Introduction à l'optimisation globale

Rodolphe Le Riche, Charlie Sire
CNRS et Ecole des Mines de Saint-Etienne

cours majeure data science 2020-21
Exploitation mathématique de simulateurs numériques

Organisation de la partie « optimisation globale » de l'UP4

Cours 1 (mercredi 9/12, 13h30-16h45)
- introduction à l'optimisation globale
- algo 1 : recherche aléatoire pure
- algo 2 : recherche locale à point initial aléatoire
- algo 3 : ES-(1+1) à pas constant

TP 1 (jeudi 10/12, 10h-11h30)
- code à trou et test pour les algos 1, 2 et 3

Cours 2 (mardi 15/12, 8h15-9h45)
- CMA-ES simplifié (algo 4),
- Optimisation et processus Gaussien : EGO (début)

TP 2 (mardi 15/12, 10h-11h30)
- suite et fin du TP 1, début EGO

Cours 3 (mercredi 16/12, 8h15-9h45)
- Optimisation et processus Gaussien : EGO (fin)
- recuit simulé

TP3 (mercredi 16/12, 10h-11h30)
- fin EGO

Evaluation : compte rendu de TP (par groupes) + entretien individuel le jeudi 7 janvier matin (soyez disponibles).
AN : mercredi 6 janvier après-midi, les enseignants vous fournissent des questions types pour vous entrainer et sont à votre disposition pour vous répondre.

L'optimisation, un modèle quantitatif pour l'aide à la décision

Formulation mathématique : $min_{x \in S \subset \mathbb{R}^n} f(x)$

f(.), la fonction coût (efforts, masse, violation de contraintes, distance à un but, coût, risque, ...).

Les contraintes, $g(x) \leq 0$, ne sont pas explicitement discutées dans ce cours (cf. « optimisation locale »). Si nécessaire, on peut penser à une pénalisation :

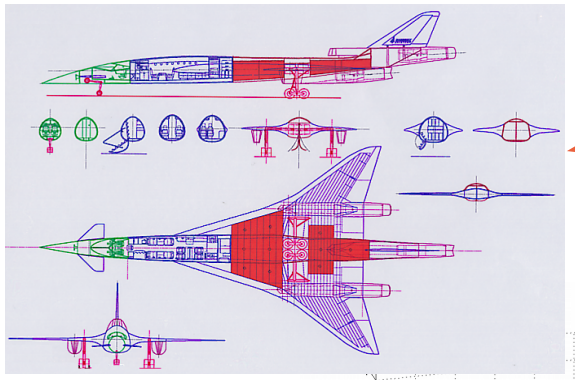
$$\min_{\substack{x \in S \subset \mathbb{R}^n \\ g(x) \leq 0}} f(x) \quad \rightarrow \quad \min_{x \in S \subset \mathbb{R}^n} f(x) + p \times max^2(0, g(x))$$

p , scalaire(s) positif(s).

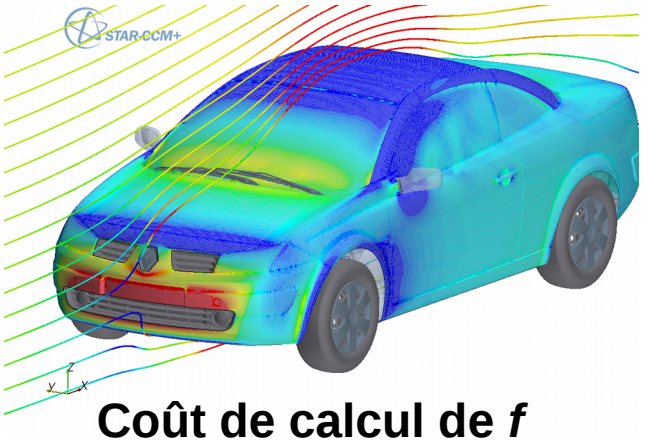
Difficultés de l'optimisation

But : $min_{x \in S \subset \mathbb{R}^n} f(x)$

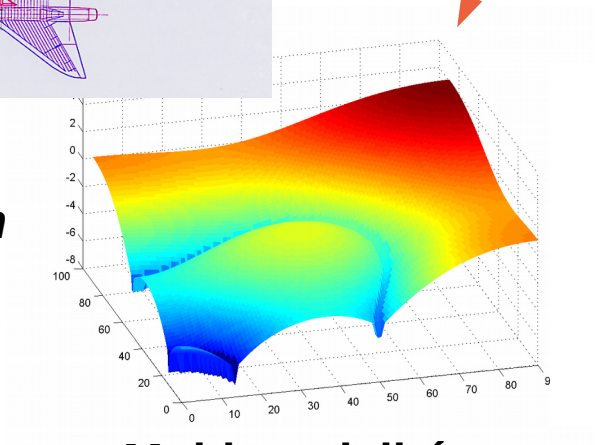
Difficultés



Nombre de variables, *n*

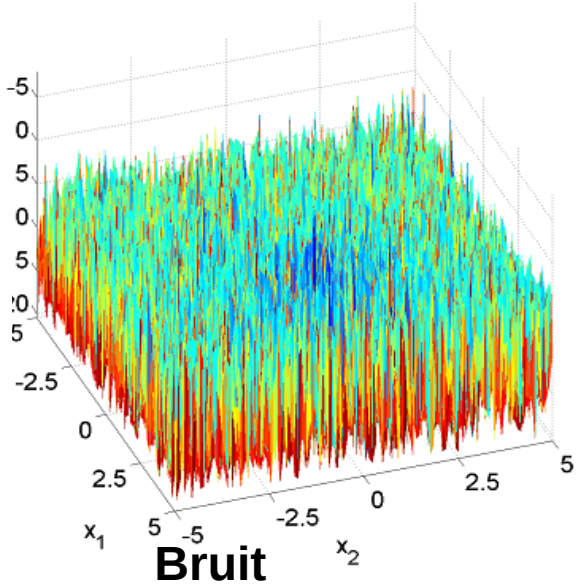


Coût de calcul de *f*



Multi-modalité
(expl. composites)

Mauvais Conditionnement



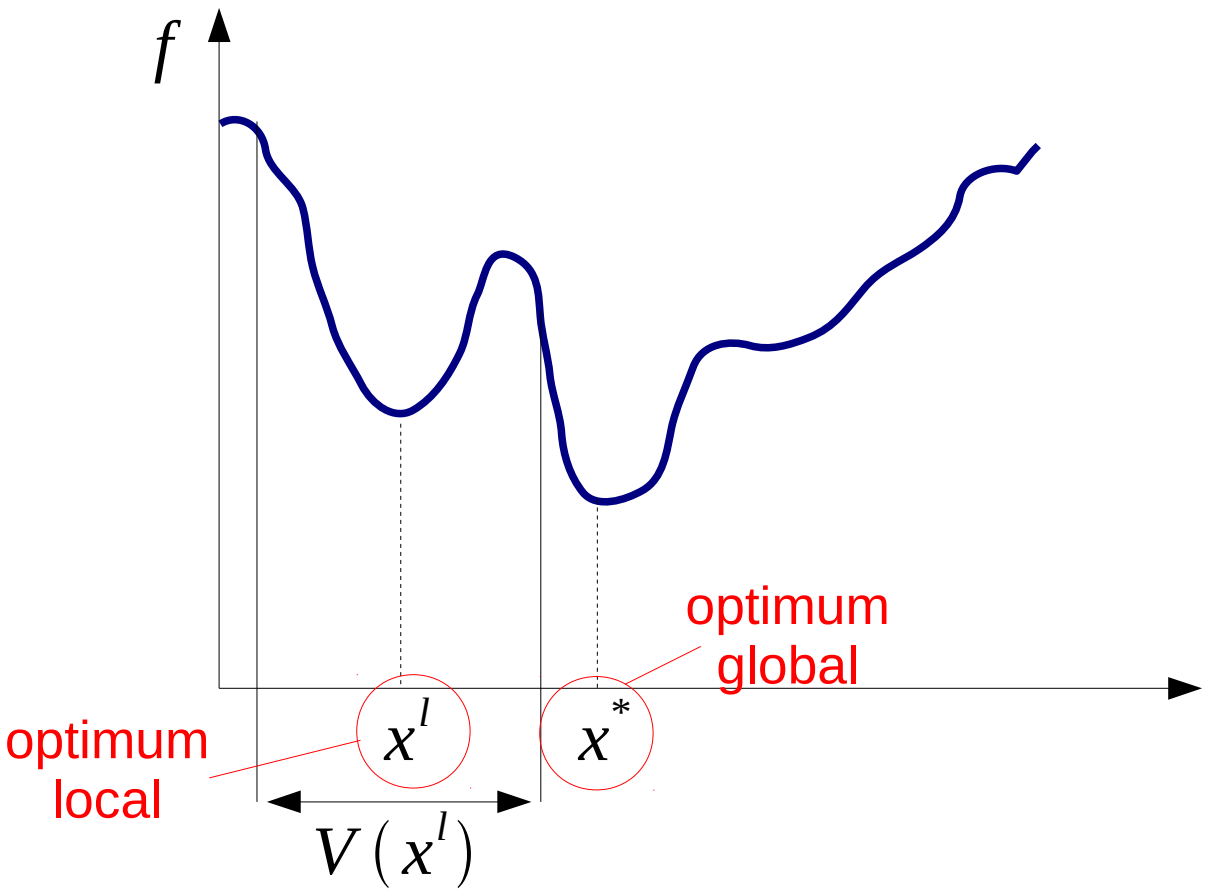
Bruit

Optimisation globale et locale

Problématique de ce cours :

