Les mesures de risques en optimisation sous incertitudes

Romain Couderc

8 décembre 2021

Sommaire

1. Présentation du problème
2. Algorithme Robust-MADS
3. Algorithme ROBOBOA
4. L'algorithme ROSA
5. Comparaisons numériques

L'optimisation sous incertitudes par mesure de risque consiste à résoudre le problème suivant :

$$\min_{x \in \mathbb{R}^n} \ \Xi\left[f(x, \xi)\right] \tag{1}$$

avec :

- $f : \mathbb{R}^n \times \mathcal{S}_1 \to \mathcal{S}_2$ une boite noire bruitée,
- $\Xi : \mathcal{S}_2 \to \mathbb{R}$ une mesure de risque permettant de traiter les incertitudes $\xi \in \mathcal{S}_1$.

Par exemple :

- avec $\mathcal{S}_1 = \Omega$, $\mathcal{S}_2 = \mathcal{L}^1(\Omega, \mathcal{F}, \mathbb{P})$ et $\Xi[\cdot] = \mathbb{E}_\xi[\cdot]$ où $(\Omega, \mathcal{F}, \mathbb{P})$ défini un espace probabilisé, le problème (1) est un problème d'optimisation en espérance, dit "neutre au risque".
- avec $\mathcal{S}_1 = \mathcal{U} \subset \mathbb{R}^m$, $\mathcal{S}_2 = \mathcal{I}_\mathbb{R}$ et $\Xi[\cdot] = \max_{\xi \in \mathcal{U}}[\cdot]$, le problème (1) est un problème d'optimisation robuste, dit "du pire cas".

L'optimisation sous incertitudes par mesure de risque consiste à résoudre le problème suivant :

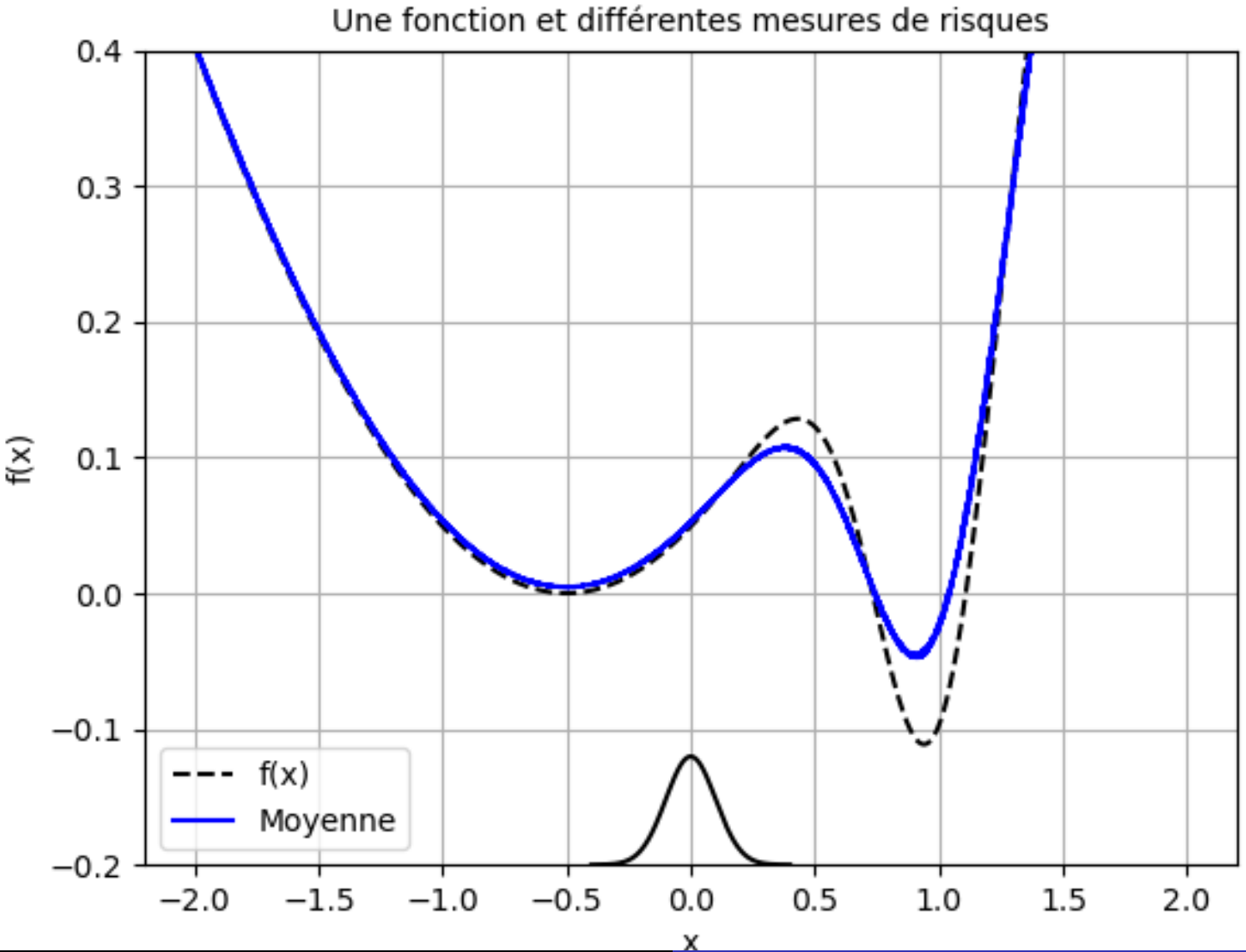
$$\min_{x \in \mathbb{R}^n} \ \Xi\left[f(x, \xi)\right] \tag{1}$$

