theta \quad (\text{linear in } \theta) \end{aligned}$$

$$\Phi(x) = [1, x_1, \dots, x_n, x_1 x_2, \dots, x_{n-1} x_n, x_1^2, \dots, x_n^2]$$

- ▶ Create a "design of experiments" (DoE) :

E.g.,  $t \geq (n+1)(n+2)/2$  points randomly chosen in  $S$

$$\Rightarrow X \equiv \{x^i\}, F \equiv \{f(x^i)\}, i=1, t$$

- ▶ Fit the surrogate to the DoE by minimizing its "empirical risk" (sum of squares error)

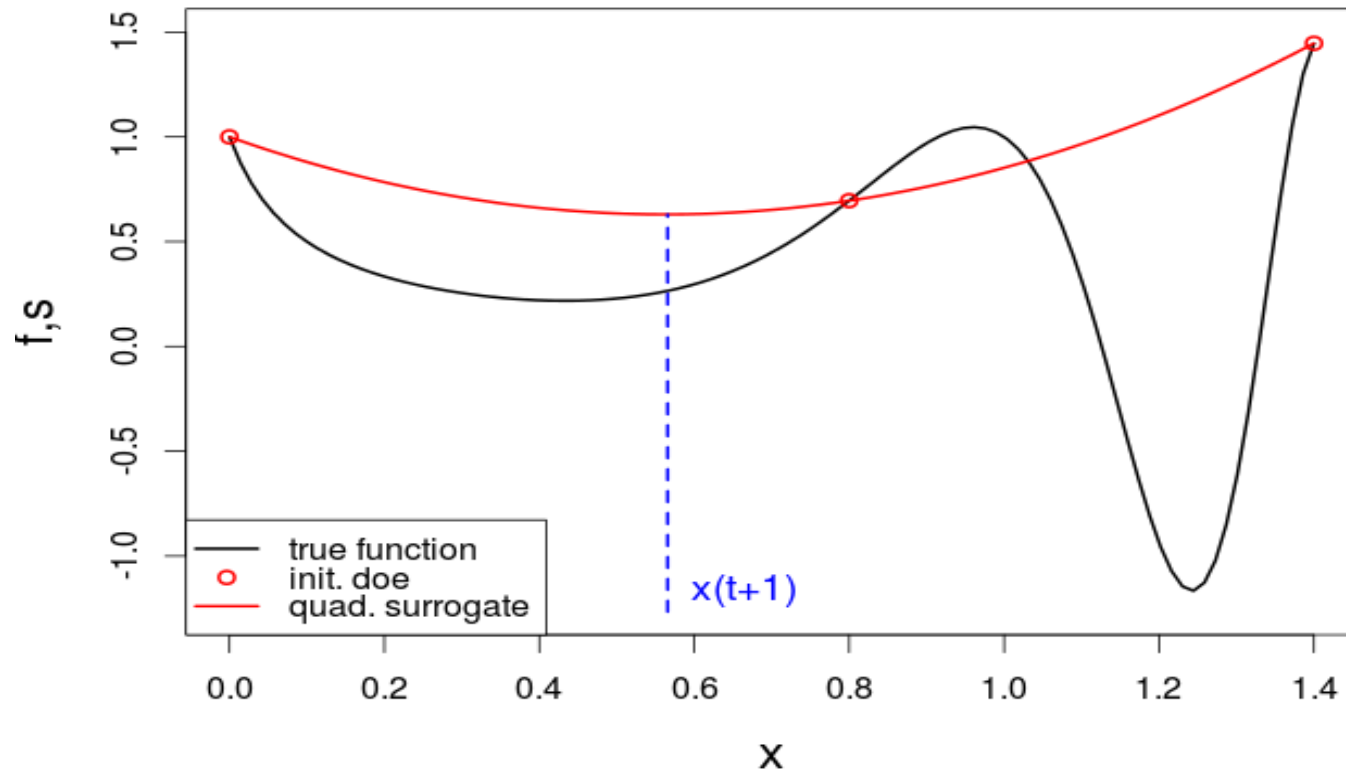
$$\theta^* = \arg \min_{\theta} \sum_{i=1}^t (f(x^i) - s(x^i; \theta))^2 \quad (\text{closed form solution exist for linear models})$$

$$\equiv \arg \min_{\theta} E(\theta, X, F)$$

- ▶ Minimize the surrogate  $x^{t+1} = \arg \min_{x \in S} s(x; \theta^*)$
-

# A first (naive) algorithm : 1D expl (1)

---



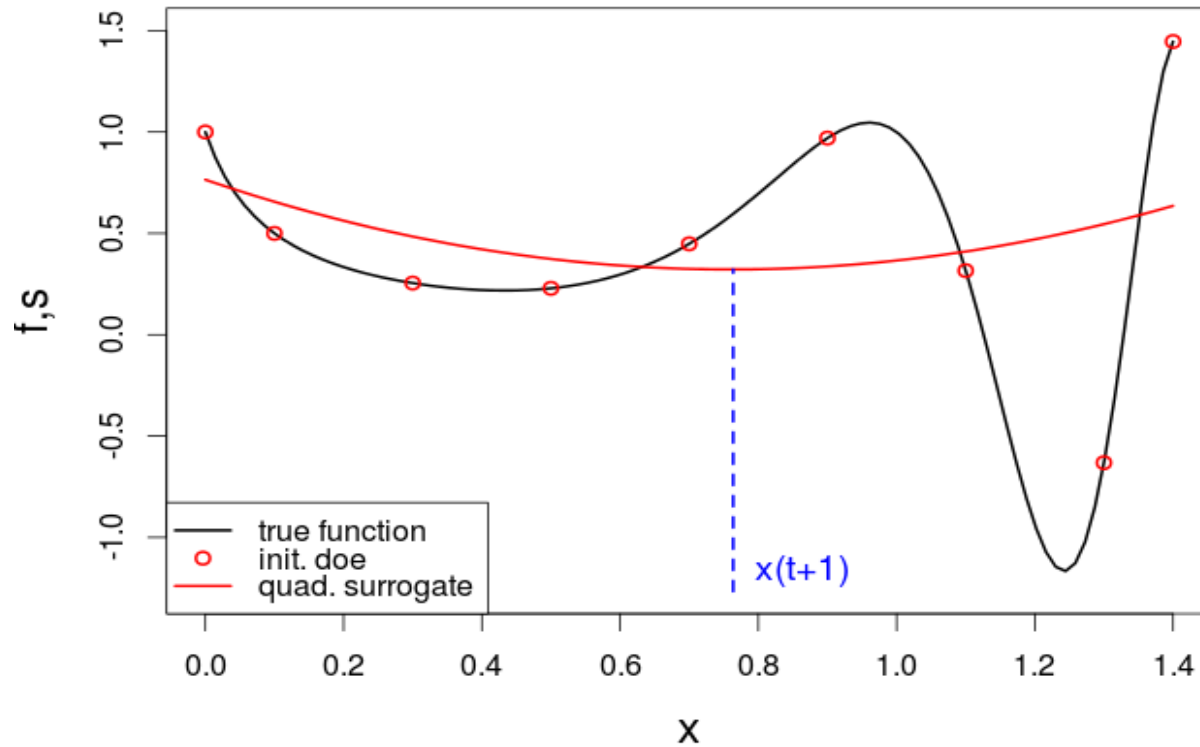
$x^{t+1}$  is not a minimizer of  $f()$

not enough points ?

---

# A first (naive) algorithm : 1D expl (2)

---



$x^{t+1}$  is still not a minimizer of  $f()$

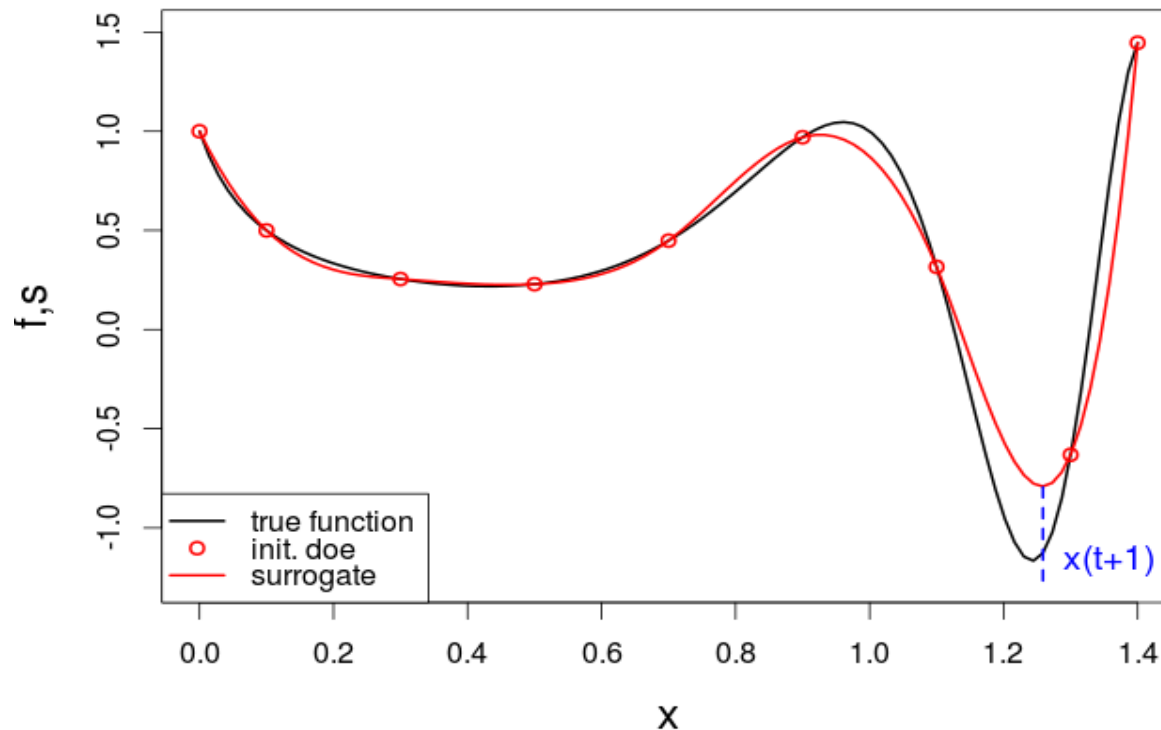
$s( ; \theta )$  too rigid (does not have the right functional form), cannot learn  $f( )$  ?

---

# A first (naive) algorithm : 1D expl (3)

---

Surrogate = cubic spline (a piece-wise 3rd degree polynomial with interpolation and smoothness properties )



Ok in 1 or 2D for a rough approximation (no convergence accuracy), but an a priori space filling DoE is very expensive.

Expl : a grid has a geometrically growing number of evaluations,  $step^n$ .