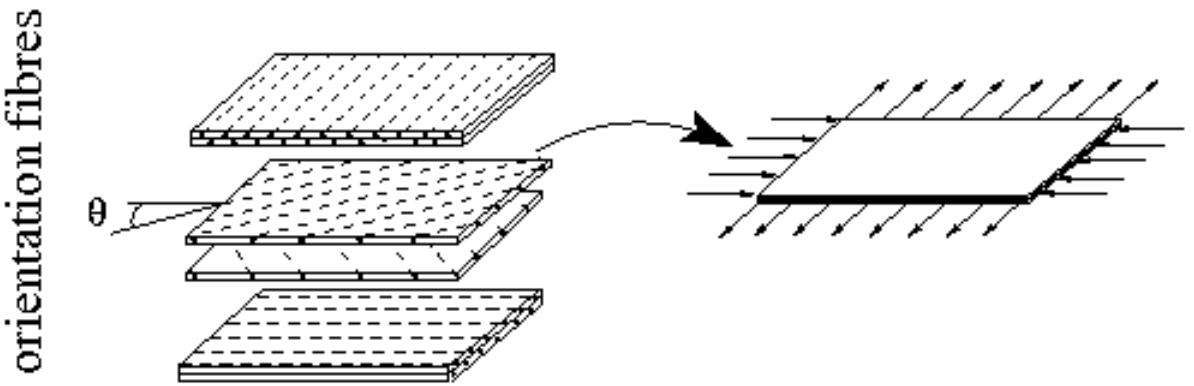
$$\min f(x)$$
$$x \in S \subset \Re^n$$

et accroître les chances de trouver x^*

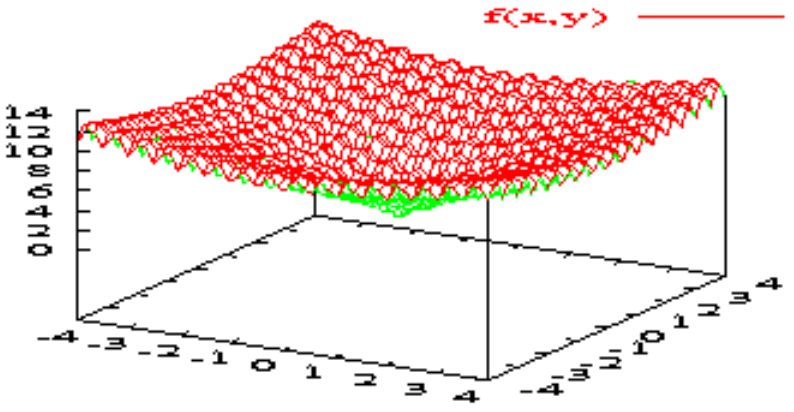


Exemples où l'optimisation globale est nécessaire

Exemple : optimisation de structures composites

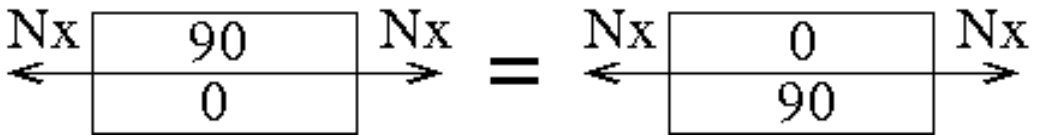


revient à trouver
le minimum
d'une fonction
comme celle–ci



Exemple : optimisation de structures composites (2)

$max_{\theta_i} \; A_{11}$, la raideur longitudinale



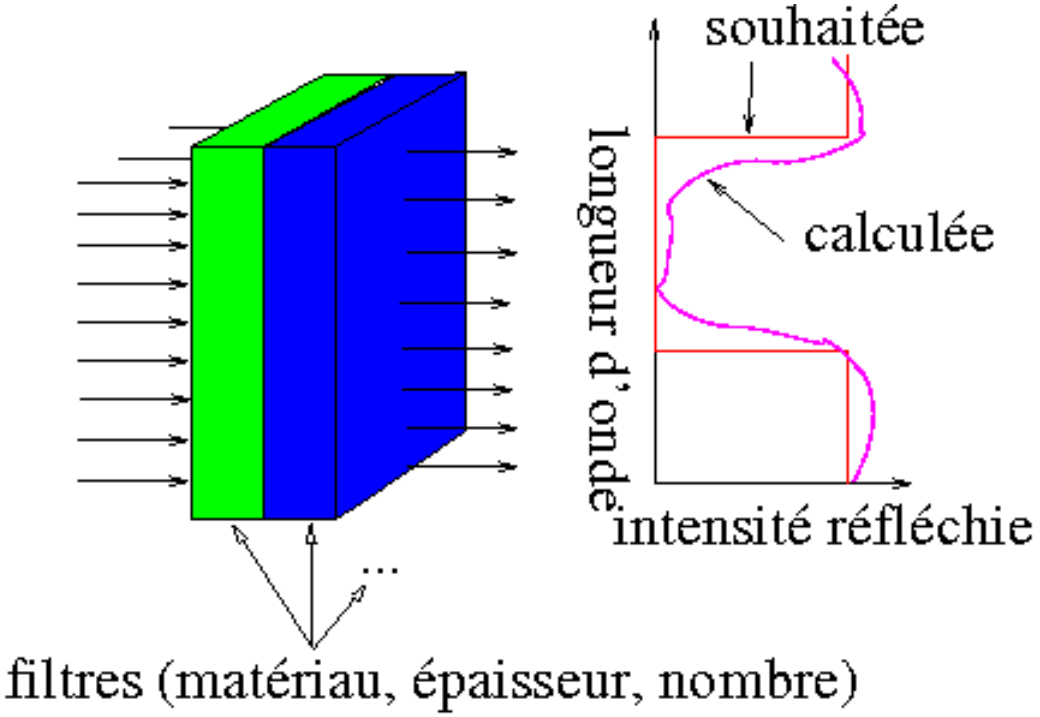
Pour les composites, les optima locaux sont nombreux.

Expl., $N_y / N_x = 0.5$, long=20 in., larg=5 in., graphite-epoxy

séquence	flambement	rupture
$(90_2/\pm 45_2/90_2/\pm 45/90_2/\pm 45_6)_s$	9998.19	10394.81
$((90_2/\pm 45_2)_2/90_2/\pm 45/90_2/\pm 45_3)_s$	9997.60	10187.93
$(\pm 45/90_4/\pm 45/90_2/\pm 45_5/90_2/\pm 45)_s$	9976.58	10187.93

Exemple : optimisation de filtres optiques (1)

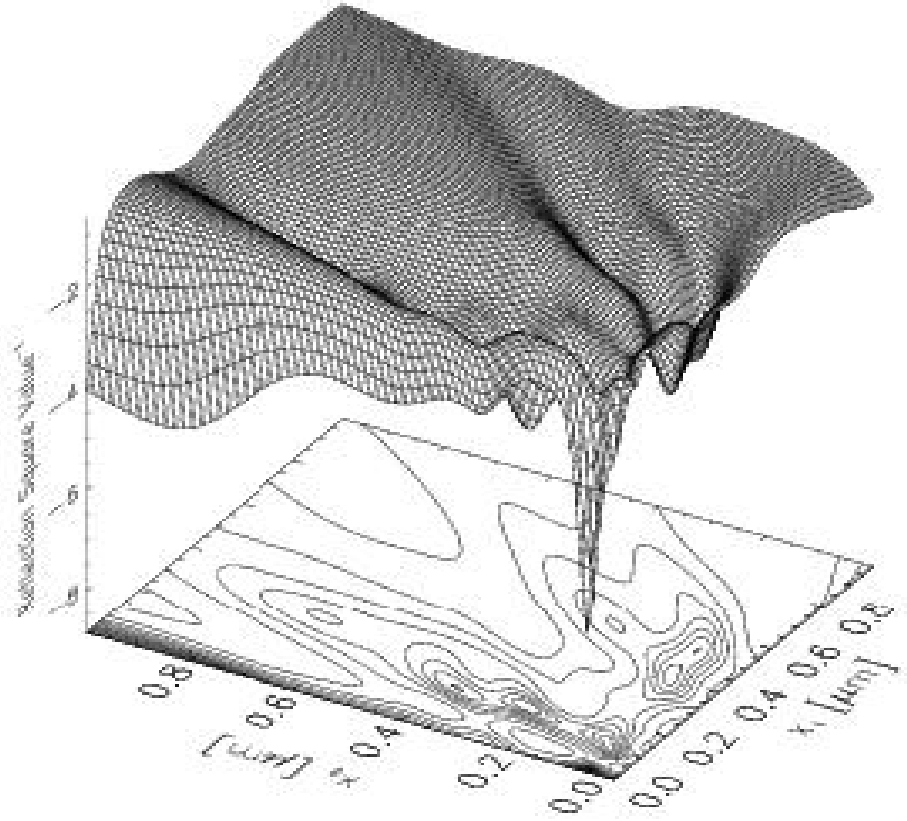
(d'après T. Bäck)



$$min_{\text{nb., mat., épais.}} \int_{\lambda_m}^{\lambda_M} [R_{\text{calc.}}(\lambda) - R_{\text{souhait}}(\lambda)]^2 \, d\lambda$$

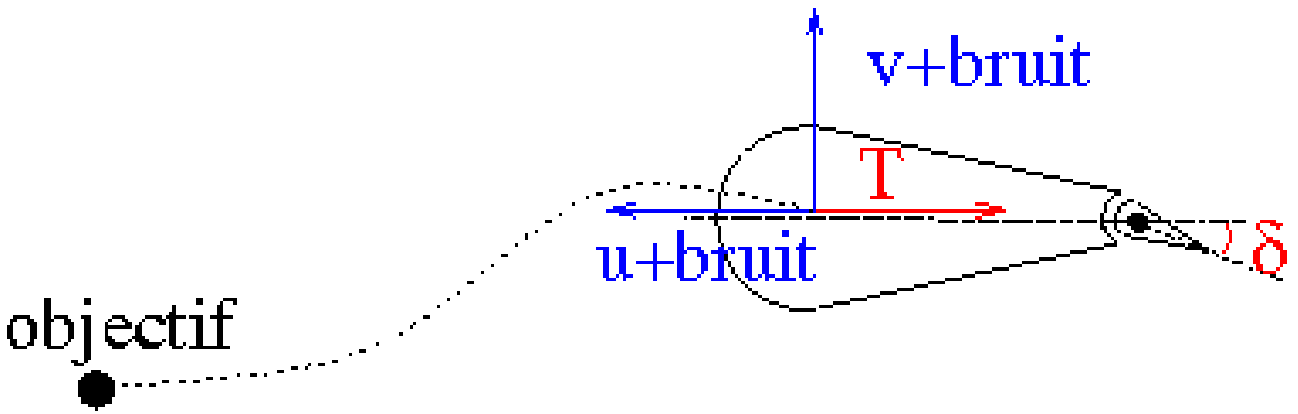
Exemple : optimisation de filtres optiques (2)

Aperçu de la topologie de la fonction écart.
Deux épaisseurs varient (x et y), z l'écart :

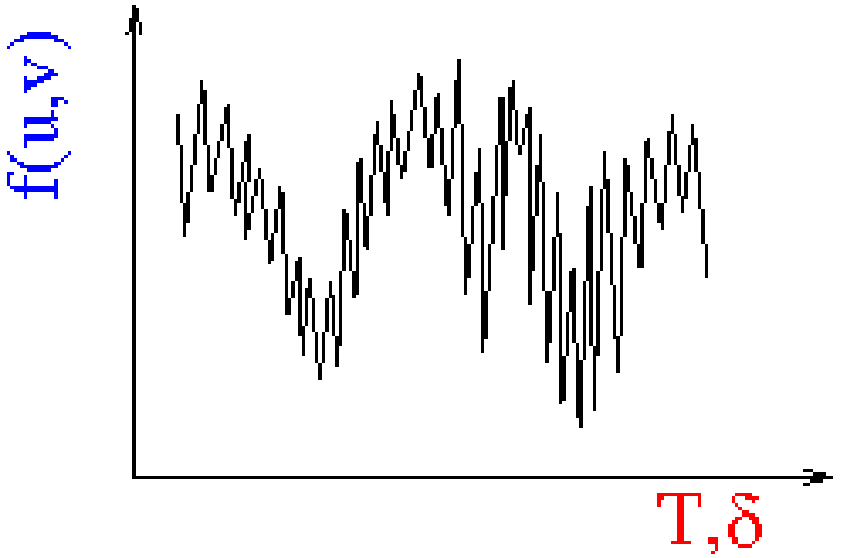


→ de quoi piéger un optimiseur local !