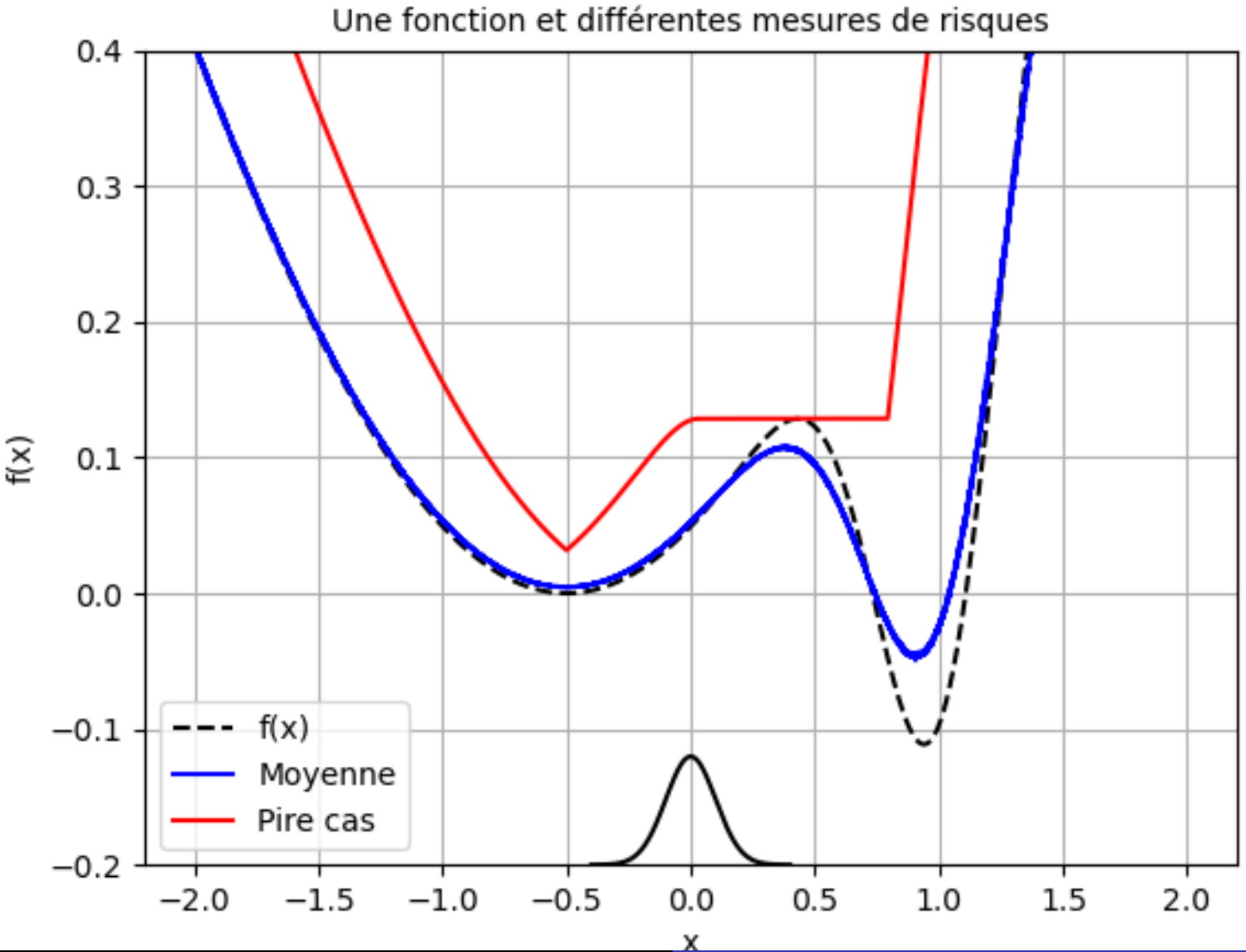
avec :