→ **Need a more greedy strategy, putting new evaluation points in the good regions of  $S$  (where  $f$  is low).**

---

# A second (naive) algorithm

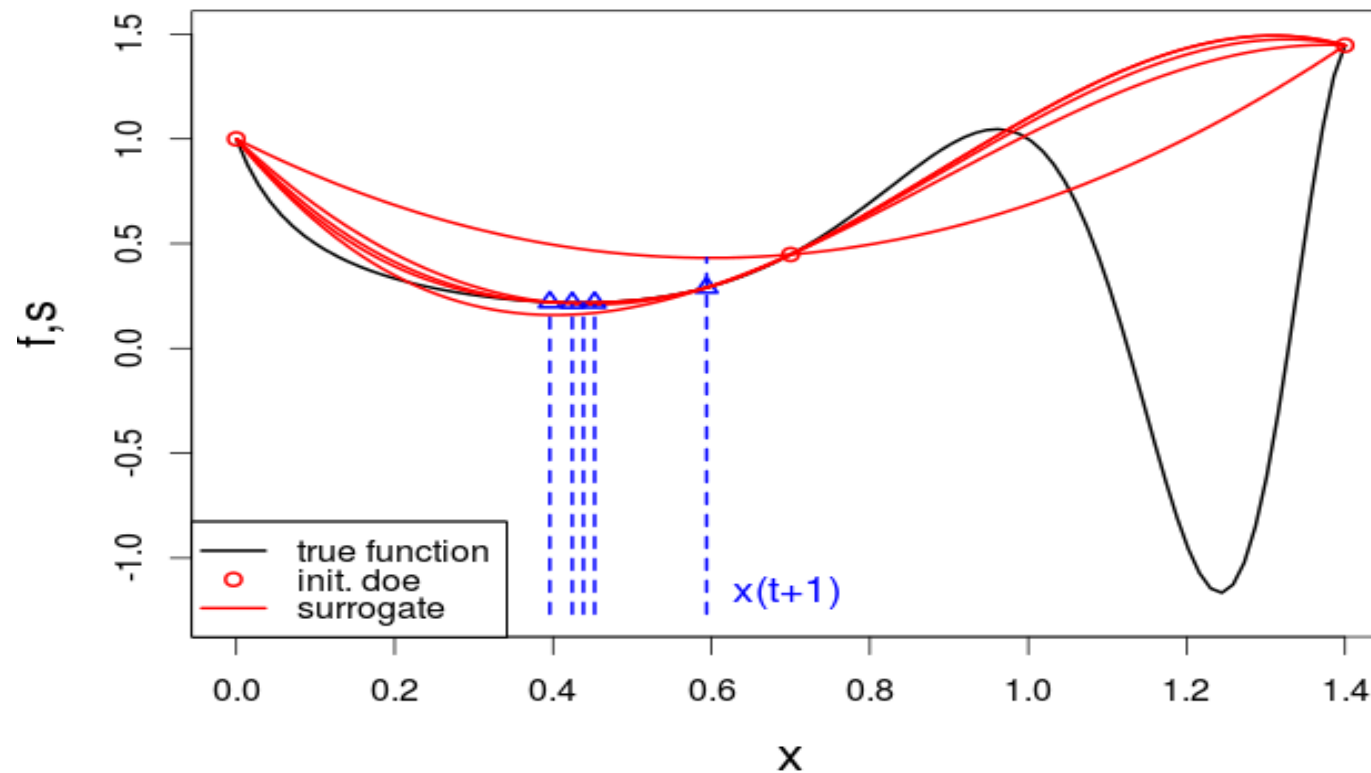
---

- ▶ Use a flexible surrogate (interpolating, or neural net with universal approximation property)
  - ▶ Create an initial DoE,  $(X, F)$ , with not too many points (at most linear in  $n$ ,  $t \approx 3n$ )
  - ▶ While (  $t < \text{budget}$  ) do
    - Fit surrogate to current DoE  $\theta^* = \arg \min_{\theta} \text{Error}(\theta, X, F)$
    - Minimize the surrogate  $x^{t+1} = \arg \min_{x \in S} s(x; \theta^*)$
    - Calculate  $f$  & update DoE  $X = \{X \cup x^{t+1}\}$  ,  $F = \{F \cup f(x^{t+1})\}$
    - $t = t+1$
  - ▶ End while
-

# A second (naive) algorithm : 1D Expl

---

( cubic spline surrogate )



Converges to a local optimum, at best ...

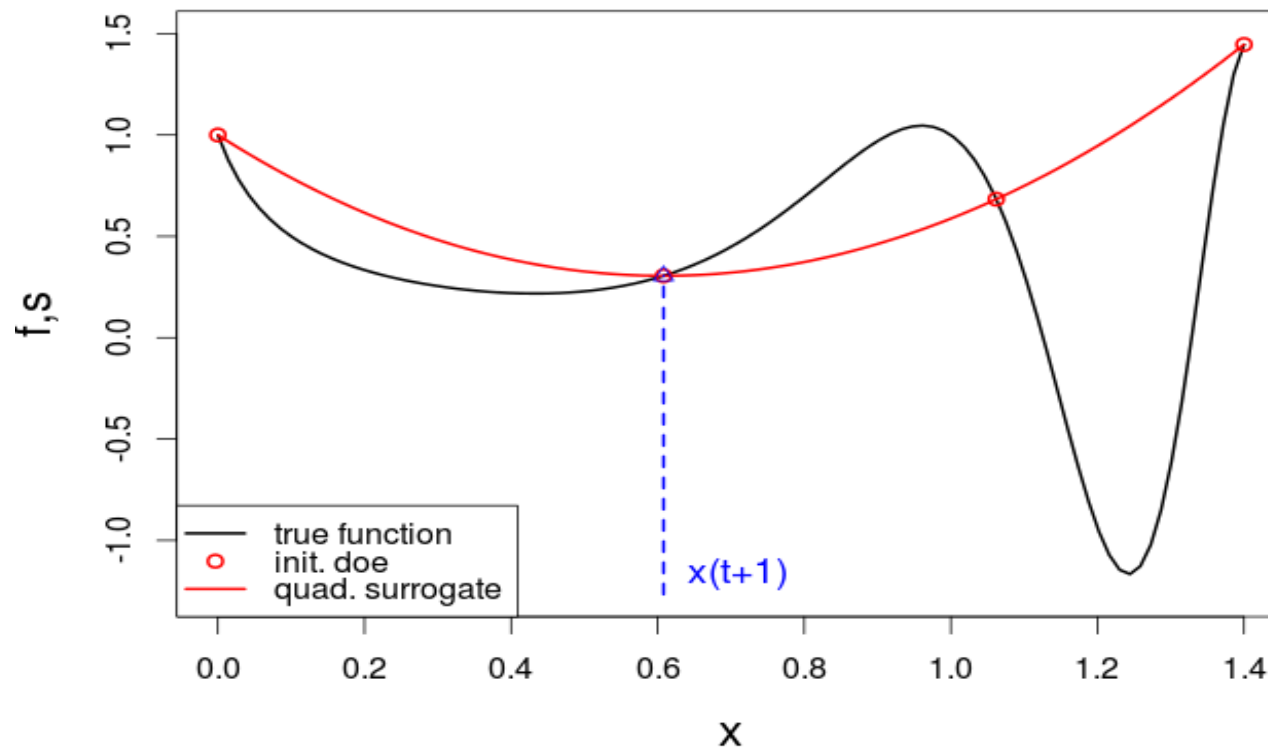
---

# A second (naive) algorithm

---

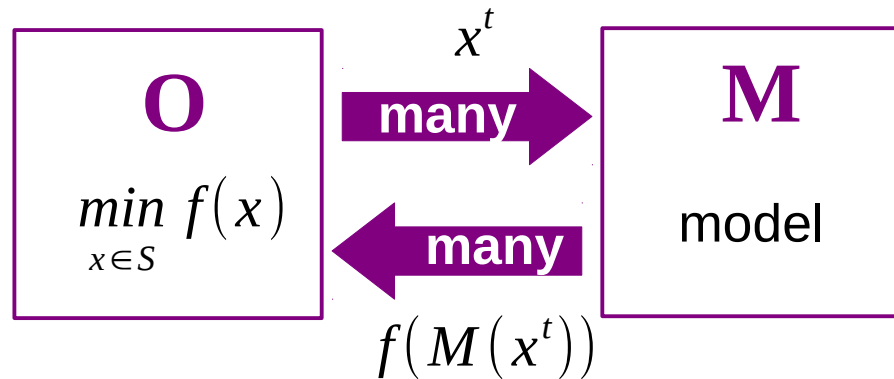
... because it can stall at non stationary points, when