Exemple : contrôle d'un système bruité



revient à minimiser une fonction comme celle–ci



Synthèse : utilité de l'optimisation globale

De nombreux problèmes réels présentent des optima locaux (dont un ou plusieurs peuvent être optima globaux).

- Les problèmes pour lesquels il y a surabondance de variables d'optimisation.
- Les f à structure (pseudo-)périodique.
- Les f bruitées.
- ... en fait c'est le cas général des fonctions qui ne sont pas unimodales (notamment pas strictement convexes).

Formulation du problème de l'optimisation globale

Soit f la profondeur d'un canal. Pour savoir si le canal est navigable, on cherche

- la profondeur minimale, f^* ,

$$P(f^*) \qquad f^* = \left\{ min\ f(x) \ \middle|\ x \in S \subset \Re^n \right\}$$

- les emplacements de profondeur minimale, X^* ,

$$P(X^*) \qquad X^* = \left\{ x \in S \subset \Re^n \ \middle|\ f(x) = f^* \right\}$$

(on notera x^* un des X^*)

Stratégie de résolution

• Par sondage du canal.

• Le **coût** de la recherche est le nombre d'analyses (sondages).

• **Phase globale** (ou d'exploration) : en l'absence d'information a priori, on couvrira le canal de sondages uniformément répartis.

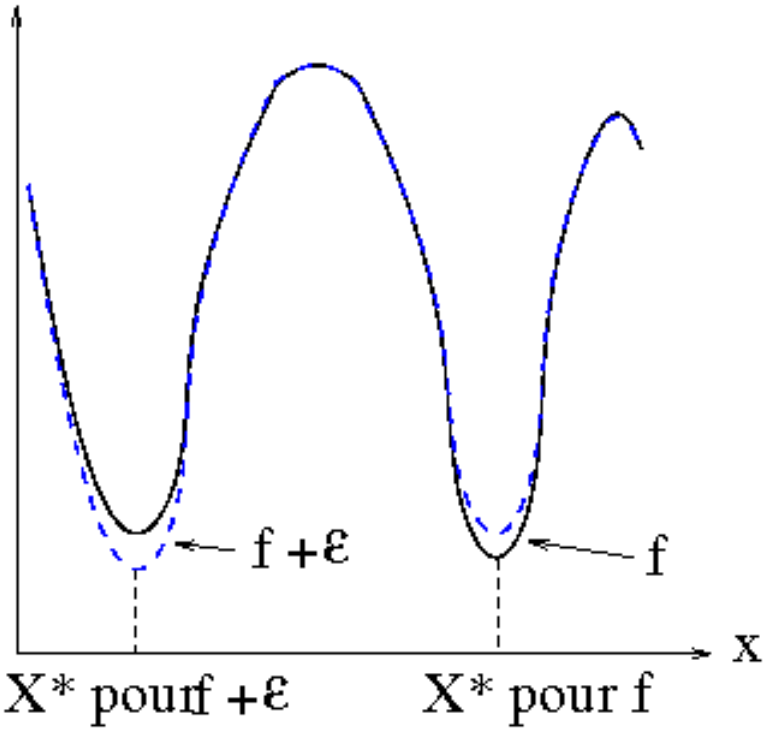
• **Phase locale** (ou d'exploitation ou d'intensification) : certaines régions plus hautes ou plus chahutées pourront être davantage sondées.

• Souvent, des **informations auxiliaires** existent qui peuvent guider la recherche : régularité du fond par régions, bornes sur f^* (trivial ici, $f^* > 0$).

L'optimisation globale : une utopie dans l'absolu

• Problème non résoluble : si il existe un optimum isolé, on ne peut pas le trouver avec une probabilité > 0 (aiguille dans une botte de foin, mat d'un voilier qui a coulé dans le canal).

• Problème instable : il peut y avoir des solutions X* arbitrairement éloignées pour une petite variation de f .



Optimum global essentiel

On s'intéresse aux problèmes que l'on a une chance de résoudre, c'est à dire ceux pour lesquels les optima globaux sont essentiels

$$f^* = min\{ y \mid \forall \epsilon > 0 \ , \ v(x \in S \mid f(x) < y + \epsilon) > 0 \}$$

où v est une mesure d'ensemble , $v(E) = \text{Vol.}(E)/\text{Vol.}(S)$

→ pas de pic isolé.

Problèmes résolubles

Si on connait f^* ou le nombre d'optima locaux, on peut savoir si on a résolu le problème.

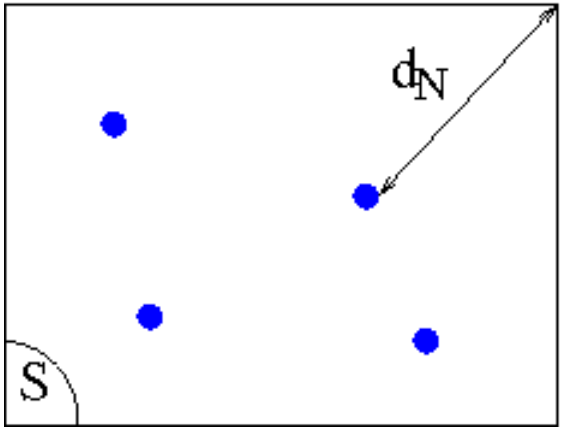
La connaissance de bornes sur la variation de f rend le problème résoluble. Expl.: problèmes Lipschitziens,

$$\exists L \mid \forall x_1, x_2 \ , \ |f(x_2) - f(x_1)| \leq L \|x_2 - x_1\|$$

En effet, avec N points calculés, on peut estimer la qualité de la solution :

$$d_N = max_{x \in S} min_{1 \leq i \leq N} \|x - x_i\|$$

$$\widehat{f^*_N} - f^* \leq L\, d_N$$



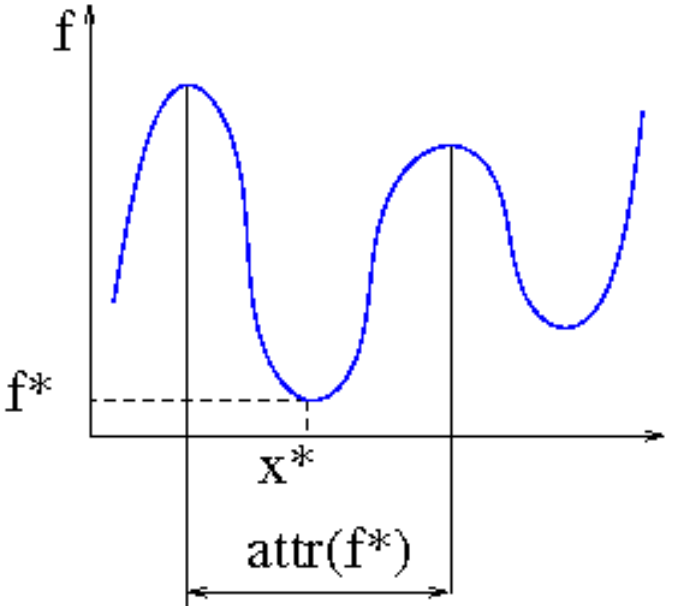
Si $L\, d_N \leq \epsilon$, la solution est connue à ϵ près. DIRECT est un algo. qui utilise L et tous les $f(x_i)$.

Région d'attraction

Soient k minima locaux, $x_1^*, \ldots, x_k^*$, t.q.

$$f(x_1^*) = f_1^* = f^* \leq \ldots \leq f(x_k^*) = f_k^*.$$

Région d'attraction, *attr(f*$_i$)* : ensemble des points de départ d'un gradient de pas infinitésimal (GI) qui mène à f^*_i .



Problèmes résolubles en probabilités

p_i, la proba. qu'un GI converge vers f_i^*, $p_i = v[attr(f_i^*)]/v(S)$.

Pour des fonctions suffisament régulières, $v[attr(f_i^*)] \uparrow$
quand $f_i^* \downarrow$, ainsi $p_1 = max_{1 \le i \le k} p_i$.

$p_1 = 1$ pour une fonction unimodale.

$p_1 \approx 1$, prbl. facile puisque la plupart des recherches locales trouvent f^*.

$p_1 \ge \delta > 0$, prbl. résoluble en probabilité, car la proba. de trouver $f_1^* \to 1$ pour une infinité d'évaluations de f.

La proba. d'avoir au moins un point aléatoire dans $attr(f^*)$ en μ coups est $P_{1,\mu} = 1-(1-p_1)^{\mu}$.

Si $p_1 \approx 0$, optimum instable. A éviter, car le modèle f n'est pas la réalité, erreurs de fabrication.

En pratique

Il y a
• un but, trouver $f(\hat{x}^*)$ le plus bas possible,
• des ressources, un nombre maximal d'analyses,
• et le problème est d'utiliser ces ressources de manière optimale.

• Il n'existera jamais d'algorithme optimal pour tous les problèmes (Th. du « No Free Lunch », D. Wolpert): quand un algorithme progresse sur une classe de fonctions, il régresse sur une autre.

• Utiliser toute connaissance a priori sur le problème : formulation du problème (choix des variables et critères), contraintes

• Un problème résoluble peut être trop coûteux pour être résolu, en particulier en grandes dimensions ($n>>1$).

Classification des méthodes d'optimisation globales (1)

Deux composantes dans toutes les méthodes :

• Composante **globale** ou **exploratrice**, nécessaire pour les fonctions chahutées.

• Composante **locale** ou **exploitatrice** ou d'**intensification**, plus efficace une fois dans attr(f*) .

Classification des méthodes d'optimisation globales (2)

Méthodes stochastiques :

- Rech. aléatoires pures. Expl. *pseudo-code fait en cours*.
- Descentes par perturbations. Expl. *ES-(1+1) à pas constant , pseudo-code fait en cours*.
- Recuits simulés
- Méthodes avec perturbations et populations (algo. évolutionnaires dont CMA-ES, PSO).
- Optimisation statistique

Toutes ces méthodes construisent (implicitement, sauf opt. statistique) une densité de probabilité d'instancier un nouveau x à chaque itération, $p(x)$.

Ces méthodes $\rightarrow$ f^* avec proba. 1 quand coût $\rightarrow \infty$ (si $p(x)>0$).