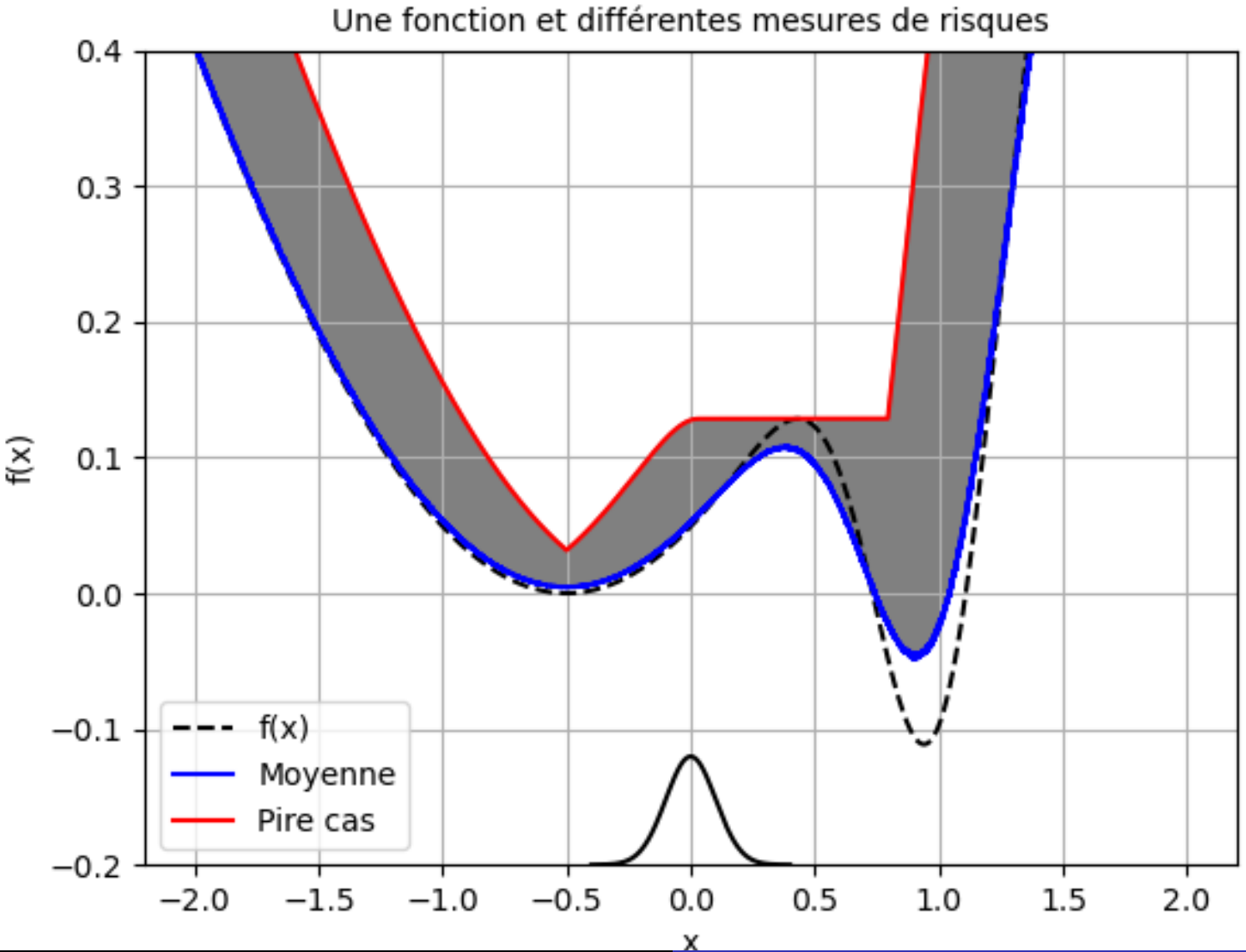
- $f : \mathbb{R}^n \times \mathcal{S}_1 \to \mathcal{S}_2$ une boite noire bruitée,
- $\Xi : \mathcal{S}_2 \to \mathbb{R}$ une mesure de risque permettant de traiter les incertitudes $\xi \in \mathcal{S}_1$.