$$s(x^{t+1}, \theta^*) = f(x^{t+1})$$

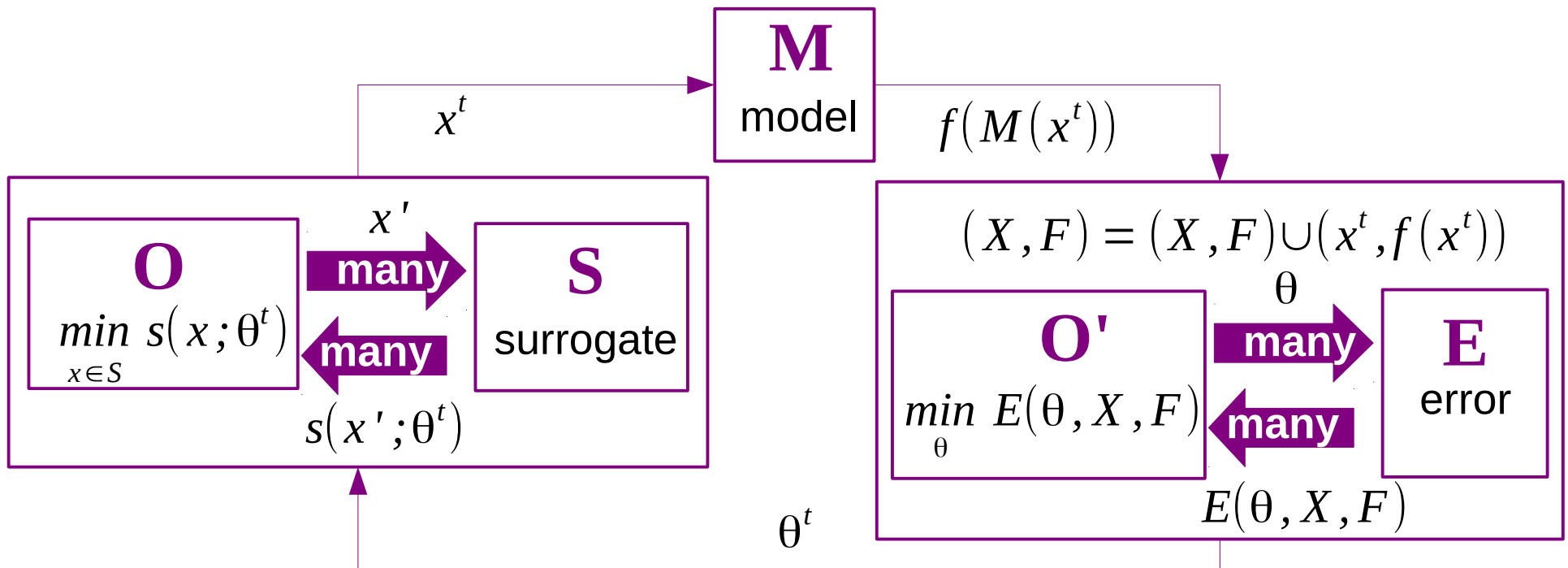


# Progress report (1)

We have replaced the costly



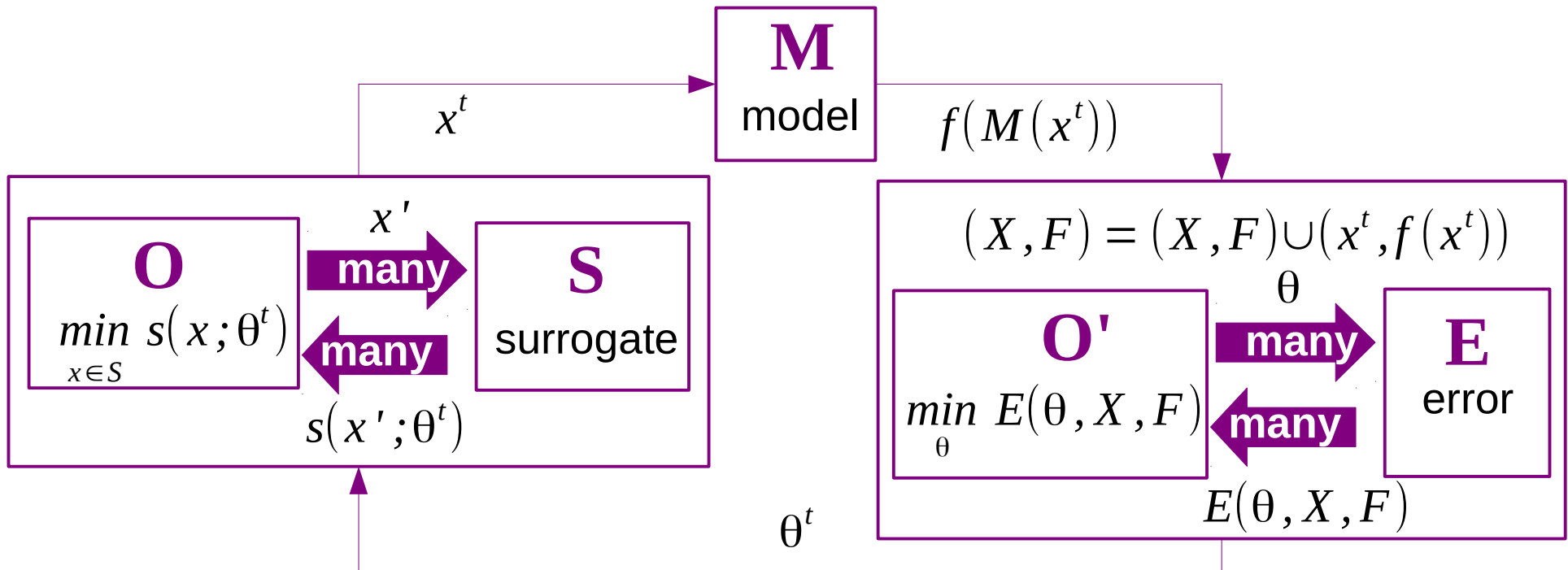
by the less costly yet not converging to stationary points





## Progress report (2)

---



**We miss a control that the surrogate S is leading the optimizer O towards better regions of the design space.**

Minimizing the surrogate is not a good enough criterion in itself to ensure that the DoE created by the iterations allows convergence to local or global optima.

---

# Outline of the talk

---

The type of strategies that ensure that the surrogate is not misleading will shape this presentation :

- Context and introduction
- ➔ • Surrogates and trust regions for local optimization
  - quadratic surrogates
  - any surrogates
- Stochastic optimization using surrogates
- Surrogates with embedded error estimates : kriging
- Ensembles of surrogates
  - unstructured
  - structured



# Quadratic surrogates and trust regions (1)

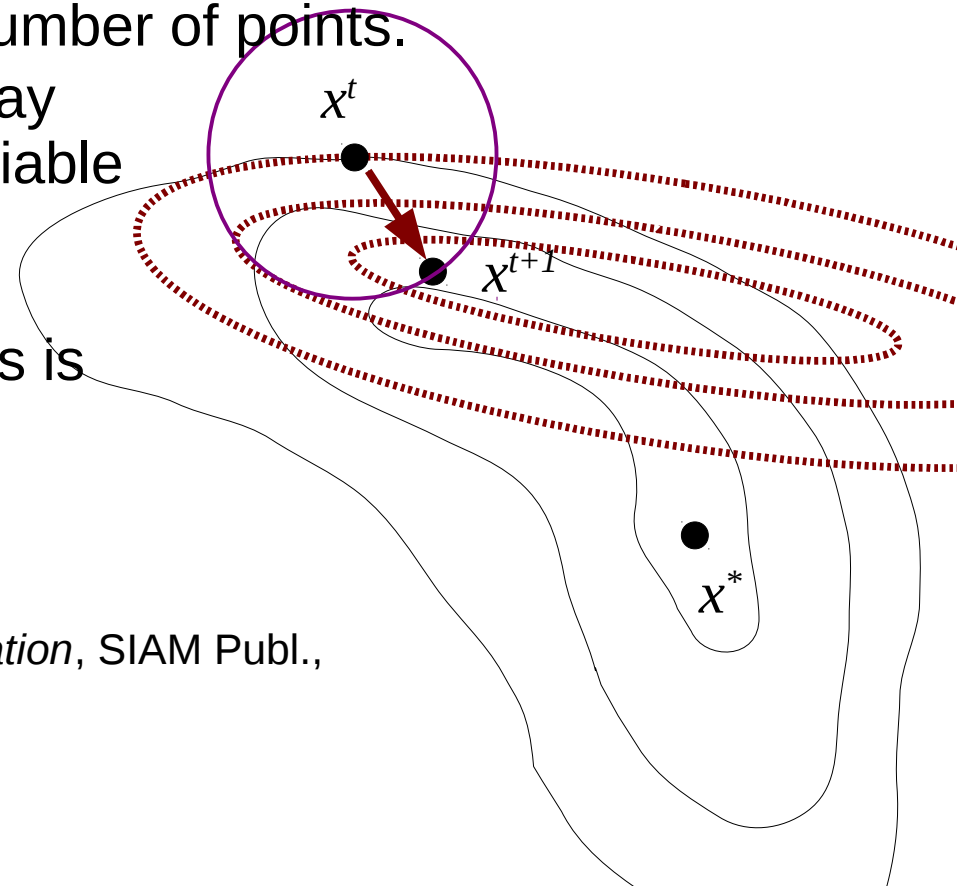
---

## Basic idea

Build a quadratic surrogate and monitor its validity in a ball around the current iterate. Define iterates by solving a minimization problem in the ball.

## Motivations

- In high dimensions (say  $> 100$ ), it may not be possible to learn flexible surrogates because of the needed number of points.
- Quadratic surrogates are rigid but may always approximate a twice differentiable function in a neighborhood (order 2 Taylor).
- The minimum of quadratic surrogates is analytically tractable.



Conn et al., *Introduction to derivative free optimization*, SIAM Publ., 2009.

---

## Quadratic surrogates and trust regions (2)

---

- ▶ Create an initial DoE  $(X, F)$  of  $m$  points,  $n+2 \leq m \leq (n+1)(n+2)/2 \dots$
- ▶ While ( not stop ) do

- Fit quadratic surrogate  $s()$  to current DoE

$$\theta^t = \arg \min_{\theta} \|\nabla_x^2 s(x; \theta) - \nabla_x^2 s(x; \theta^{t-1})\|_F \quad \text{s.t.} \quad s(x^i; \theta) = f(x^i) \quad , \quad i=1, m$$

- Minimize the surrogate within the trust region

$$x' = \arg \min_{x \in S} s(x; \theta^t) \quad \text{such that} \quad \|x - x^t\| \leq \Delta_t$$

- Calculate  $f$  & check validity of surrogate

$$\rho = (f(x^t) - f(x')) / (s(x^t; \theta^t) - s(x'; \theta^t))$$

- Update trust region radius, current iterate and DoE

$$\text{If } (\rho \geq \mu > 0) \{ \uparrow \Delta_t, x^{t+1} = x' \} \quad \text{else} \{ \downarrow \Delta_t, x^{t+1} = x^t \}$$

Add  $x'$  and remove a point from  $(X, F)$  depending on  
dist. to  $x^{t+1}$  and identifiability of  $\theta$  ,  $t=t+1$

- ▶ End while

\* M.J.D. Powell, The BOBYQA algorithm for bound constrained optimization without derivatives, TR Cambridge, 2009

# Quadratic surrogates and trust regions (3)

---

Surrogate usefulness controlled through trust region

The regularization scheme (minimization of Hessian distances) makes  $(n+1)(n+2)/2$  points to determine the parameters of the quadratic surrogate not necessary and allows an  $O(n)$  optimization cost.

Identifiability of  $\theta$  : conditioning of the linear system that comes from 1st order optimality conditions of the surrogate fitting sub-problem.

Because the surrogate is local, trust region methods are local optimization methods.

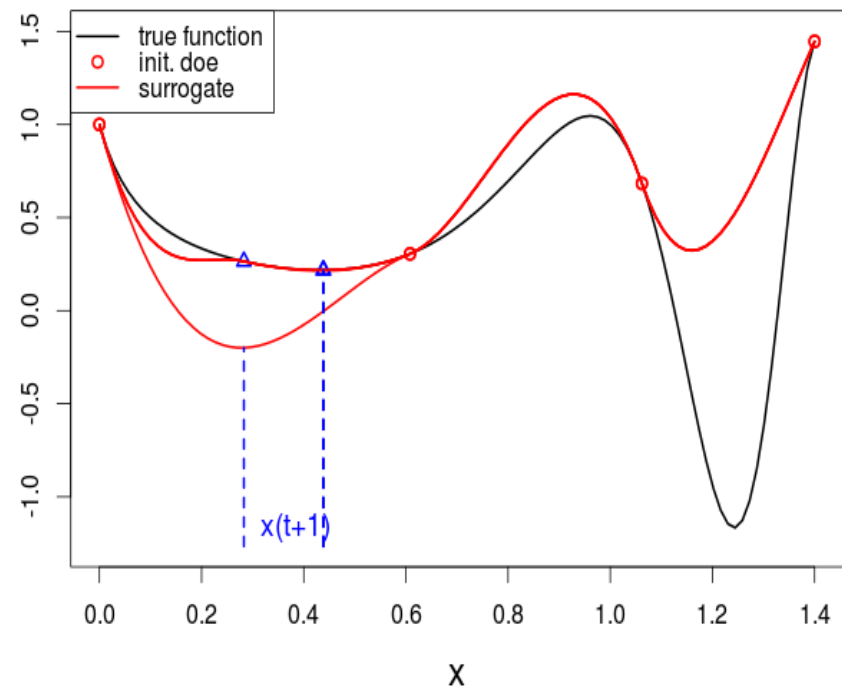
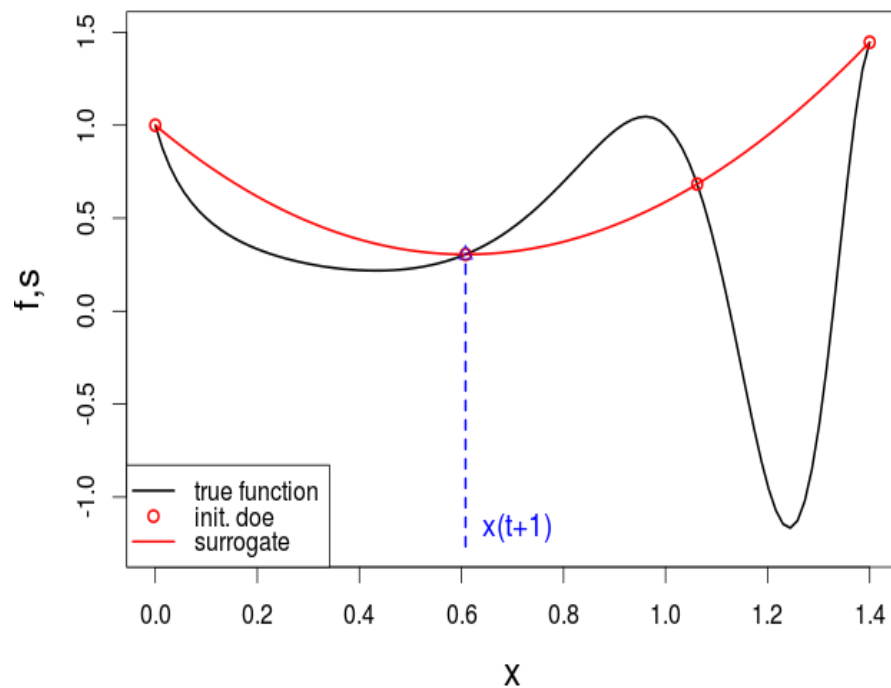
The BOBYQA algorithm is a state-of-the-art derivative free method for bound constrained minimization. It has been tested up to dimension  $n=320$ .