Recherche aléatoire pure, pseudo-code

- Initialiser x^{min}, x^{max}, t^{max}, $t=0$, $\hat{f}^* = +\infty$
- Tant que $t < t^{max}$ faire,
  - $x' \leftarrow \mathcal{U}[x^{min}, x^{max}]$
  - calculer $f(x')$, $t \leftarrow t+1$
  - si $f(x') < \hat{f}^*$, $\hat{x}^* \leftarrow x'$, $\hat{f}^* \leftarrow f(x')$
    Fin si

Fin tant que

Organigramme simplifié d'un *ES-(1+1)*

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

Tant que $t < t_{max}$ faire,

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

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

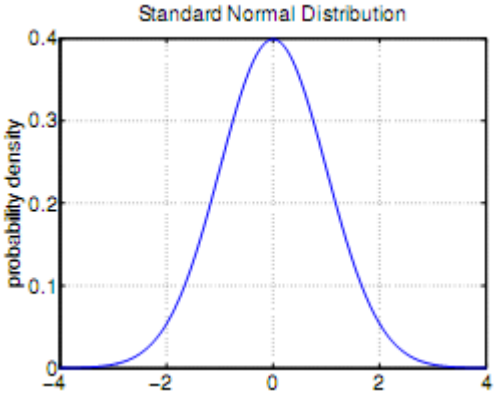
Si $f(x')<f(x)$, $x = x'$, $f(x) = f(x')$ Fin si

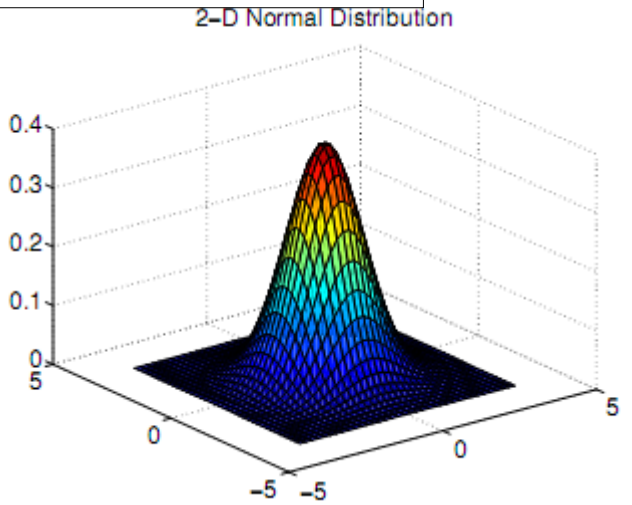
Mettre à jour m *(e.g., $m=x$)* et C

Fin tant que

Loi normale

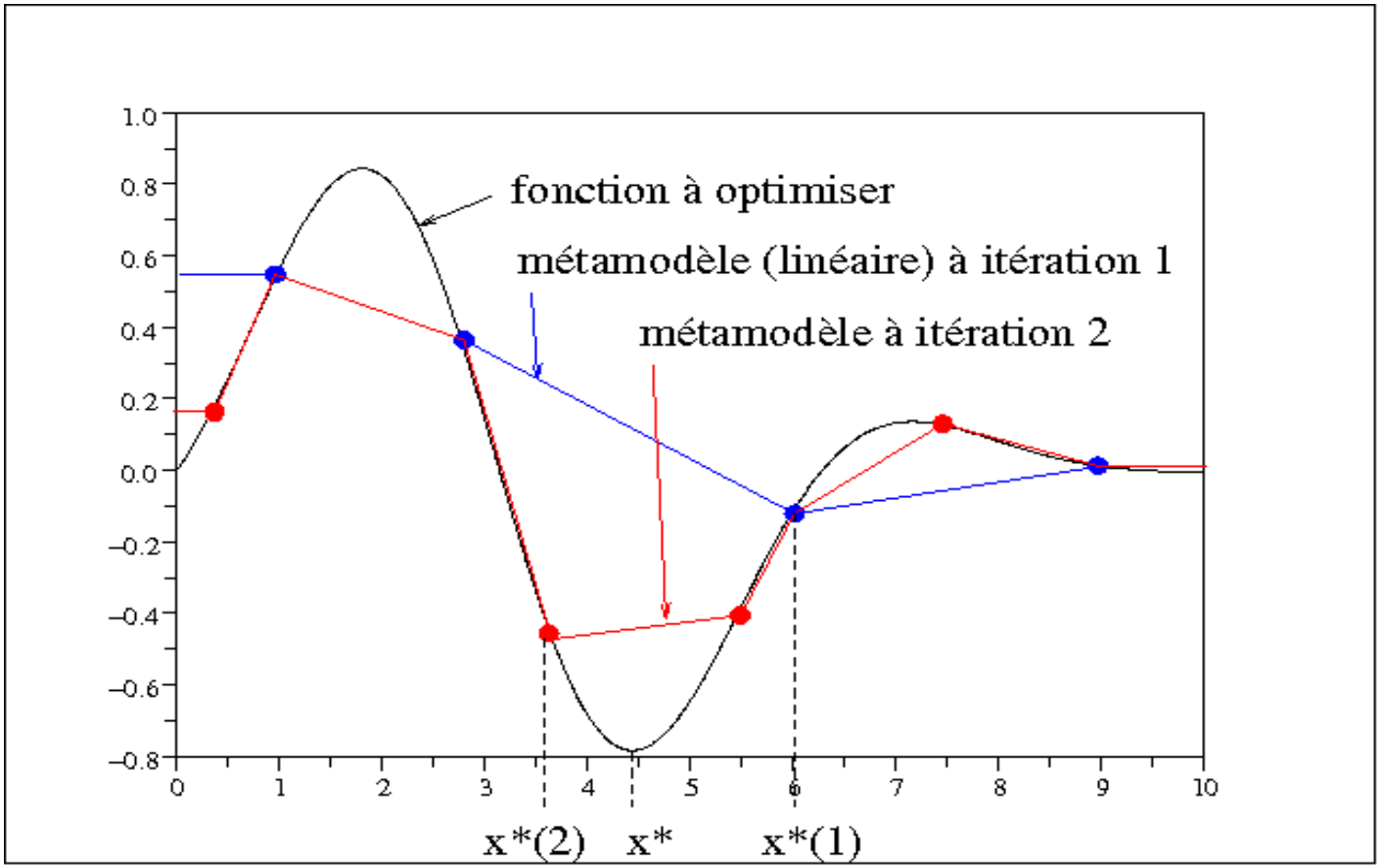
$N(m,C)$





Classification des méthodes d'optimisation globales (3)

Méthodes énumératives déterministes : Elles construisent une approximation globale de f et, souvent, de l'incertitude associée (métamodèle).
Expl. : DIRECT (et méthodes Lipschitziennes), EGO .



Classification des méthodes d'optimisation globales (4)

Méthodes ré-utilisant les recherches locales :

- (local puis global) Redémarrage aléatoire de recherches locales. *pseudo-code fait en cours*.
- (local puis global) Séquence de recherches locales guidées pour ne pas converger vers un optimum local déjà trouvé ;
  - Expl. en pénalisant $f \rightarrow f + P(x_1^*,..., x_c^*)$ (descentes généralisées).
  - Expl. analyse de proximité permet d'interrompre recherches locales convergeant vers des zones déjà explorées.
- (global puis local) Analyse de proximité (clustering) appliquée aux meilleurs points échantillonnés pendant la phase globale pour identifier les régions prometteuses (puis recherches locales).

redémarrage aléatoire de recherches locales

- Initialiser x^{min}, x^{max}, t^{max}, $t = 0$, $\hat{f}^* = +\infty$
- Tant que $t < t^{max}$ faire,
  - $x^{init} \leftarrow \mathcal{U}[x^{min}, x^{max}]$
  - Lancer la recherche locale depuis x^{init}
    $[x', f(x'), t'] \leftarrow$ RECH_LOC(x^{init})
    x' optimum local, $f(x')$ sa fct. coût,
    t' nombre d'appels à f de RECH_LOC
  - $t \leftarrow t + t'$
  - Si $f(x') < \hat{f}^*$
    $\hat{x}^* \leftarrow x'$, $\hat{f}^* \leftarrow f(x')$
    Fin Si

Fin tant que

Nous allons voir
- 2 méthodes stochastiques, CMA-ES et le recuit simulé
- et 1 méthode énumérative déterministe, EGO.

Ce sont des méthodes de recherche dans le volume, contrairement aux méthodes qui cherchent à réduire la dimension du problème (par exemple les méthodes basées sur les gradients).

De nombreuses méthodes ne seront pas discutées (DIRECT, PSO, recuit simulé, couplage local/global, transformation de *f* , Tabu, ...).

Tests des méthodes d'optimisation globale

• La plupart des méthodes d'optimisation globales utilisent des nombres aléatoires. Expl., choix pts initiaux.

• Typiquement, le coût d'une optimisation est le nombre d'évaluations de la fonction objectif, f.

• Ne pas juger les méthodes en 1 exécution, mais qualifier la distribution des résultats (moy., σ^2) à un coût donné.

• La plupart des méthodes possèdent des paramètres dont le réglage optimal dépend de f. cf. Th. ``No Free Lunch".

Cours 2

Algorithmes stochastiques : CMA simplifié, recuit simulé

Organigramme simplifié d'un *ES-(1+1)*

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

Tant que $t < t_{max}$ faire,

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

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

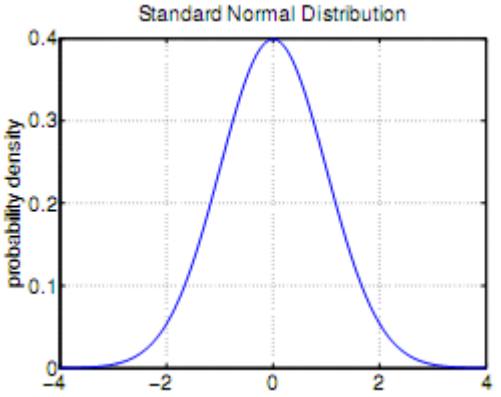
Si $f(x')<f(x)$, $x = x'$, $f(x) = f(x')$ Fin si