Par exemple :

- avec $\mathcal{S}_1 = \Omega$, $\mathcal{S}_2 = \mathcal{L}^1(\Omega, \mathcal{F}, \mathbb{P})$ et $\Xi[\cdot] = \mathbb{E}_\xi[\cdot]$ où $(\Omega, \mathcal{F}, \mathbb{P})$ défini un espace probabilisé, le problème (1) est un problème d'optimisation en espérance, dit "neutre au risque".
- avec $\mathcal{S}_1 = \mathcal{U} \subset \mathbb{R}^m$, $\mathcal{S}_2 = \mathcal{I}_\mathbb{R}$ et $\Xi[\cdot] = \max_{\xi \in \mathcal{U}}[\cdot]$, le problème (1) est un problème d'optimisation robuste, dit "du pire cas".


Une fonction et différentes mesures de risques
f(x)
x
f(x)
Moyenne
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Une fonction et différentes mesures de risques
f(x)
x
f(x)
Moyenne
Pire cas
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Une fonction et différentes mesures de risques
f(x)
x
f(x)
Moyenne
Pire cas
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

Minimisation en espérance

L'algorithme Robust-MADS [1] cherche à résoudre le problème suivant :

$$\min_{x \in \mathbb{R}^n} \tilde{f}(x)$$

où

$$\tilde{f}(x) = \int_{\mathbb{R}^n} f(u) \exp\left(-\frac{||u-x||^2}{\sigma^2}\right) du$$

Remarque, si ξ un vecteur aléatoire Gaussien, on a :

$$\begin{aligned}
\mathbb{E}_\xi[f(x+\xi)] &= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{\mathbb{R}^n} f(x+\xi) \exp\left(-\frac{||\xi||^2}{\sigma^2}\right) d\xi \\
&= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{\mathbb{R}^n} f(u) \exp\left(-\frac{||u-x||^2}{\sigma^2}\right) du \\
&= \frac{\tilde{f}(x)}{\sqrt{2\pi\sigma^2}}
\end{aligned}$$

Estimation de $\tilde{f}$

Estimation de la fonction $\tilde{f}^l$, après l évaluations de fonction, par

$$\tilde{f}^l(x) = \frac{\sum_{v \in V^l} P(x,v) f(v)}{\sum_{v \in V^l} P(x,v)}$$

avec

$$P(x,v) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{||x-v||^2}{\sigma^2}\right)$$

Principe de l'algorithme

Soit X^k l'ensemble des candidats solutions à l'itération k par l'étape de sonde ou l'étape de recherche. Pour chaque candidat $x \in X^k$, la procédure suivante s'applique :

1. Mise à jour du cache : $V^l = V^{l-1} \cup \{x\}$.
2. Calcul de $f^l(v)$ pour tout $v \in V^l$.
3. Sélection du meilleur point $x^l \in \text{argmin}\{\tilde{f}^l(v) : v \in V^l\}$.
   - si $x^l = x$ alors on augmente la taille de la sonde.
   - si $x^l \neq x^{l-1}$ on garde la même taille de sonde.
   - Dans les deux cas, on passe à l'itération suivante.
4. Dans le cas où aucun point candidat ne fournit une amélioration du meilleur point à date, on diminue la taille de la sonde.

Résultat de convergence : x^* est la limite des optima locaux du treillis qui devient infiniment fin.