---

# General surrogates and trust regions

---

Optimization with any surrogate converges to a stationary point if a trust region strategy is used and the gradient of the true function is fitted at data points.




Alexandrov et al., *A trust region framework for managing the use of approximation models in optimization*, Structural Optimization, 1998.

Giunta and Eldred, *Implementation of a trust region model management strategy in the DAKOTA optimization toolkit*, AIAA-2000-4935, 2000.

---

# Outline of the talk

---

- Context and introduction
- Surrogates and trust regions for local optimization
  - quadratic surrogates
  - any surrogates
-  • Stochastic optimization using surrogates
- Surrogates with embedded error estimates : kriging
- Ensembles of surrogates
  - unstructured
  - structured

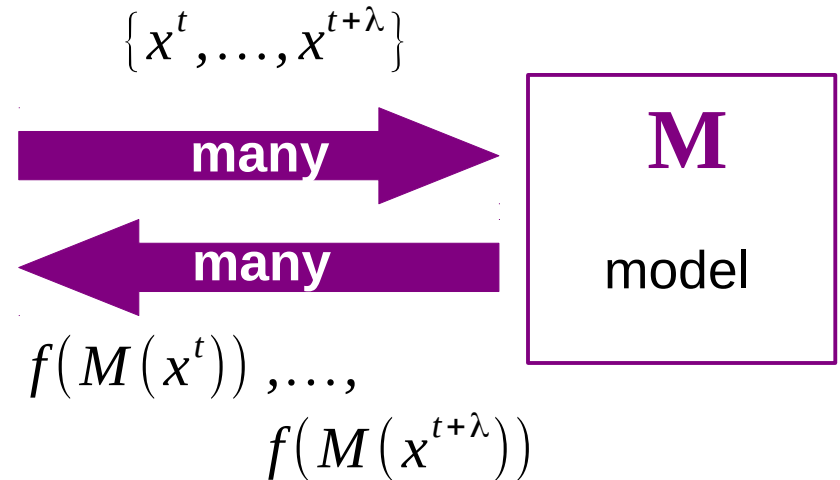


# Surrogates and stochastic optimization

---

**O**

- (rank  $x$ 's according to their  $f$ 's)
- update internal state  $\alpha^t$  of optimizer
- sample  $\lambda$  new  $x$ 's according to a pdf  $d(x; \alpha^t)$



For expl., in CMA-ES  $\alpha^t \equiv (m^t, C^t)$  where  $d(x; \alpha^t)$  is  $N(m^t, C^t)$

Two main implementations of surrogates in stochastic optimization :

- a) as a sampling filter mechanism
- b) as a generation-wise surrogate

following I.G. Loshchilov, (*Surrogate-assisted evolutionary algorithms*, PhD, 2013) but many other implementations, e.g.,

- Kern et al., 2006, local Meta-model CMA-ES
- Runarsson 2004 & 2006, approx. ranking & ordinal regression

...

---

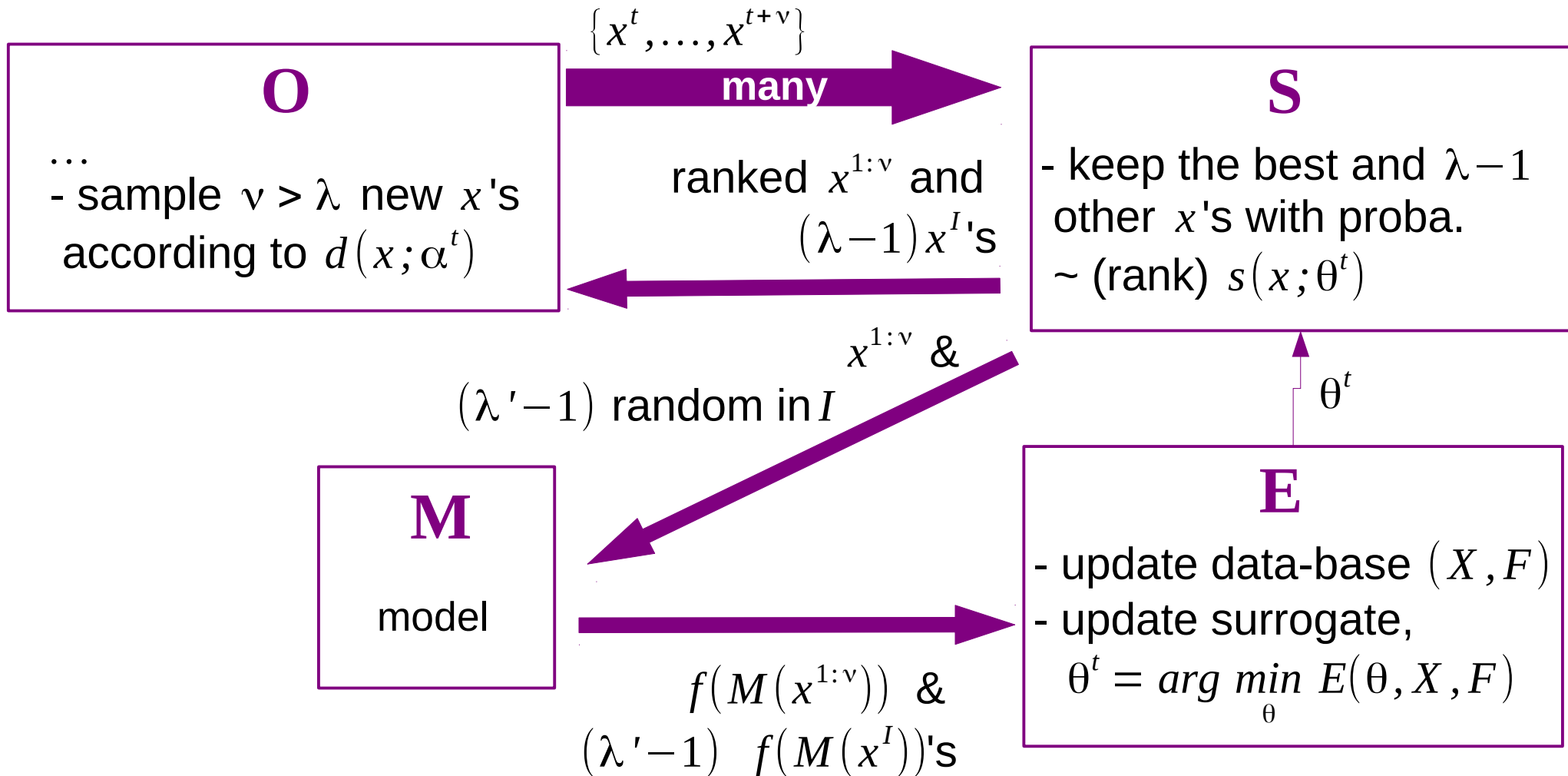


# Surrogates and stochastic optimization

## Filtered sampling (1)

simplified ACM-ES algorithm

$$v > \lambda > \lambda'$$



# Surrogates and stochastic optimization

## Filtered sampling (2)

---

### Implementation issues

**Ordinal regression** (ranking SVM, Herbrich et al., 1999) surrogates for rank based optimizers in order to preserve invariance property w.r.t. any monotonous transformation of  $f()$ .

Do not mistake the data-base and the population. In ACM-ES, the data-base is made of the  $30 \times \sqrt{n}$  to  $70 \times \sqrt{n}$  most recently evaluated points.

Having a **probabilistic choice of the  $x$ 's** for CMA population and for the data-base is necessary **to preserve points diversity**. Otherwise, think of  $\nu \rightarrow \infty$ , all the points would tend to the surrogate optimum, as strategy we have criticized.

### Performance

Speed-ups going from 2 to 4 were observed for dimensions 2 to 40 except for the (difficult) Rastrigin function <-- exploration / intensification trade-off and there is no surrogate usefulness control in ACM-ES ( $\nu$  is fixed).

Herbrich, Graepel, Obermayer, *Support Vector Learning for Ordinal Regression*, ICANN 99.