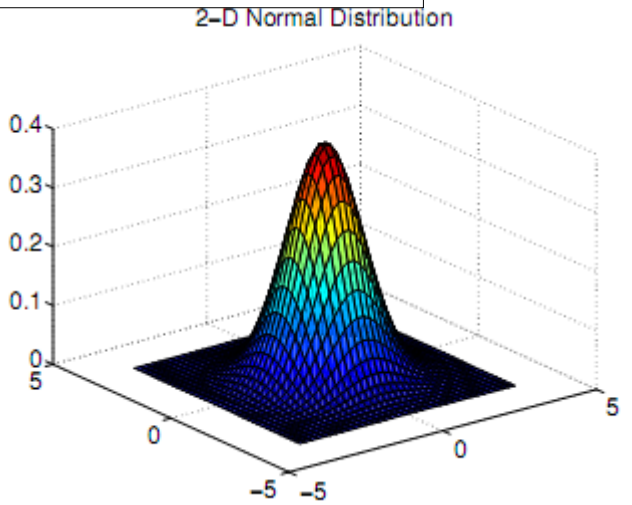
Mettre à jour m *(e.g., $m=x$)* et C

Fin tant que

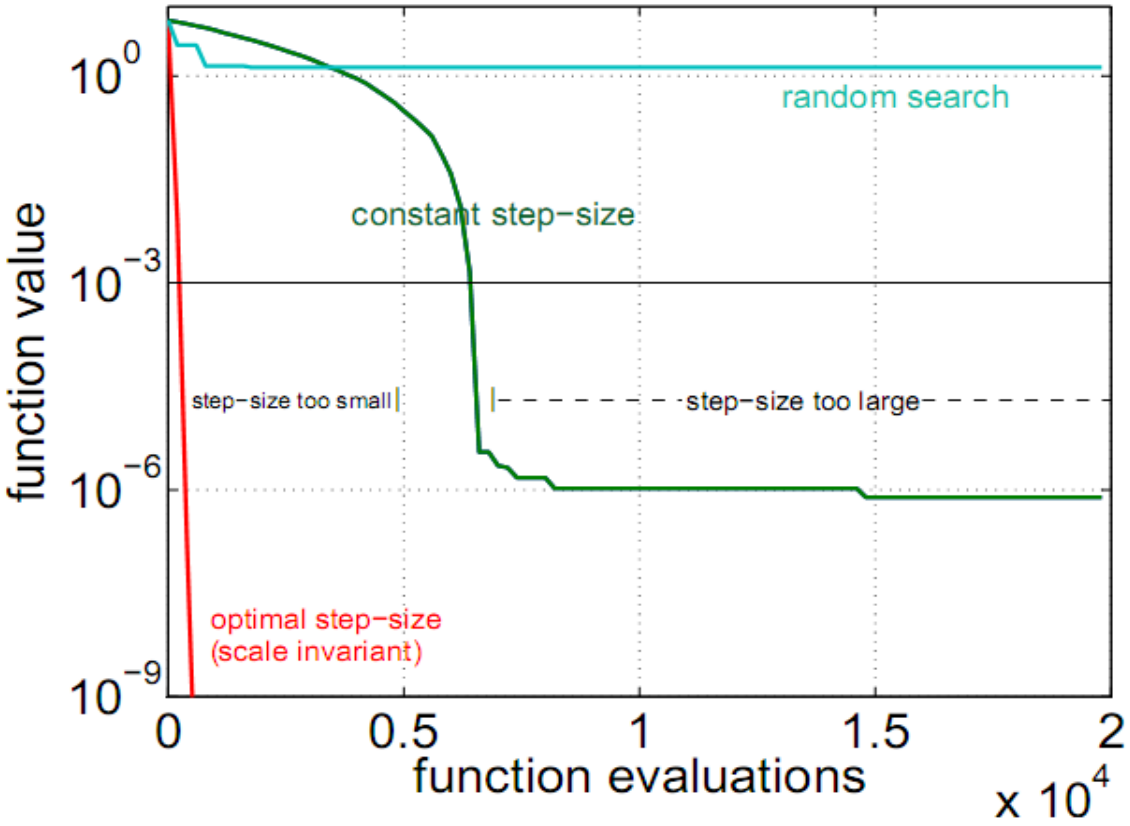
Loi normale

$N(m,C)$





Adapter le pas (C^2) est 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)

Ici ES(1+1) à pas isotrope : $\boldsymbol{C} = \sigma^2 \mathrm{I}$, σ est le pas.
Avec un pas optimal ($\approx \|x\|/n$) sur la fonction sphère, la performance ne se dégrade qu'en $O(n)$ (à comparer à DIRECT) !

ici, théorie sur le choix du pas optimal dans le cadre de ES-(1+1) pour les fonctions quadratiques

Méthode stochastique : CMA-ES

(N. Hansen et al., à partir de 1996, puis développements avec A. Auger)

CMA-ES = *Covariance Matrix Adaptation Evolution Strategy* = optimisation par échantillonage et mise à jour d'une gaussienne.

La méthode état de l'art en optimisation stochastique. Fonctionne sur un principe fondamentalement différent d'EGO et DIRECT ← un point de vue complémentaire.

Caractéristiques de CMA-ES

CMA-ES est une stratégie d'évolution *ES-*(μ,λ) :

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

Tant que $t < t_{max}$ faire,

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

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

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

Mettre à jour m et C avec les μ meilleurs, $x^{1:\lambda}$,...,$x^{\mu:\lambda}$

Fin tant que

m *et* C sont mis à jour en utilisant

- les **pas** qui ont le mieux réussi,
- un **cumul dans le temps** de ces pas.

Ici :

expliquer la différence entre la covariance des bons points et la covariance des bons pas

introduire la moyenne temporelle des matrices de covariance

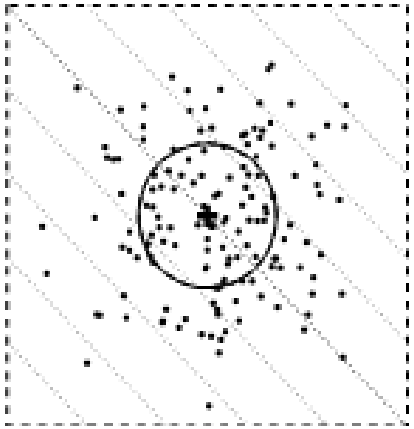
introduire la mise à jour de rang 1

introduire la notion de pas cumulé

CMA-ES simplifié : adaptation de C^2 par les derniers bons pas

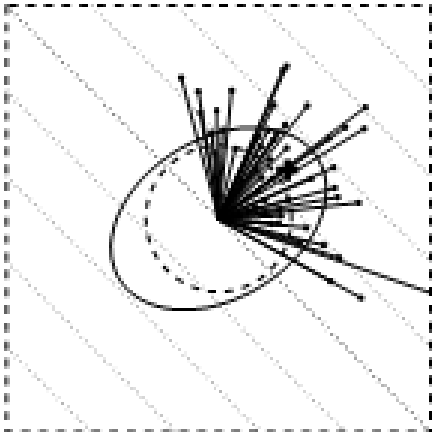
(A. Auger et N. Hansen, 2008)

Initialisation : $m \in S$, $C = I$, $c_1 \approx 2/n^2$



échantillonage

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

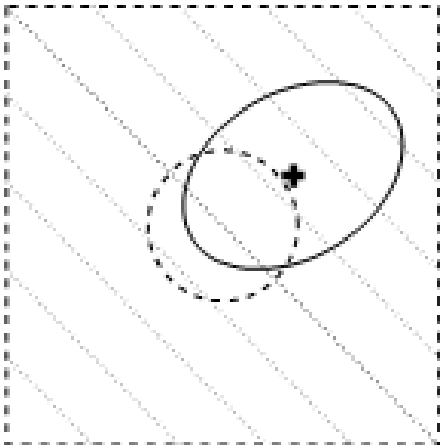


sélection

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

m.à j. C de rang 1

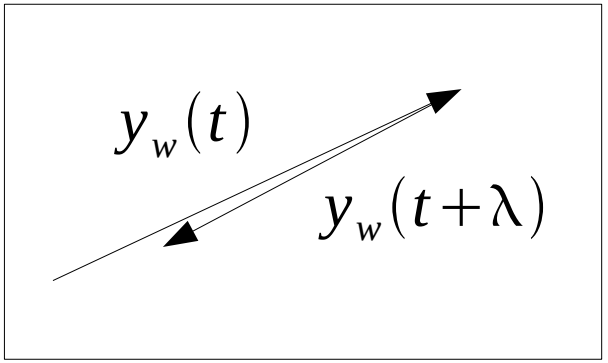
$$C \leftarrow (1 - c_1) C + c_1 y_w y_w^T$$



m.à j. m

$$m \leftarrow m + y_w$$

CMA-ES simplifié : cumul temporel des bons pas



Lorsque les pas respectifs sont anti-corrélés, il faut pouvoir réduire le pas.
Impossible avec $y_w\, y_w^T$ car = $(-y_w)\,(-y_w)^T$.

Cumul (amortissement exponentiel dans le temps) :

$$p_c \leftarrow \underbrace{(1-c_c)}_{\text{amortissement}}\, p_c + \underbrace{\sqrt{1-(1-c_c)^2}\sqrt{\mu}}_{\text{normalisation}}\, y_w$$

$$C \leftarrow (1-c_1)C + c_1\, p_c\, p_c^T \quad \leftarrow \text{rang 1}$$

$$c_c \approx 3/n \;\;,\;\; c_1 \approx 2/n^2$$

Le CMA-ES état de l'art

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

Caractéristiques supplémentaires (/ transparents précédents) :

- Facteurs de pondération, $y_w = \sum_{i=1}^{\mu} w_i \, y^{i:\lambda}$
- Mises à jour de rangs 1 et m simultanées.
- Pas global adapté par cumul temporel, $C \rightarrow \sigma^2 \, C$.
- Redémarrage avec taille de population croissante ou double population (une grande, une petite).

Méthode stochastique : les recuits simulés

Simulated annealing, Kirkpatrick, Gelatt and Vecchi, 1983.

"Le**s"** car de nombreuses implémentations sont possibles, il s'agit donc d'une famille d'algorithmes.

le recuit simulé : introduction

• Comme les algos évolutionnaires (dont CMA-ES), on peut les présenter à travers une métaphore, celle du recuit (annealing) en métallurgie :

des atomes cherchant un état minimal d'énergie libre (f ici) et dont les transitions (x vers x') sont liées à la température. Un état d'énergie minimal (métal sans défauts $\approx min\ f$) est atteint en maîtrisant la décroissance de température.

• Populaire car peut être utilisé avec des nombres réels et discrets.

• **(nouveau) L'algorithme change de comportement en fonction du nombre d'appels t à la fonction coût f.**

le recuit simulé : principe

$min_{x \in S \subset \mathbb{R}^n} f(x)$, x ``position des atomes",

X position vue comme une variable aléatoire,

$f(x)$ ou $f(X)$ énergie associée

Perturbation de $x \rightarrow x'$ (position candidate)

Si $f(x') < f(x)$, accepter x'

Sinon $(f(x') \geq f(x))$, accepter x' avec probabilité

$$P_{\text{acc}} = \exp\left(-\frac{f(x') - f(x)}{k_B T}\right)$$

(critère de Metropolis, 1953)

k_B constante de Boltzmann

le recuit simulé : analyse mathématique

Algo. populaire du fait de résultats mathématiques

• Une adaptation des algorithmes de Metropolis-Hastings (Markov Chain Monte Carlo).