Optimisation dans le pire cas

L'algorihme ROBOBOA [2] cherche à résoudre le problème suivant :

$$\min_{x \in \mathbb{R}^n} \max_{u \in \mathcal{U}} f(x, u)$$

L'idée de l'algorithme est de résoudre ce problème par la méthode inexacte d'approximations successives ("Inexact method of outer approximation"), c'est à dire de résoudre successivement

$$\max_{u \in \mathcal{U}} f(\hat{x}, u) \text{ et } \min_{x \in \mathbb{R}^n} \max_{u \in \hat{\mathcal{U}} \subseteq \mathcal{U}} f(x, u)$$

Deux difficultés dans cette approche :

- le contexte d'optimisation sans dérivée $\rightarrow$ Méthode à région de confiance.
- la taille de l'ensemble $\hat{\mathcal{U}} \rightarrow$ "Manifold sampling".

Optimisation dans le pire cas

L'algorihme ROBOBOA [2] cherche à résoudre le problème suivant :

$$\min_{x \in \mathbb{R}^n} \max_{u \in \mathcal{U}} f(x, u)$$

L'idée de l'algorithme est de résoudre ce problème par la méthode inexacte d'approximations successives ("Inexact method of outer approximation"), c'est à dire de résoudre successivement

$$\max_{u \in \mathcal{U}} f(\hat{x}, u) \text{ et } \min_{x \in \mathbb{R}^n} \max_{u \in \hat{\mathcal{U}} \subseteq \mathcal{U}} f(x, u)$$

Deux difficultés dans cette approche :

- le contexte d'optimisation sans dérivée $\rightarrow$ Méthode à région de confiance.
- la taille de l'ensemble $\hat{\mathcal{U}} \rightarrow$ "Manifold sampling".

Optimisation dans le pire cas

L'algorihme ROBOBOA [2] cherche à résoudre le problème suivant :

$$\min_{x\in\mathbb{R}^n} \max_{u\in\mathcal{U}} f(x,u)$$

L'idée de l'algorithme est de résoudre ce problème par la méthode inexacte d'approximations successives ("Inexact method of outer approximation"), c'est à dire de résoudre successivement

$$\max_{u\in\mathcal{U}} f(\hat{x},u) \text{ et } \min_{x\in\mathbb{R}^n} \max_{u\in\hat{\mathcal{U}}\subseteq\mathcal{U}} f(x,u)$$

Deux difficultés dans cette approche :

- le contexte d'optimisation sans dérivée $\rightarrow$ Méthode à région de confiance.
- la taille de l'ensemble $\hat{\mathcal{U}} \rightarrow$ "Manifold sampling".

Résultat de convergence

Sous les conditions suivantes :

- f et ∇f sont localement Lipschitz continues.
- $\mathcal{U}$ est compact.
- Etant donné Δ, $x^k \in \mathbb{R}^n$ et $\mathcal{U}^k$, il existe $\kappa_f > 0$ et $\kappa_g > 0$ indépendant de Δ, x^k et $\mathcal{U}^k$ tel que $\forall u^j \in \mathcal{U}^k$

$$|f(x^k + s, u^j) - m_j^k(x^k + s)| \leq \kappa_f \Delta^2 \qquad \forall s \in \mathcal{B}(0, \Delta)$$
$$|\nabla f(x^k + s, u^j) - \nabla m_j^k(x^k + s)| \leq \kappa_f \Delta \qquad \forall s \in \mathcal{B}(0, \Delta).$$

Alors le point d'accumulation x^* généré par l'algorithme ROBOBOA est tel que

$$0 \in \partial \left(\max_{u \in \mathcal{U}} f(x^*, u) \right).$$

Optimisation adaptative

Soit f une fonction intégrable, le quantile de niveau α en x est défini comme :

$$t_x^*(\alpha) = \inf\{t \ : \ \mathbb{P}(f(x,\xi) \leq t) \geq 1-\alpha\}$$

On définit ensuite $CVaR_\alpha(f(x,\xi))$ comme étant :

$$\begin{aligned} CVaR_\alpha(f(x,\xi)) &= \mathbb{E}_\xi[f(x,\xi)|f(x,\xi) \geq t_x^*(\alpha)] \\ &= t_x^*(\alpha) + \tfrac{1}{\alpha}\mathbb{E}_\xi[(f(x,\xi) - t_x^*(\alpha))_+] \\ &= \inf_{t\in\mathbb{R}} \mathbb{E}_\xi[f(x,\xi) + (t - f(x,\xi))_+ + (\alpha^{-1}-1)(f(x,\xi)-t)_+]. \end{aligned}$$