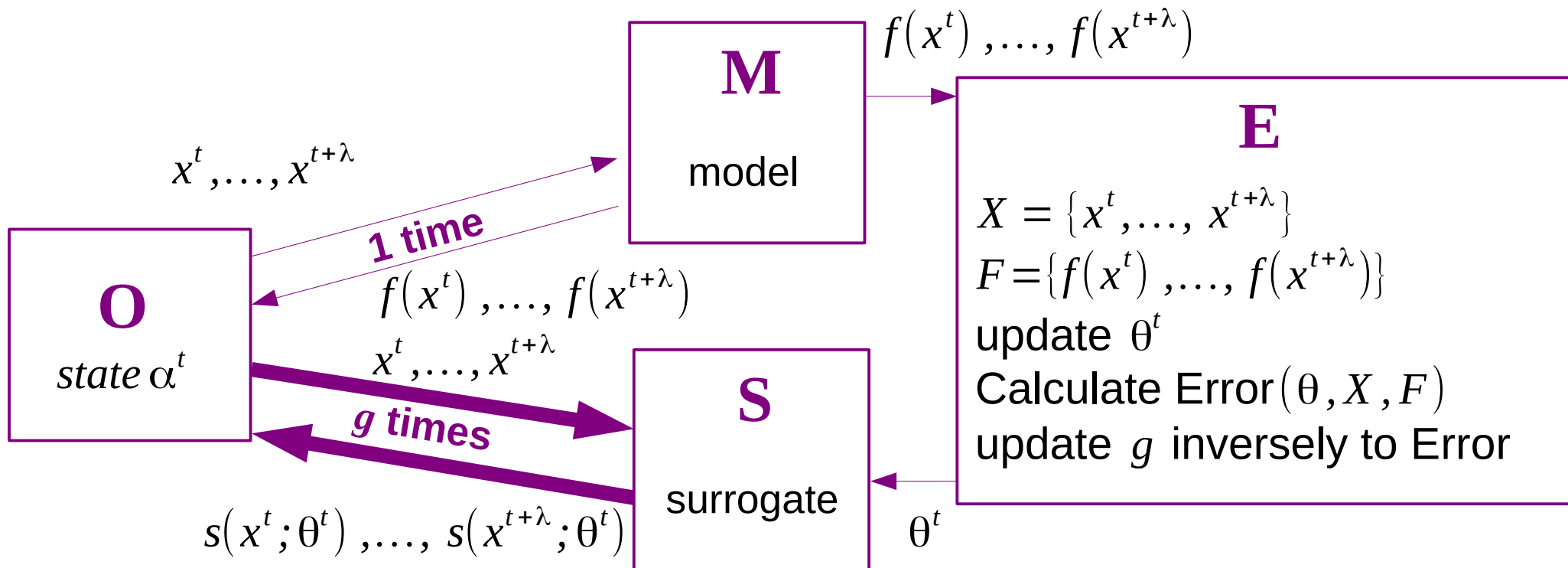
---

# Surrogates and stochastic optimization

## Generation-wise surrogate (1)

---

Principle : optimize on the surrogate for  $g$  iterations and then for 1 iteration on the true function. Adjust  $g$  according to the surrogate error.



# Surrogates and stochastic optimization

## Generation-wise surrogate (2)

---

- $g$  known as **surrogate life-length** in Y. Jin, *A comprehensive survey of fitness approximation in EC*, Soft Comp., 2005.
- Points diversity is automatically guaranteed by the use of the stochastic optimizer.
- This algorithm uses surrogate error to adjust some parameters (the surrogate life-length).

### Performance :


~ saACM-ES , Loshchilov 2012, which has speed-ups of 2 to 3 for  $n=2$  to 20 over CMA-ES (in a version where other surrogate hyper-parameters\* are optimized by minimizing surrogate error).

\* the difference between surrogate parameters and hyper-parameters is that the hyper-parameters are typically set outside of the surrogate specific functions. Expl: regularization constants  $C$  in Support Vector Machines.

---

# Outline of the talk

---

- Context and introduction
- Surrogates and trust regions for local optimization
  - quadratic surrogates
  - any surrogates
- Stochastic optimization using surrogates
-  • Surrogates with embedded error estimates : kriging
- Ensembles of surrogates
  - unstructured
  - structured



# A very short introduction to kriging (1)

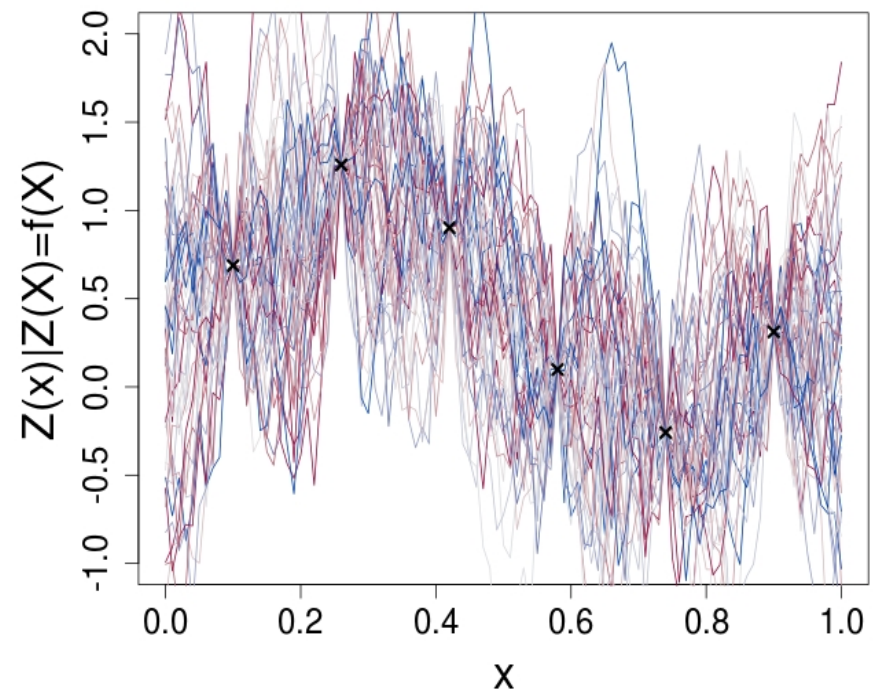
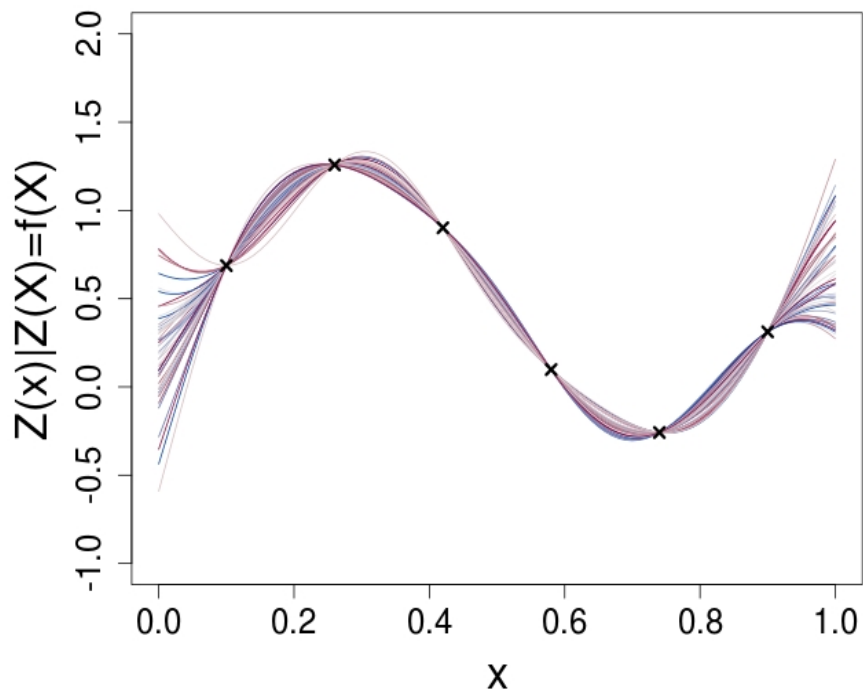
---

Kriging = conditional Gaussian processes

Random process are the function pendant to random variables

A sample = a function of  $x$

conditional = the samples are forced to go through the data points  $X, F$



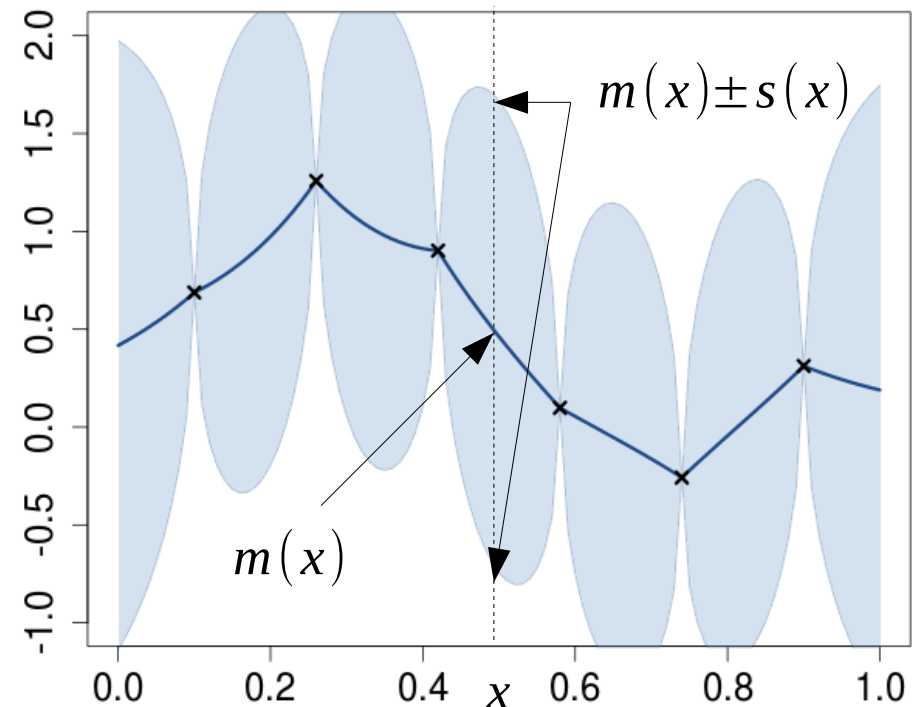
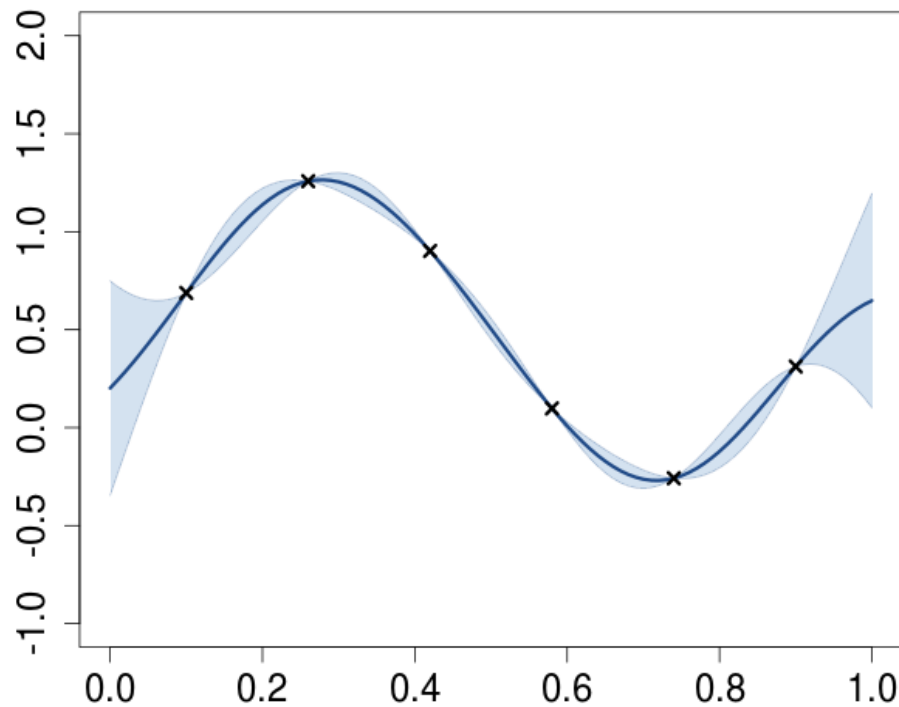
[see Rasmussen & Williams, *GPML*, 2006 for more explanations]