• T constante : l'application répétée du recuit simulé (accepte si meilleur ou critère de Metropolis) produit la densité limite

$$p(X=x) = k_T \exp\left(-\frac{f(x)}{k_B T}\right)$$

(k_T constante de normalisation)

(cf. Introduction to stochastic search and optimization, J.C. Spall, chap.8, 2003)

• T variable : convergence asymptotique (t grand) vers les optima globaux si la température

$$T > const / log(t)$$

(schéma trop lent en pratique)

le recuit simulé : algorithme générique

à choisir, -> diff. versions de l'algo (dont non continues)

1. Initialisations

   $T(t)$ (décroissance en température en fonction de t) ,

   $V(x)$ (perturbation stochastique dans un voisinage de x) ,

   $t^{\max}$, $\hat{x}^*$, $\hat{f}^* = f(\hat{x}^*)$, $t \leftarrow 1$, $T \leftarrow T(1)$

Tant que $t < t^{\max}$ faire

2. Perturbation de $\hat{x}^*$, $x' = V(\hat{x}^*)$ (position candidate)
3. Calculer $f(x')$, $t \leftarrow t+1$
4. Acceptation ou non de la perturbation

   Si $f(x') < f(x)$, $\hat{x}^* \leftarrow x'$, $\hat{f}^* \leftarrow f(x')$

   Sinon $(f(x') \geq f(x))$,

   $$P_{\text{acc}} = \exp\left(-\frac{f(x')-f(x)}{T}\right) \quad , \quad u \sim U[0,1]$$

   Si $u < P_{\text{acc}}$, $\hat{x}^* \leftarrow x'$, $\hat{f}^* \leftarrow f(x')$ Fin Si

   Fin Si
5. Mise à jour température , $T \leftarrow T(t)$

Fin tant que

le recuit simulé : version standard dans R^n

Perturbations gaussiennes

$$V(x) = x + \sigma N(0,1)$$

→ choisir σ

Décroissance de la température linéaire

$$T(t) = a \times t + b$$

a et b tels que $P_{acc} = P_0$ au début et $P_{acc} = P_f$ à la fin (2 éq. à 2 inc.)

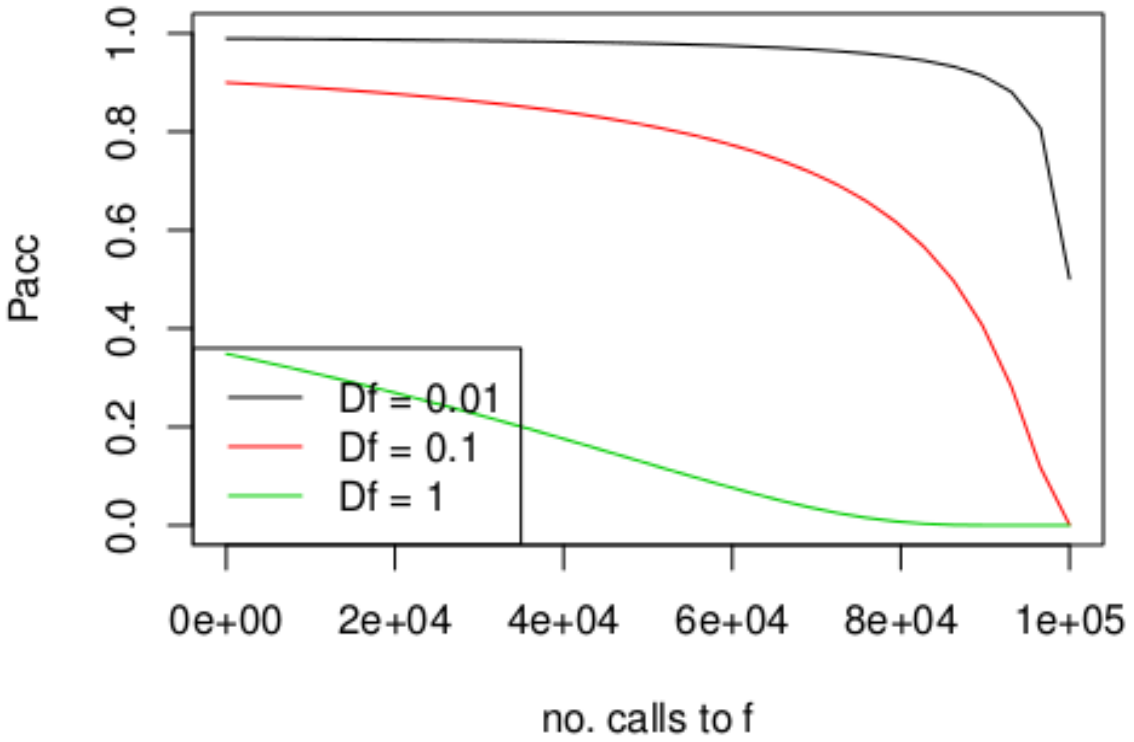
le recuit simulé : exemple

Décroissance de la température linéaire

$$T(t) = a \times t + b$$

a et b tels que $P_{\text{acc}} = P_0$ au début et $P_{\text{acc}} = P_f$ à la fin (2 éq. à 2 inc.)

(expl.) $P_{\text{acc}}(t = 10\ ,\ \Delta f = 0.1) = 0.9$ et $P_{\text{acc}}(t = 100000\ ,\ \Delta f = 0.1) = 10^{-3}$



Cours 3

Krigeage et optimisation : EGO

Méthode déterministe : EGO

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

EGO = Efficient Global Optimization = utilisation d'un métamodèle de krigeage et maximisation du progrès espéré à chaque itération.

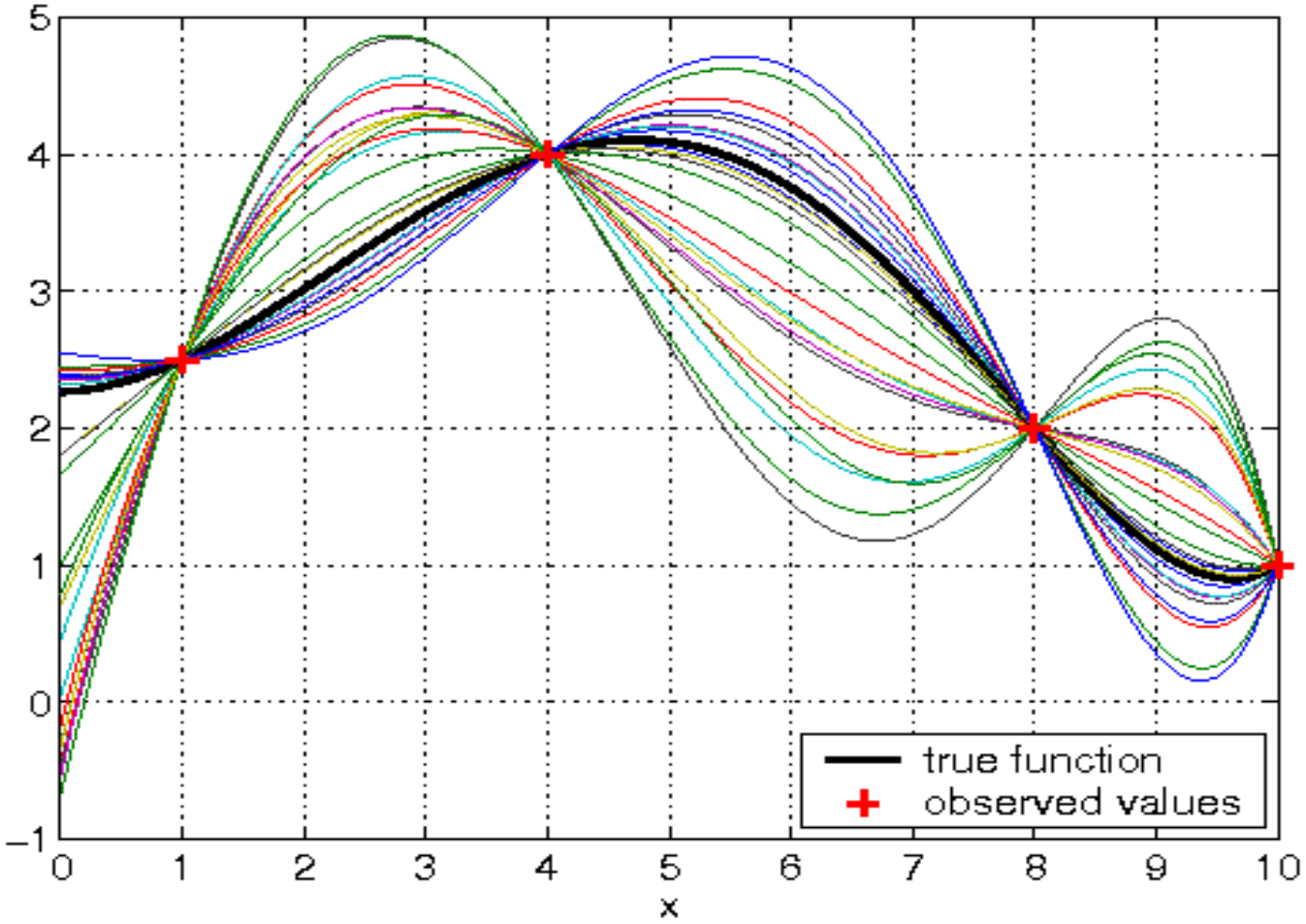
[Comme DIRECT, construit une suite de points dense dans S. Le krigeage remplace les rectangles.]

Améliorations : J. Villemonteix et al., 2006; D. Ginsbourger et al., 2007.

Krigeage (1/2)

f a été observées aux points x_1, ... , x_t .