L'algorithme ROSA cherche à résoudre le problème suivant :

$$\min_{(x,t)\in\mathcal{X}\times\mathbb{R}} \mathbb{E}_\xi[\underbrace{f(x,\xi) + (t - f(x,\xi))_+ + (\alpha^{-1}-1)(f(x,\xi)-t)_+}_{h_\alpha(x,t,\xi)}]$$

Optimisation adaptative

Soit f une fonction intégrable, le quantile de niveau α en x est défini comme :

$$t_x^*(\alpha) = \inf\{t \ : \ \mathbb{P}(f(x,\xi) \leq t) \geq 1-\alpha\}$$

On définit ensuite $CVaR_\alpha(f(x,\xi))$ comme étant :

$$\begin{aligned} CVaR_\alpha(f(x,\xi)) &= \mathbb{E}_\xi[f(x,\xi)|f(x,\xi) \geq t_x^*(\alpha)] \\ &= t_x^*(\alpha) + \tfrac{1}{\alpha}\mathbb{E}_\xi[(f(x,\xi) - t_x^*(\alpha))_+] \\ &= \inf_{t\in\mathbb{R}} \mathbb{E}_\xi[f(x,\xi) + (t - f(x,\xi))_+ + (\alpha^{-1}-1)(f(x,\xi)-t)_+]. \end{aligned}$$

L'algorithme ROSA cherche à résoudre le problème suivant :

$$\min_{(x,t)\in\mathcal{X}\times\mathbb{R}} \mathbb{E}_\xi[\underbrace{f(x,\xi) + (t - f(x,\xi))_+ + (\alpha^{-1}-1)(f(x,\xi)-t)_+}_{h_\alpha(x,t,\xi)}]$$

Principe de l'algorithme

L'algorithme est fondé sur la méthode d'approximation stochastique dont la mise à jour est la suivante :

$$x^{k+1} = x^k - a^k h_\alpha(\mathbf{x}^k, t^k, \xi^k)\xi_x^k$$

où la suite a^k est une suite scalaire positif vérifiant

$$\sum_k a^k = \infty \text{ et } \sum_k (a^k)^2 < \infty.$$

La suite des itérés va alors "traquer" le comportement asymptotique de l'équation différentielle suivante

$$\dot{x}(u) = -\nabla_x \mathbb{E}_\xi(h_\alpha(x(u), t(u), \xi))$$

qui converge vers l'ensemble défini par :

$$H_x = \{x : \nabla_x \mathbb{E}_\xi(h_\alpha(x(u), t(u), \xi) = 0\}$$

Résultat de convergence

Soit $\mathcal{X} \subseteq \mathbb{R}^n$ un ensemble compact et $(\Omega, \mathcal{F}, \mathbb{P})$ un espace probabilisé. On fait l'hypothèse que :

- Les vecteurs incertains ξ_x et ξ_p sont indépendants.
- Les fonctions de répartitions marginales de chaque variable de ξ_x sont connues.
- La fonction $f(x, \cdot)$ est bornée pour tout $x \in \mathcal{X}$.
- La fonction $f(x, \cdot)$ est $\mathcal{F}$-measurable pour tout $x \in \mathbb{R}^n$.

Alors les itérés x^k tendent vers K lorsque $k \to \infty$ où K est un sous ensemble compact des points d'équilibre asymptotiquement stable de l'ensemble H_x.

Les tests numériques sont conduits sur un ensemble de problème test de la littérature. Les éléments clés sont les suivants :

- Un problème test est une combinaison d'une fonction objectif bruité et d'un ensemble incertain particulier.
- Les ensembles incertains sont compacts pour pouvoir comparer les trois algorithmes.
- La taille des problème test varie de 2 à 100 pour la dimension de x et de n à $2n-2$ pour la dimension de ξ.
- Les résultats sont présentés à partir de profil de donnée dont le test de convergence est le suivant pour $\alpha \in (0,1]$:

$$t^*_{\mathbf{x}^0}(\alpha) - t^*_{\mathbf{x}^e}(\alpha) \geq (1-\tau)(t^*_{\mathbf{x}^0}(\alpha) - t^*_{\mathbf{x}^*}(\alpha))$$