---

# A very short introduction to kriging (2)

---

Statistical model of  $f(x)$  :  $F(x) \sim N(m(x), s^2(x))$



(  $m(x) \pm 1.96 s(x)$  mean and 95% confidence interval )

---

# A very short Introduction to kriging (3)

---

and  $F$  is correlated in space,

$$\mathbf{c}(x) = \left[ \text{Cov}(F(x), F(x^i)) \right]_{i=1, M}$$
$$\mathbf{C} = \left[ \text{Cov}(F(x^i), F(x^j)) \right]_{i, j}$$

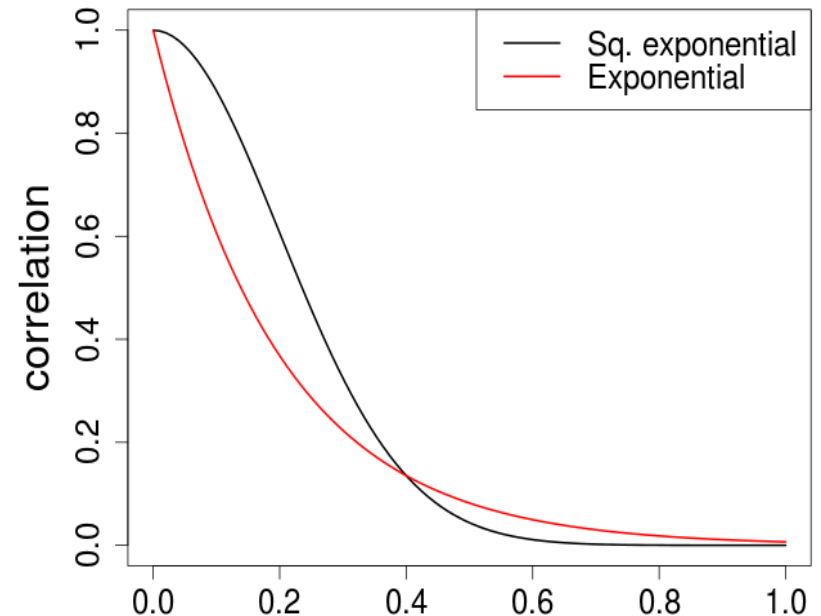
Kriging average :  $m(x) = \mu + \mathbf{c}^T(x) \mathbf{C}^{-1} (\mathbf{f} - \mu \mathbf{1})$

Kriging variance :  $s^2(x) = \sigma^2 - \mathbf{c}^T(x) \mathbf{C}^{-1} \mathbf{c}(x)$

**Assumptions** :  $f(x)$  is a sample of Gaussian process with a given parameterized (stationary) kernel

$\text{Cov}(F(x), F(x')) =$  a function of  $|x - x'|$  and parameters  $\theta$  (length scale)

( not all functions are kernel functions )



(left) squared exponential ,  $\text{Cor}(F(x), F(x')) = \exp(-1/2 * (|x - x'|/\theta)^2)$  ,  
(right) exponential ,  $\text{Cor}(F(x), F(x')) = \exp(-(|x - x'|/\theta))$  ,  $\theta=0.2$

---



# Surrogates with embedded error criteria

---

The kriging prediction variance,  $s^2(x)$ , opens the way to a large family of criteria for controlling the quality of the surrogate during optimization.

Optimizing with surrogate has

- a main goal : provide an iterate  $x^i$  with a low  $f(x^i)$
- a secondary goal : have the DoE of iterates  $(X,F)$  allow a surrogate that is accurate in high performance regions of the design space.

but since we don't know a priori where are the good regions of the design space, this amounts to an intensification / exploration compromise.

Mise en abyme (multi-crit for mono-crit) : can be seen as the two criteria problem,

$$\left\{ \begin{array}{l} \min_{x \in S} m(x) \\ \max_{x \in S} s(x) \end{array} \right. \quad \text{although the next single criteria may be more meaningful ...}$$

[cf. D.R. Jones, *A Taxonomy of Global Optimization Methods based on Response Surfaces*, JOGO, 2001]

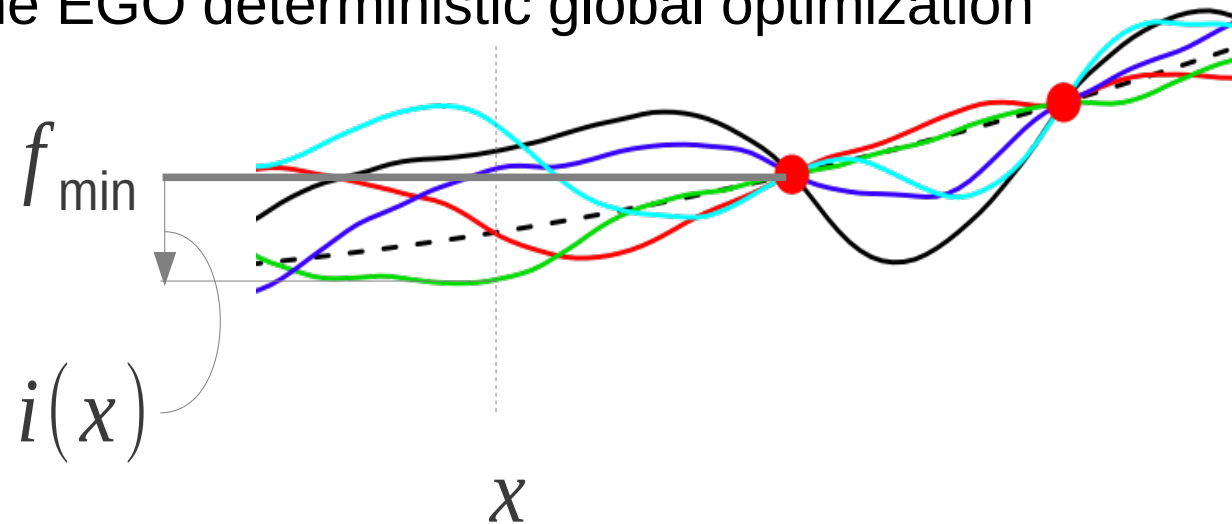
---

# Expected Improvement criterion

A natural measure of progress : the improvement,

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

- 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.