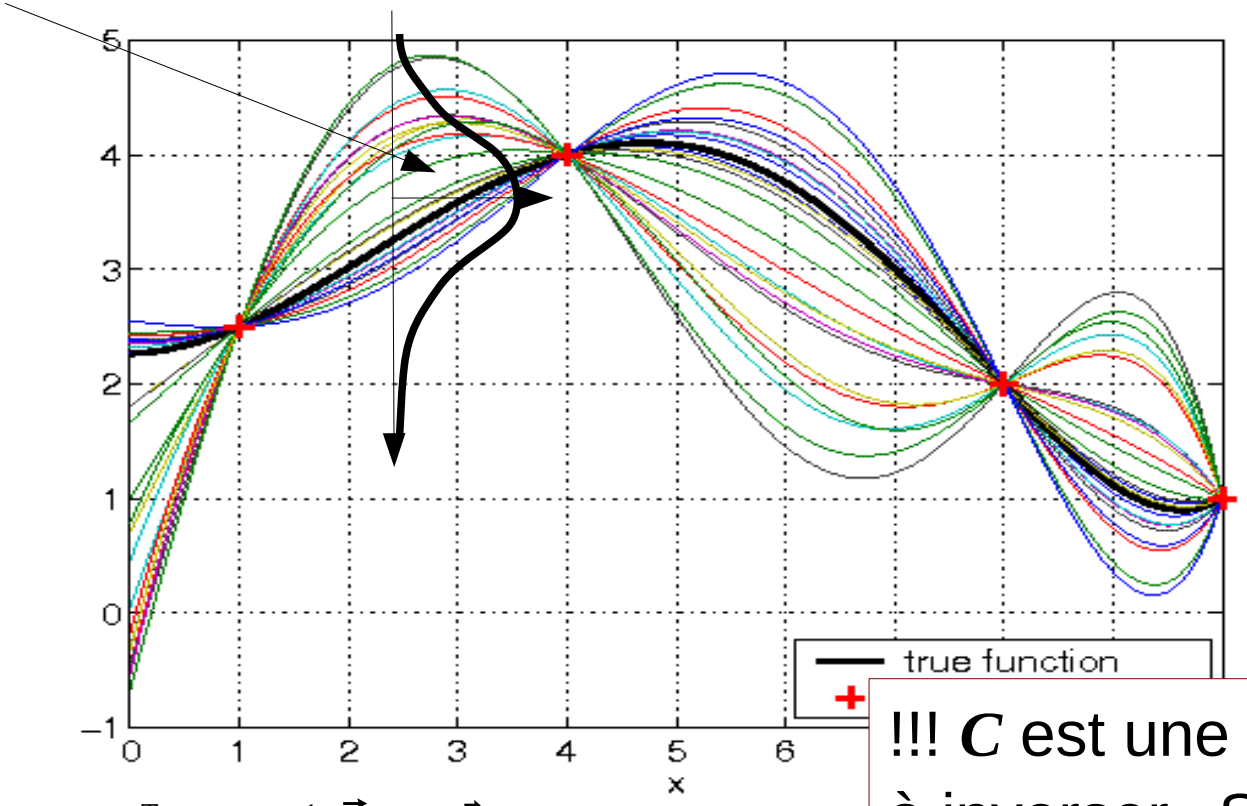
Mais de nombreuses fonctions restent possibles :



Krigeage (2/2)

Krigeage = processus gaussien conditionnel.

$$[\, F(x) \mid f(x^1), \ldots, f(x^t)\,] \;\sim\; N\left(m_{OK}(x)\;,\; s^2_{OK}(x)\right)$$



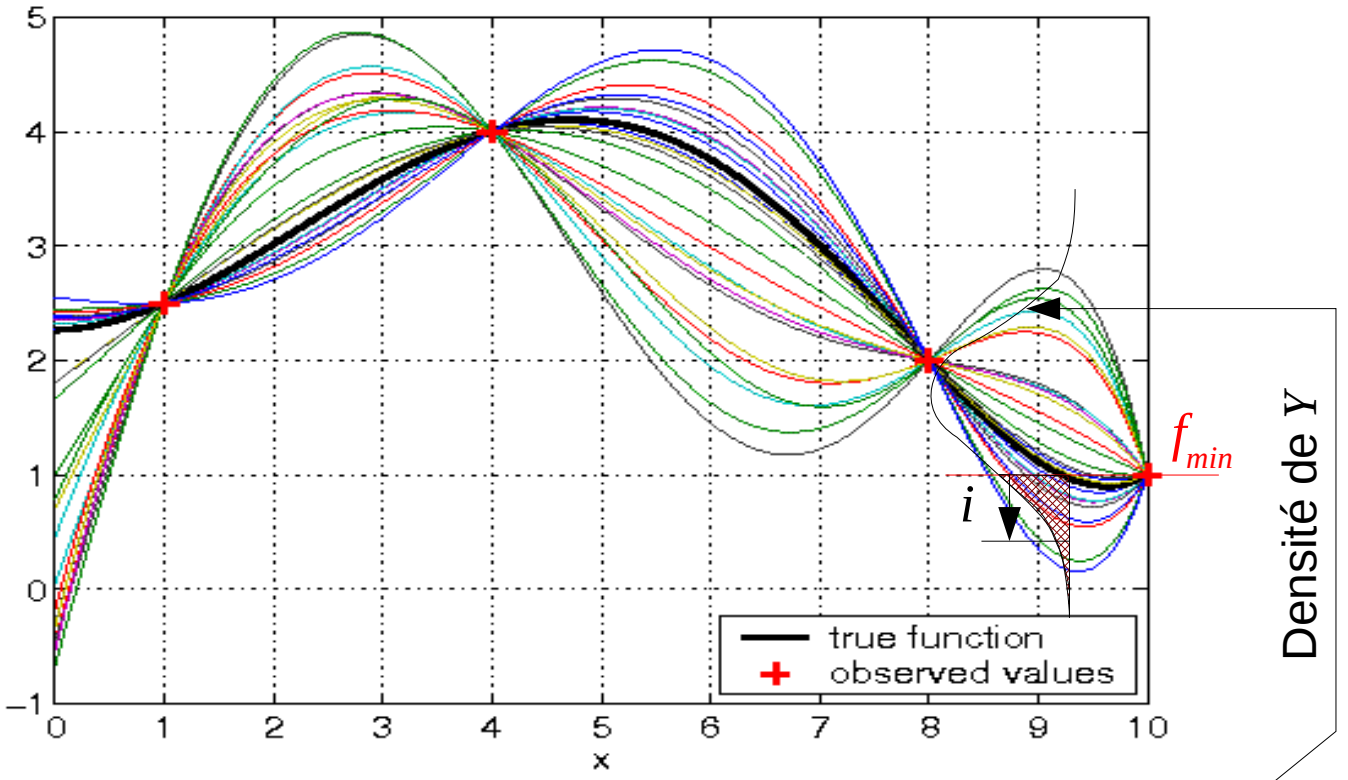
$$m_{OK}(x) = \hat{\mu} + \vec{c}^{\,T}(x)\boldsymbol{C}^{-1}(\vec{f} - \hat{\mu}\,\vec{1})$$

$$s^2_{OK}(x) = \sigma^2 - \vec{c}^{\,T}(x)\boldsymbol{C}^{-1}\vec{c}(x) + \frac{(1 - \vec{c}^{\,T}(x)\boldsymbol{C}^{-1}\vec{1})^2}{\vec{1}^{\,T}\boldsymbol{C}^{-1}\vec{1}}$$

!!! $\boldsymbol{C}$ est une matrice $t \times t$ à inverser. Si n est grand, t doit l'être aussi ...

Progrès espéré (EI)

EI = Expected Improvement , quantifie le compromis entre exploration et exploitation.



$$I(x) = max(f_{min} - Y(x), 0) \quad où \quad Y(x) = [\, F(x) \,|\, \vec{f} \,]$$

$$EI(x) = (f_{min} - m_{OK}(x))\, \Phi\left(\frac{f_{min} - m_{OK}(x)}{s_{OK}(x)}\right) + s_{OK}(x)\, \phi\left(\frac{f_{min} - m_{OK}(x)}{s_{OK}(x)}\right)$$

demo en cours

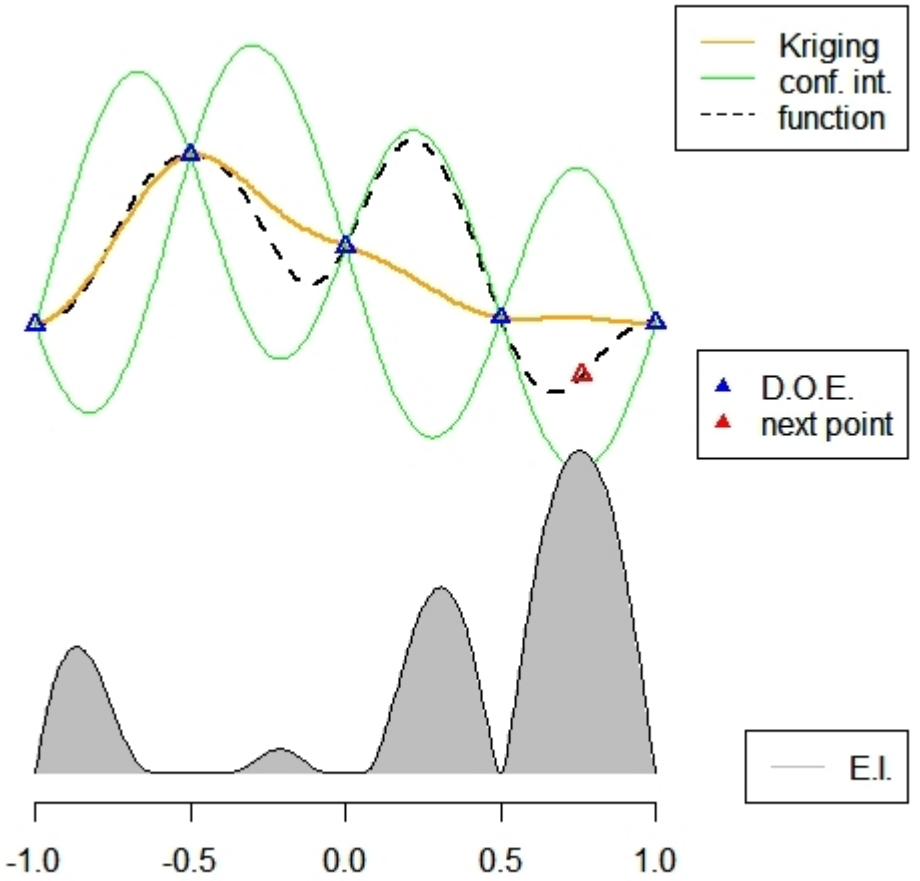
Ici :

· preuve de la formule de l'EI
· preuve de la monotonie de l'EI avec la moyenne et la variance
· que se passe-t-il quand var=0 ? EI(xi) = 0