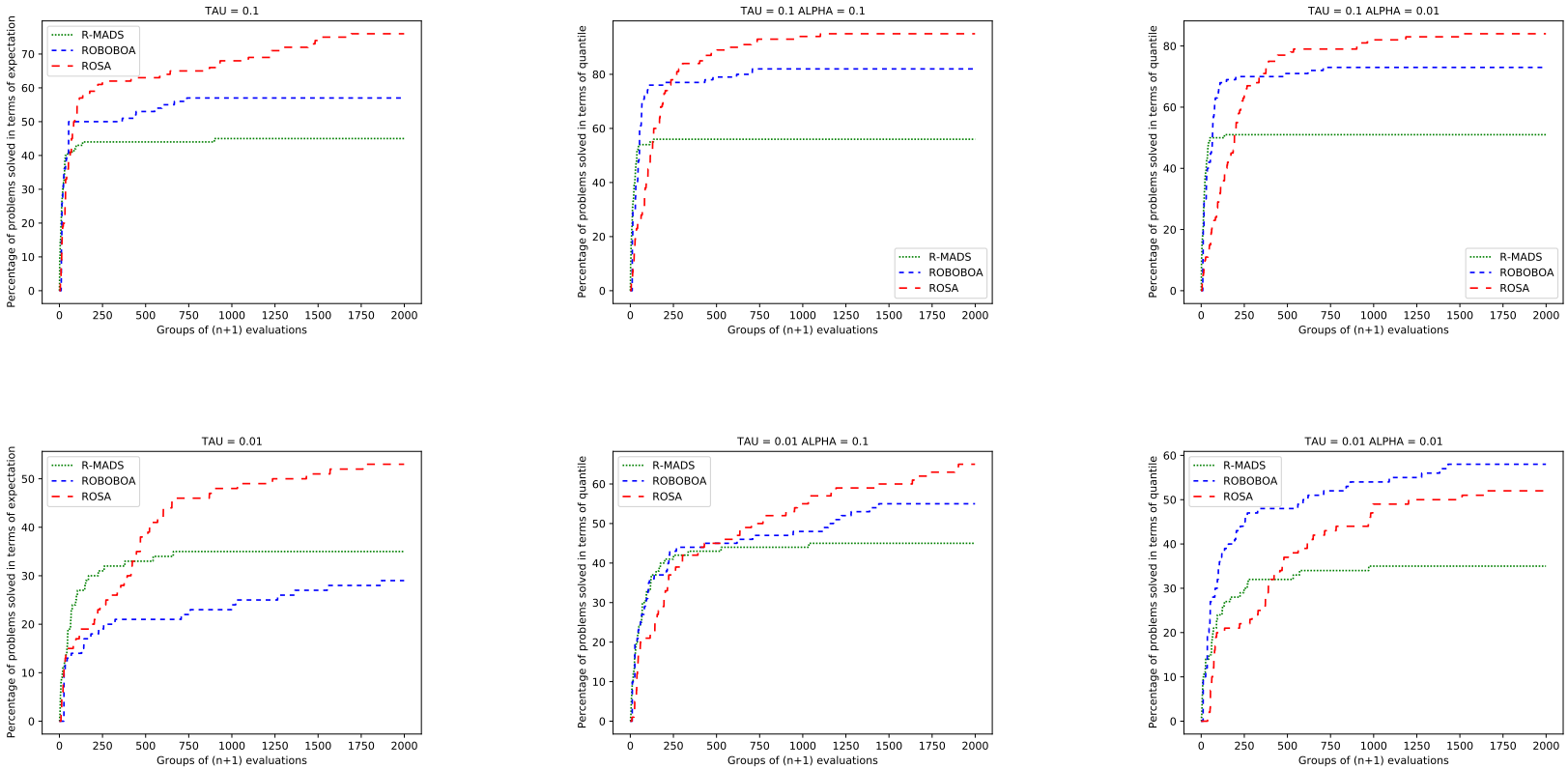


Figure – Résultats pour les problèmes test avec $n \leq 10$ for $\tau = 0.1$ (en haut) et $\tau = 0.01$ (en bas)