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

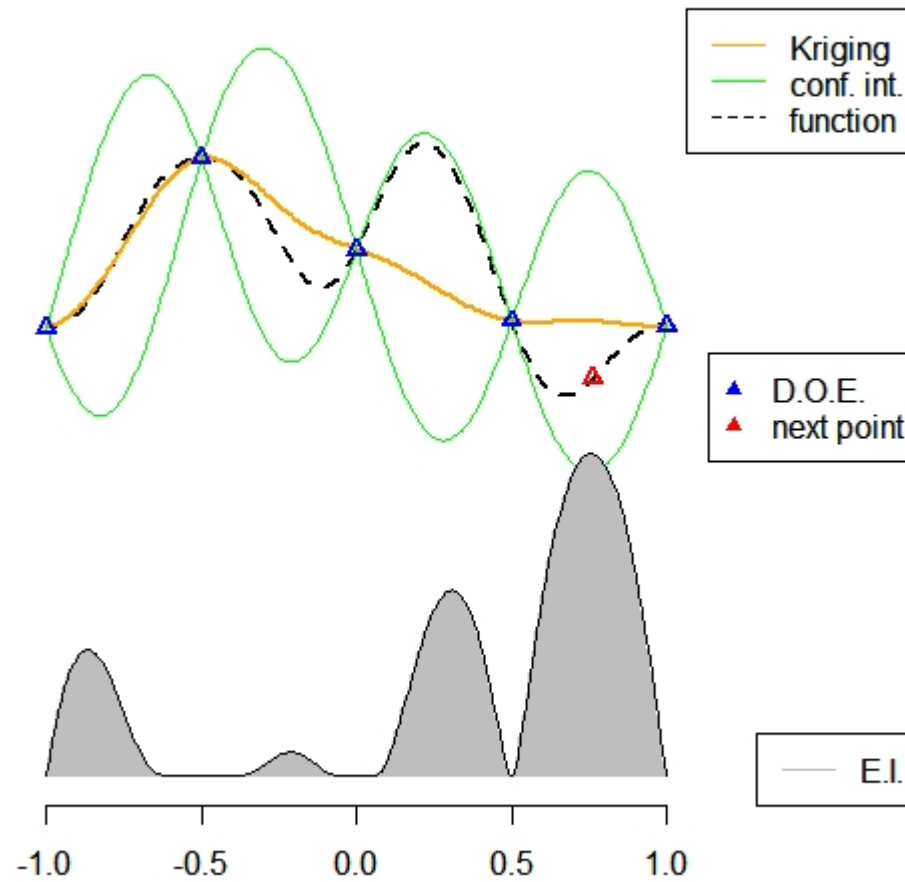
# kriging-based approaches

## EI criterion, 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 ...

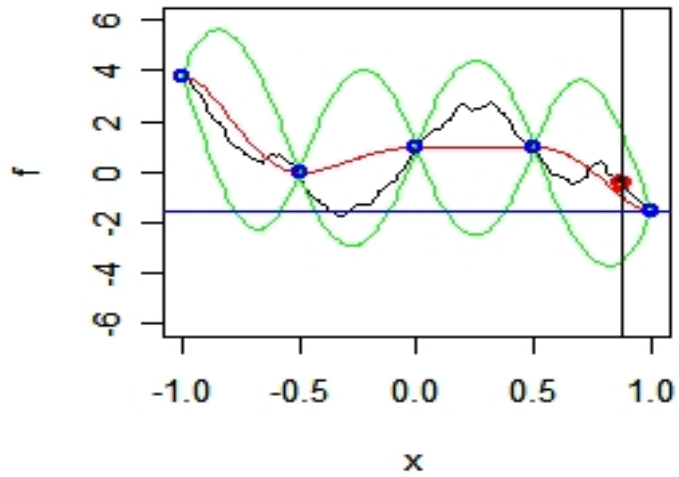
---

# kriging-based approaches

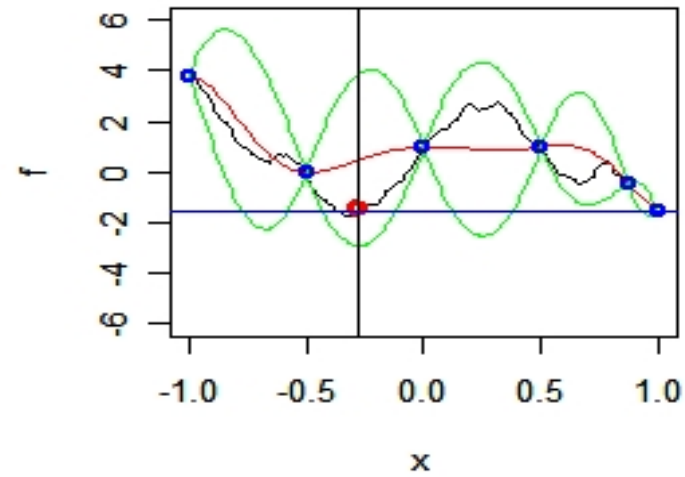
## El criterion : example

---

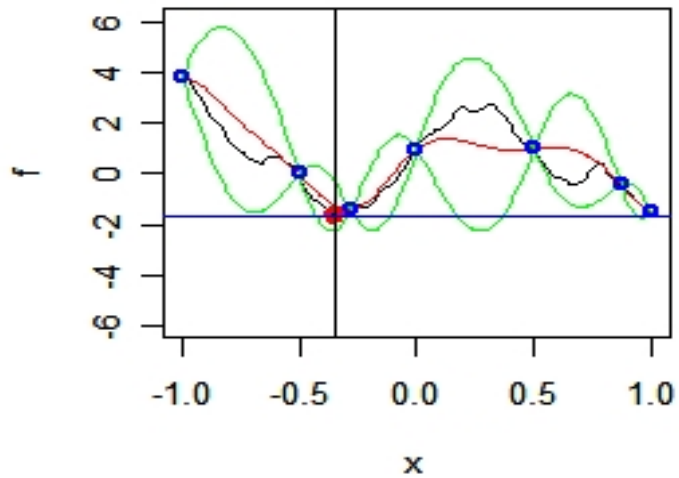
iteration  
1



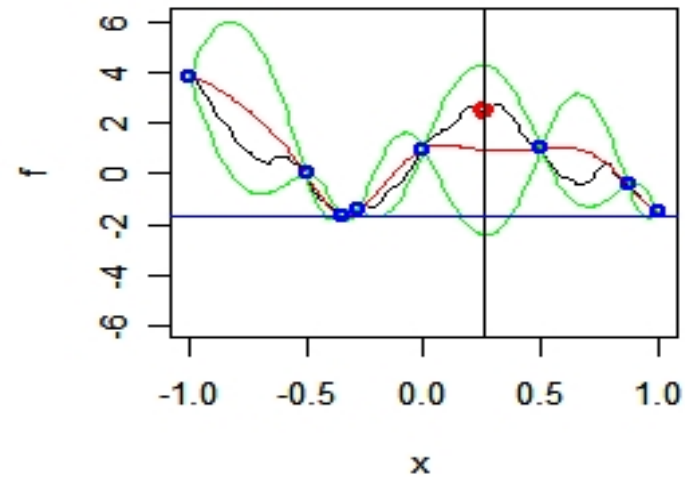
iteration  
2



iteration  
3



iteration  
4

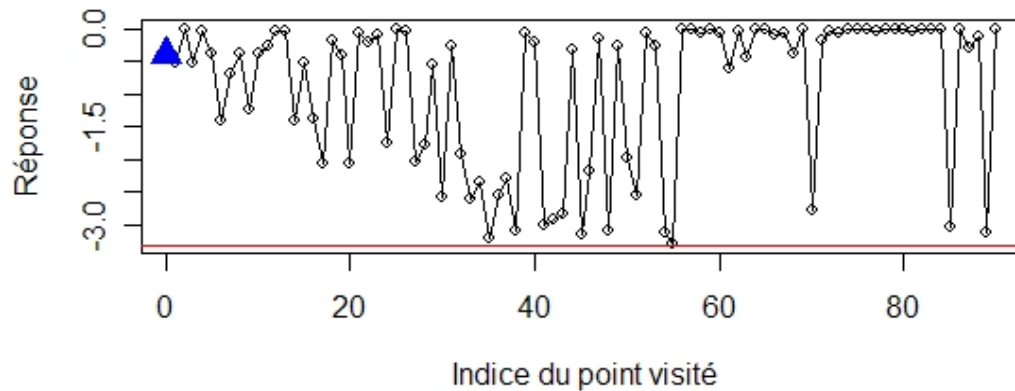


# kriging-based approaches

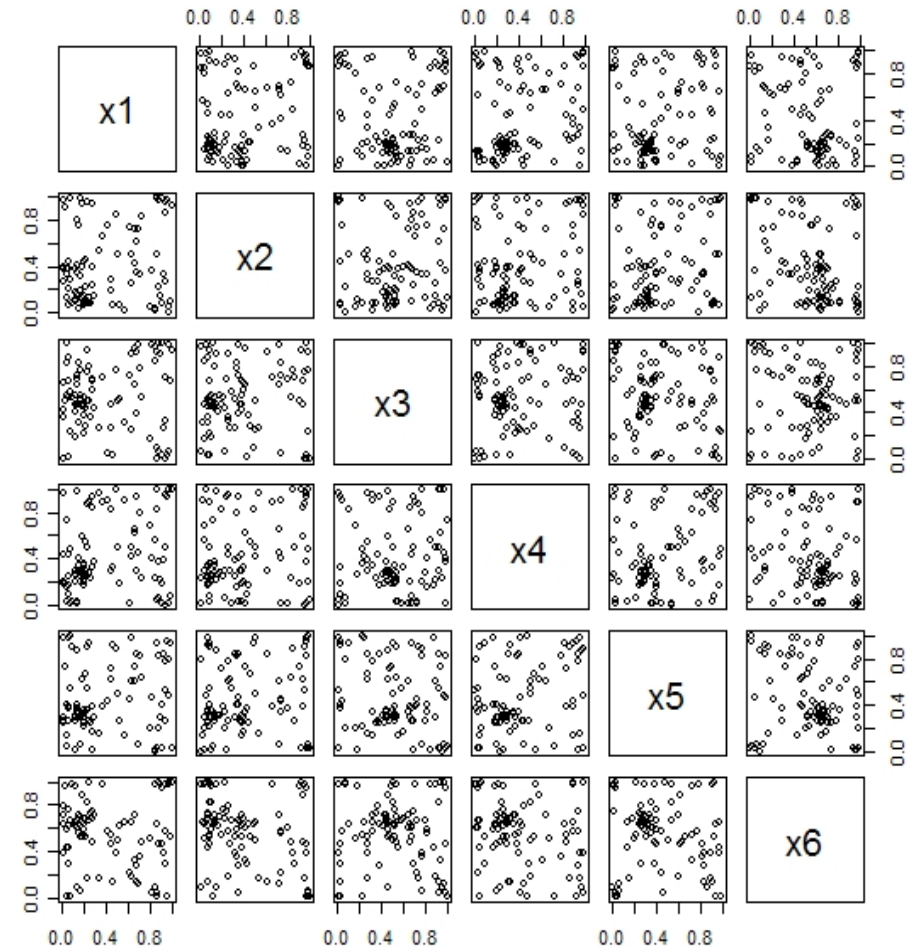
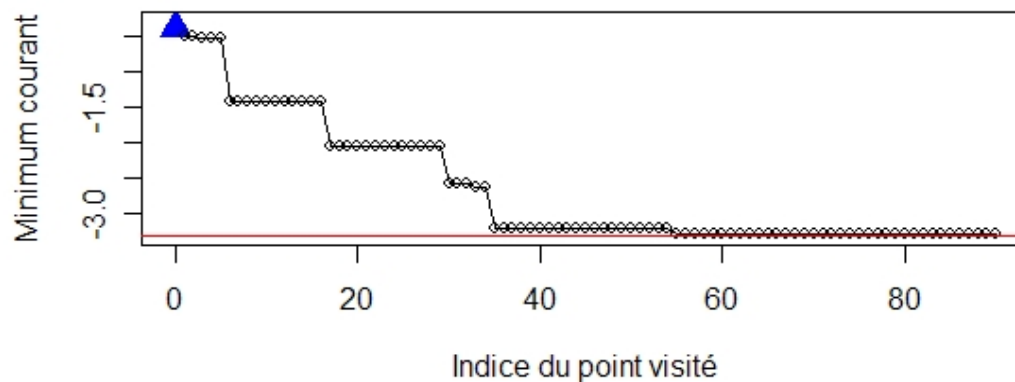
## EI criterion : 6D example

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

Séquence des valeurs observées durant EGO



Séquence du minimum courant durant EGO



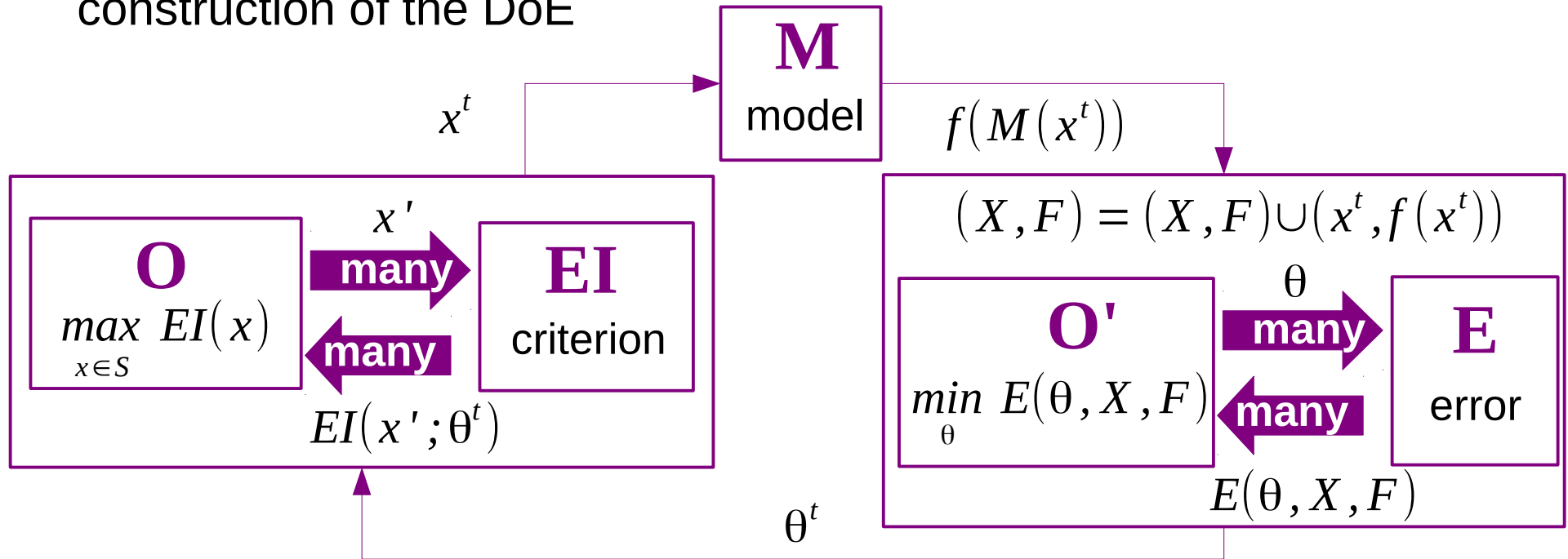
(DiceOptim, D. Ginsbourger et al., 2009)

# kriging-based approaches

## EI criterion : comments

---

Our first example of surrogate criterion, including progress on  $f$  and construction of the DoE



Computational complexity : for kriging, the error is typically minus the likelihood or the cross-validation error --> a  $(t * t)$  covariance matrix need to be inverted many times,  $O(t^3)$ .

---

# kriging-based approaches

## A one-stage approach (1)

So far, optimization of the surrogate criterion and construction of the surrogate (as another optimization problem) have been separated.