Une itération d'EGO

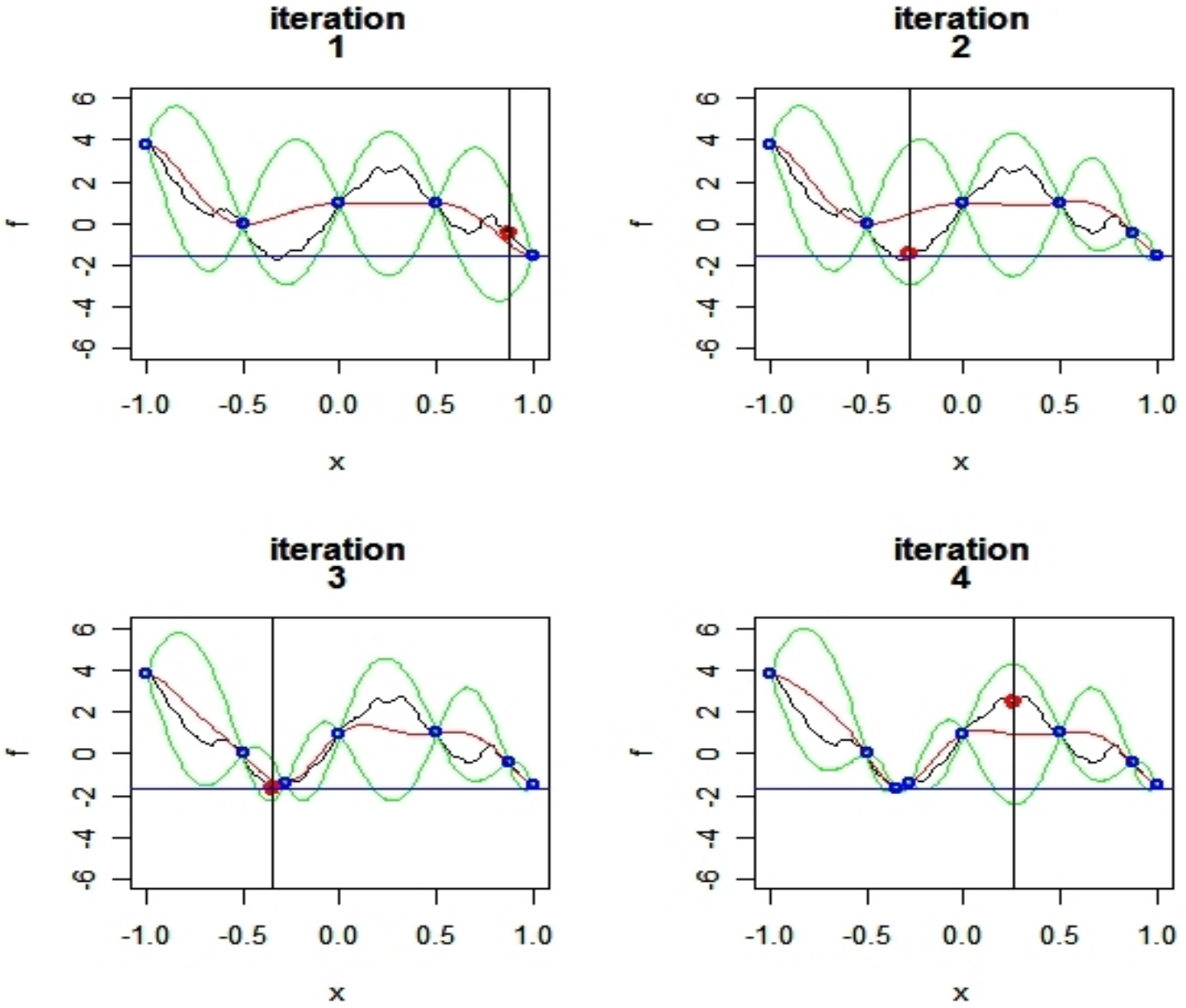
A chaque itération, EGO ajoute aux t points connus celui qui maximise EI,

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



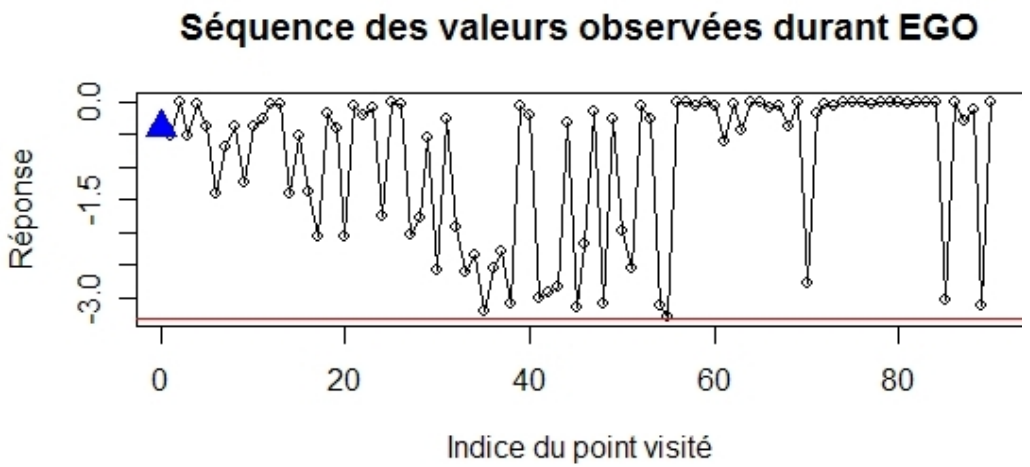
puis le krigeage est mis à jour ...

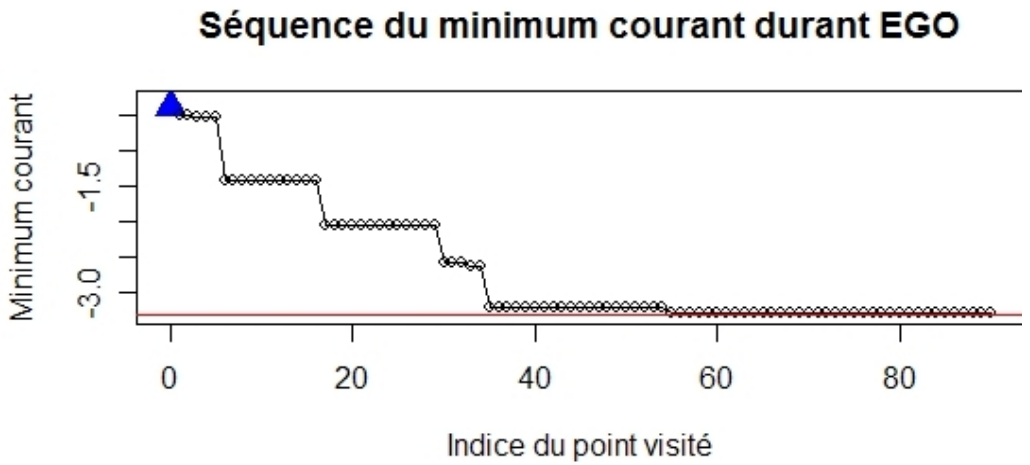
EGO : exemple

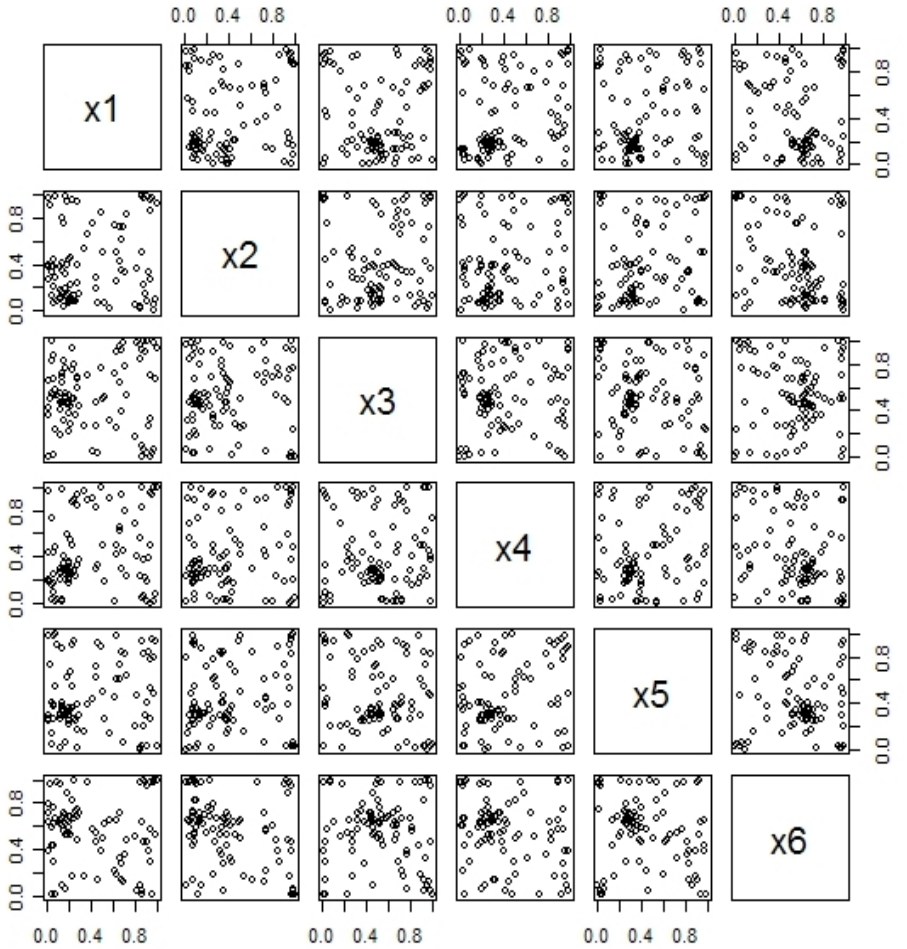

iteration
1
iteration
2
iteration
3
iteration
4
f
x
-6
-4
-2
0
2
4
6
-1.0
-0.5
0.0
0.5
1.0

EGO : exemple en 6D

Fonction de Hartman, $f(x^*)=-3.32$, 10 points dans le plan d'expérience initial.







(DiceOptim, D. Ginsbourger, 2009)

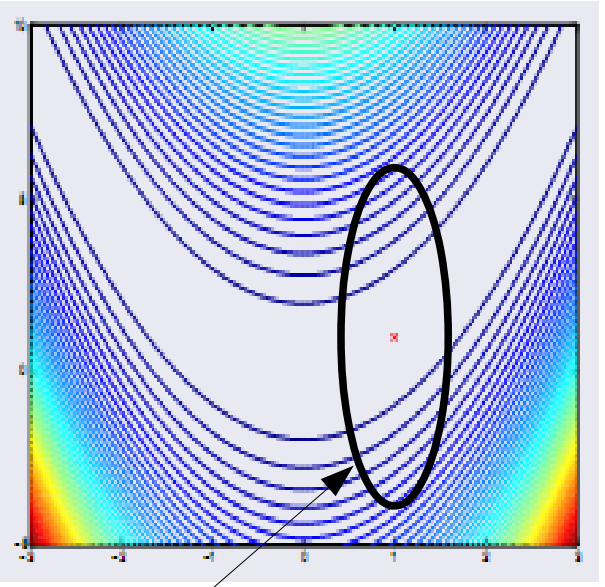
Discussion : EGO vs. CMA-ES

EGO a un métamodèle ← adapté aux fonctions coûteuses. CMA-ES peut appeler des millions de fois *f*, pas EGO (pb. d'inversion de mat. cov)

En présence d'une vallée étroite,

EGO raccourcit les portées et augmente σ, donc augmente s_{OK} et continue à affecter des ressources dans tout S,

CMA-ES adapte C pour la rendre proportionnelle à l'inverse du Hessien de f → la recherche devient plus locale, moins de ressources affectées loin de la vallée.



Iso-densité de probabilité de C correctement adaptée

→ EGO est plus global que CMA-ES (dont les versions les plus robustes sur les cas tests académiques utilisent le redémarrage avec population croissante) mais CMA-ES converge (précision) mieux. CMA-ES a été utilisé en hautes dimensions (de l'ordre de 100), EGO jusqu'en dimension 20. La robustesse d'EGO en dimension est un sujet de recherche (fonctions de covariance, krigeage additif).