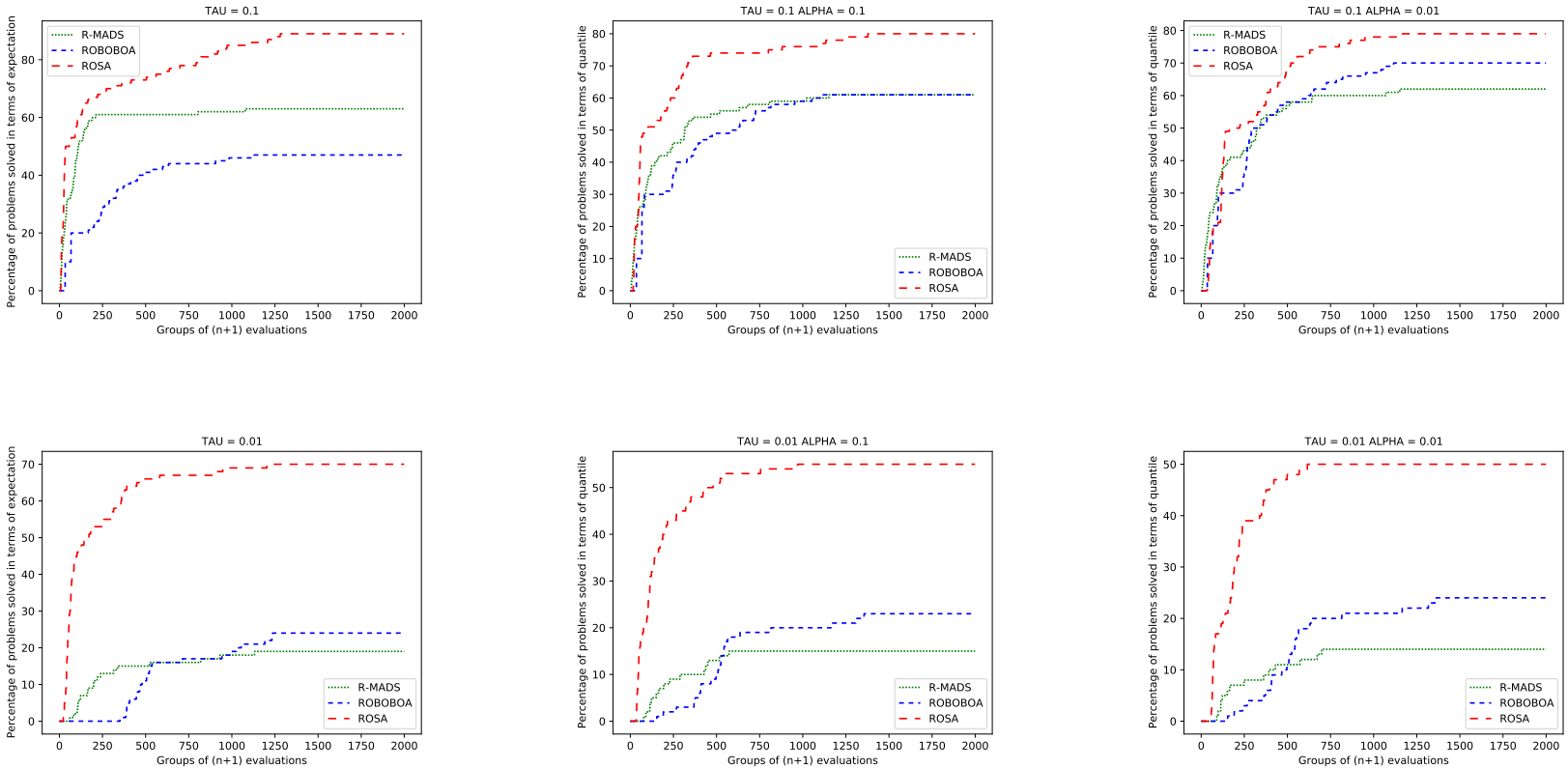


Figure – Résultats pour les problèmes test avec $n > 10$ for $\tau = 0.1$ (en haut) et $\tau = 0.01$ (en bas)

[1] C. Audet, A. Ihaddadene, S. Le Digabel, and C. Tribes. Robust optimization of noisy blackbox problems using the Mesh Adaptive Direct Search algorithm. *Optimization Letters*, 12(4) :675–689, 2018.

[2] Matt Menickelly and Stefan M Wild. Derivative-free robust optimization by outer approximations. *Mathematical Programming*, 179(1) :157–193, 2020.