One stage approach : maximize the likelihood of the data points conditional on an hypothetical optimum  $(x, f^{\text{target}})$  :

$$\text{CL}(x, \theta) = \text{Prob}(X, F \mid (x, f^{\text{target}}), \theta) \quad (\text{closed form from the multivariate normal law family})$$

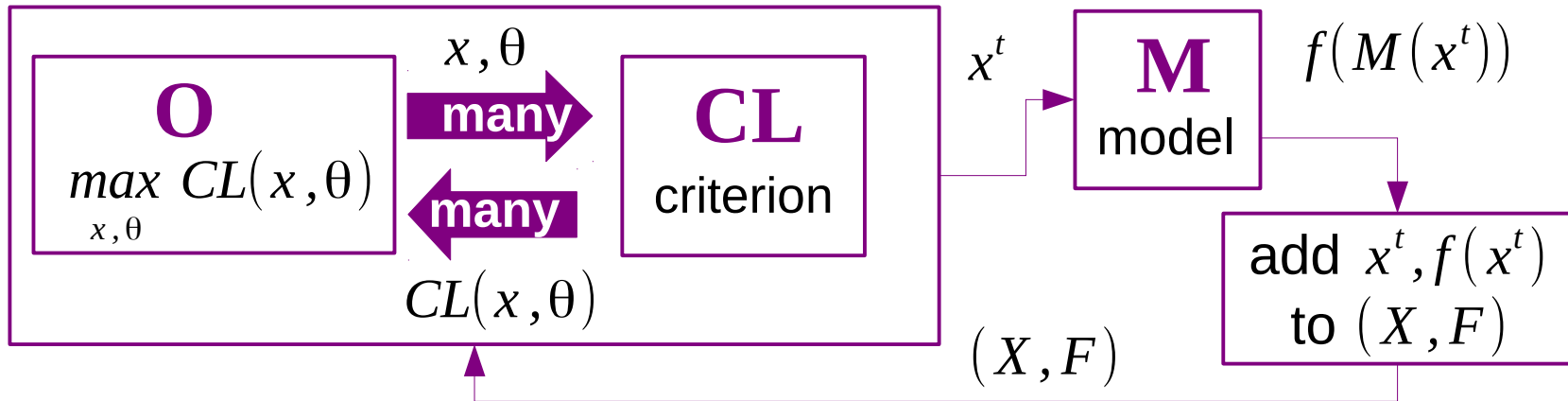
Jones, 2001 (cf. earlier).

A. I. J. Forrester and D.R. Jones, *Global Optimization of Deceptive Functions with Sparse Sampling*, 2010.

---

# kriging-based approaches

## A one-stage approach (2)



*Pros* :  $x$  and the surrogate are chosen together, which partly removes the initial guess on the surrogate that decides which  $x$  is sampled.

*Cons* : guess on  $f^{target}$ , the optimization problem is of larger dimension ( $n + \dim(\theta)$ ).



# Some other kriging-based criteria

---

- Probability of Improvement : Stuckmann 1988, Chaudhuri et al. 2012
- Statistical lower bound  $m(x) - \alpha s(x)$  : Cox and John 1997
- Quantile improvement (for noisy functions) : Picheny et al. 2013
- Multi-points EI : Ginsbourger et al., 2010
- Multi-points PI (many targets), statistical lower bounds (many  $\alpha$ 's) : Jones 2001

A. Chaudhuri, R. T. Haftka, F. Viana, *Efficient Global Optimization with Moving Target for Probability of Improvement*, 2012.

D.D. Cox and S. John, *SDO: a statistical method for global optimization*, Multidisciplinary Design Optimization: State of the Art, SIAM, 1997.


Ginsbourger, D., Le Riche, R. and Carraro, L., *Kriging is well-suited to parallelize optimization*, Computational Intelligence in Expensive Optimization Problems, Springer, 2010.

V. Picheny, D. Ginsbourger, Y. Richet, G. Caplin, *Quantile-based optimization of noisy computer experiments with tunable precision*, Technometrics, 2013

---

# Outline of the talk

---

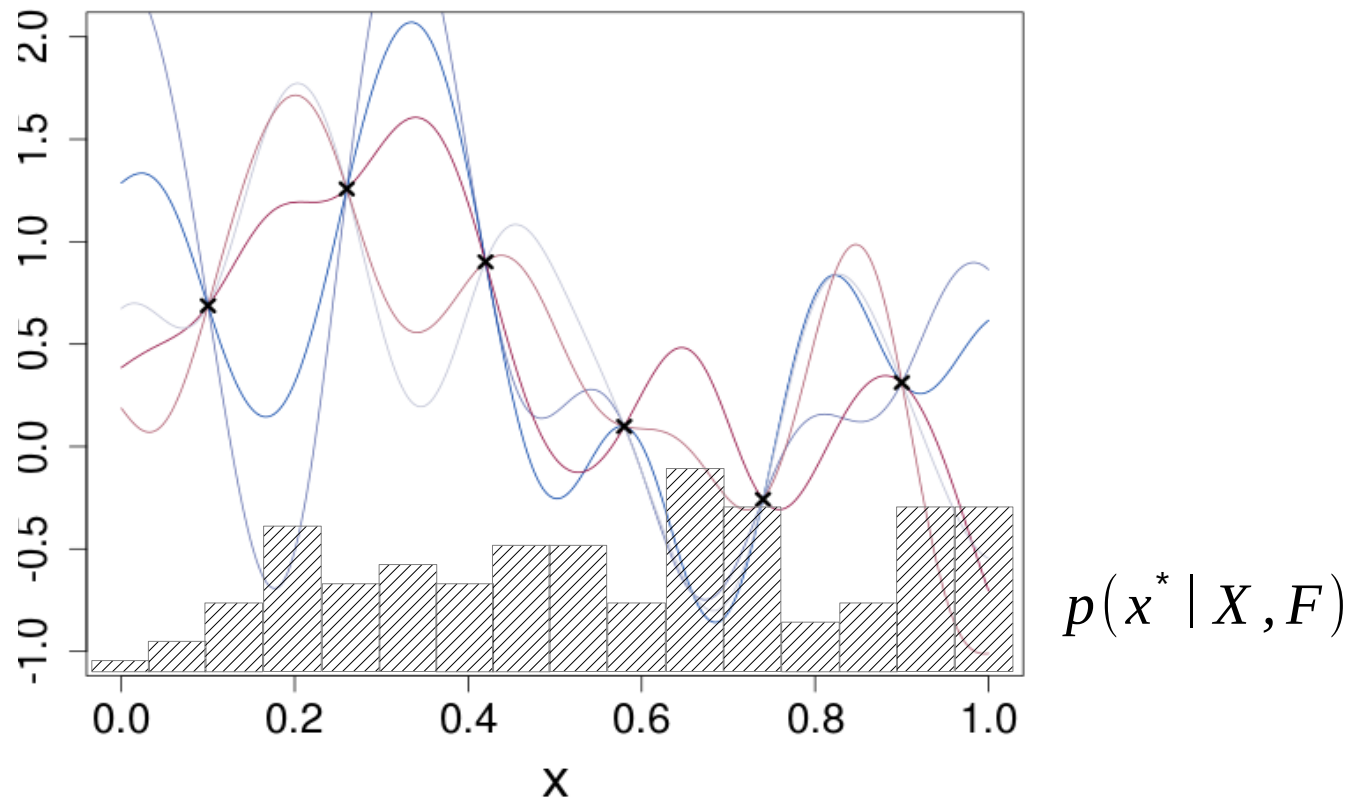
- Context and introduction
- Surrogates and trust regions for local optimization
  - quadratic surrogates
  - any surrogates
- Stochastic optimization using surrogates
- Surrogates with embedded error estimates : kriging
-  • Ensembles of surrogates
  - unstructured
  - structured



# Structured ensemble of surrogates

---

Each of the kriging samples is seen as a possible surrogate (a set of surrogates indexed by the random event  $\omega$ ).



There is a distribution of optima knowing  $(X, F)$  of density  $p(x^* | X, F)$

---

# Structured ensemble of surrogates

## An informational approach

---

Principle : the next iterate is the one that provides the most information on the location of optima

⇒ The next iterate,  $x^{t+1}$ , is the point that reduces the most the conditional entropy of  $p(x^* | ((X, F) \cup (x^{t+1}, F(x^{t+1})))) \equiv p^{t+1}(x^* | x^{t+1})$

$$x^{t+1} = \arg \min_{x \in S} \int_S -p^{t+1}(u | x) \log(p^{t+1}(u | x)) du$$

Pros : a nice way to summarize the contribution of a lot of surrogates.

Cons : high computational complexity.

J. Villemonteix, E. Vazquez, E. Walter, *An informational approach to the global optimization of expensive-to-evaluate functions*, JOGO, 2006.

---

# Unstructured ensemble of surrogates

## Multiple points generation

---

The simplest way to use many (say  $m$ ) surrogates is to generate one iterate per surrogate (with your favorite surrogate optimization technique) and keep them all.

For  $i=1, m$  do

    update  $i$ -th surrogate ,  $s^i(x)$  , with  $(X, F)$

$$x^i = \arg \min_{x \in S} c(x, s^i(x))$$

$$(X, F) = (X, F) \cup (x^i, f(x^i))$$

End

F.A.C. Viana, R.T. Haftka, L.T. Watson, *Efficient Global Optimization algorithm assisted by multiple surrogate techniques*, JOGO, 2013.

---

# Unstructured ensemble of surrogates

## Synthesizing many surrogates

---

One can make one (hopefully better) surrogate of many

surrogates by linear combination,  $\hat{s}(x) = \sum_{i=1}^m w_i^* s^i(x)$

The simplest way to choose the weights  $w_i$  is to optimize them to minimize the squared error (matrix notation),

$$w^* = \arg \min_{w \in \mathbb{R}^m} \|f(X) - s(X)w\|^2$$

$$\text{(normal equations)} \Rightarrow w^* = [s(X)^T s(X)]^{-1} s(X)^T f(X)$$

$$\text{optim. weighted prediction : } \hat{s}(x) = (f(X)^T s(X)) [s(X)^T s(X)]^{-1} \begin{pmatrix} s^1(x) \\ \dots \\ s^m(x) \end{pmatrix}$$

( compare to the kriging average formula : this is an interpolation in the space of surrogates instead of  $S$  )

E. Acar and M. Rais-Rohani, *Ensemble of metamodels with optimized weight factors*, SMO, 2008.

A. Chaudhuri, R. Le Riche and M. Meunier, *Estimating Feasibility Using Multiple Surrogates and ROC Curves*, AIAA/SDM conf., 2013

---

# Conclusions : what about multi-objective optimization

---

Surrogates and optimization have had long-term, yet increasingly intricate relationships. And we just discussed mono-objective optimization. With multi-objective optimization, the range of possibilities still grows ...

- What does it change to go from mono-objective optimization to multi-objective optimization ?
  - Should one build one surrogate per objective function independently ? Not independently ? Or a unique surrogate to learn something about Pareto optimality in the space of optimization variables ?
  - Multi-objective problems are more difficult than mono-objective ones, so they need more points to be sampled : are there any computational limitations that will be hit ?
-

# Acknowledgements

---

Thanks to

- The Lorentz Center for supporting and housing the SAMCO workshop
  - Dimo Brockhoff, Michael Emmerich, Boris Naujoks and Tobias Wagner, the scientific organizers of the workshop for inviting me.
-