




Risk-adverse optimization by Conditional Value-at-Risk and stochastic approximation

Charles Audet, Jean Bigeon, Romain Couderc & Michael Kokkolaras

18/05/2022

Contents

- 1 Problem setting
- 2 Expectation minimization  the Robust-MADS algorithm
- 3 Worst-case optimization : the ROBOBOA algorithm
- 4 Risk-adverse optimization : the ROSA algorithm
- 5 Numerical results

Risk adverse optimization aims to solve the following problem :

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

with :

- $f : \mathbb{R}^n \times \mathcal{S}_1 \rightarrow \mathcal{S}_2$ a noisy blackbox,
- $\Xi : \mathcal{S}_2 \rightarrow \mathbb{R}$ a risk measure to handle the uncertainties $\xi \in \mathcal{S}_1$.

For instance :

- with $\mathcal{S}_1 = \Omega$, $\mathcal{S}_2 = \mathcal{L}^1(\Omega, \mathcal{F}, \mathbb{P})$ and $\Xi[\cdot] = \mathbb{E}_\xi[\cdot]$ where $(\Omega, \mathcal{F}, \mathbb{P})$ defined a probability space, problem (1) is a problem of expectation minimization, said "risk-free".
- with $\mathcal{S}_1 = \mathcal{U} \subset \mathbb{R}^m$, $\mathcal{S}_2 = \mathcal{C}(\mathcal{U})$ and $\Xi[\cdot] = \max_{\xi \in \mathcal{U}}[\cdot]$, problem (1) is a robust optimization problem, said "worst-case".

Risk adverse optimization aims to solve the following problem :

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

with :

- $f : \mathbb{R}^n \times \mathcal{S}_1 \rightarrow \mathcal{S}_2$ a noisy blackbox,
- $\Xi : \mathcal{S}_2 \rightarrow \mathbb{R}$ a risk measure to handle the uncertainties $\xi \in \mathcal{S}_1$.

For instance :

- with $\mathcal{S}_1 = \Omega$, $\mathcal{S}_2 = \mathcal{L}^1(\Omega, \mathcal{F}, \mathbb{P})$ and $\Xi[\cdot] = \mathbb{E}_\xi[\cdot]$ where $(\Omega, \mathcal{F}, \mathbb{P})$ defined a probability space, problem (1) is a problem of expectation minimization, said "risk-free".
- with $\mathcal{S}_1 = \mathcal{U} \subset \mathbb{R}^m$, $\mathcal{S}_2 = \mathcal{C}(\mathcal{U})$ and $\Xi[\cdot] = \max_{\xi \in \mathcal{U}}[\cdot]$, problem (1) is a robust optimization problem, said "worst-case".

Problem setting

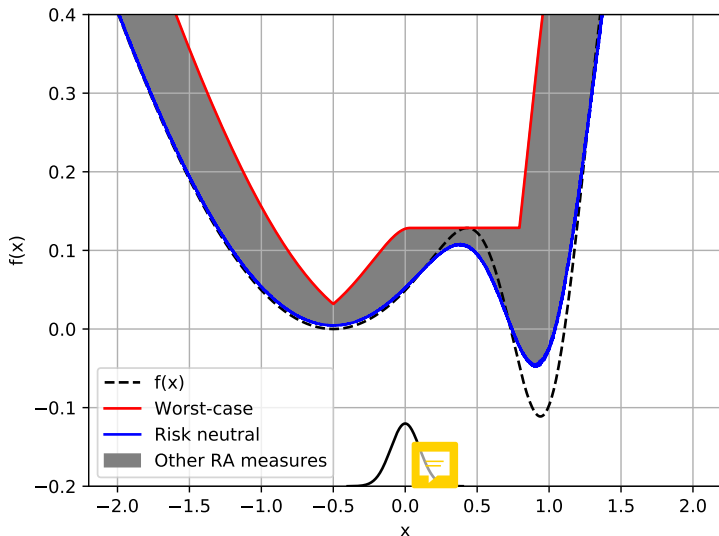
Expectation minimization : the Robust-MADS algorithm

Worst-case optimization : the ROBOBOA algorithm

Risk-adverse optimization : the ROSA algorithm

Numerical results

Références



Expectation minimization

The Robust-MADS algorithm [1] solves the following problem :

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

where

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

Note that if ξ is a Gaussian random vector, then :

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

Convergence : x^* is the limit of the mesh local optima which becomes infinitely thin.

Expectation minimization

The Robust-MADS algorithm [1] solves the following problem :

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

where

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

Note that if ξ is a Gaussian random vector, then :

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

Convergence : x^* is the limit of the mesh local optima which becomes infinitely thin.

Worst-case optimization

The ROBOBOA algorithm [2] solves the following problem :

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

The idea is to use the inexact method of outer approximation, that means solve successively the problems :

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

Two issues in this approach :

- Use a derivative free algorithm \rightarrow Trust region method.
- the size of the set $\hat{\mathcal{U}} \rightarrow$ Manifold sampling.

Worst-case optimization

The ROBOBOA algorithm [2] solves the following problem :

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

The idea is to use the inexact method of outer approximation, that means solve successively the problems :

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

Two issues in this approach :

- Use a derivative free algorithm \rightarrow Trust region method.
- the size of the set $\hat{\mathcal{U}} \rightarrow$ Manifold sampling.

Worst-case optimization

The ROBOBOA algorithm [2] solves the following problem :

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

The idea is to use the inexact method of outer approximation, that means solve successively the problems :

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

Two issues in this approach :

- Use a derivative free algorithm \rightarrow Trust region method.
- the size of the set $\hat{\mathcal{U}} \rightarrow$ Manifold sampling.

Convergence

Under the following conditions :

- f and ∇f are locally Lipschitz continuous.
- \mathcal{U} is a compact set.
- Since $\Delta, x^k \in \mathbb{R}^n$ and \mathcal{U}^k , there exist $\kappa_f > 0$ and $\kappa_g > 0$ independent from Δ, x^k and \mathcal{U}^k such that $\forall u^j \in \mathcal{U}^k$

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

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

Then the accumulation point x^* generated by the ROBOBOA algorithm is such that :

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

Adaptive risk

Let f be an intergable function, the quantile of level α at x is defined by :

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

Then, $CVaR_\alpha(f(x, \xi))$ is :

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

The ROSA algorithm [3] solves the following problem :

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

Adaptive risk

Let f be an intergable function, the quantile of level α at x is defined by :

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

Then, $CVaR_\alpha(f(x, \xi))$ is :

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

The ROSA algorithm [3] solves the following problem :

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

Algorithm process

The algorithm is based on Gaussian smoothing [4] and stochastic approximation. Basically, the update is the following

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

where the step sizes a^k is a sequence of positive scalar such that

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

The sequence of iterates x^k is going to track the asymptotic behavior of the following ordinary differential equation :

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

which converges to the set :

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

Convergence

Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a compact set and $(\Omega, \mathcal{F}, \mathbb{P})$ a probabilistic space. Under the assumptions that :

- the uncertain vectors ξ_x et ξ_p are independent.
- The marginal distribution of each variable of ξ_x are known.
- The function $f(x, \cdot)$ is bounded for all $x \in \mathcal{X}$.
- The function $f(x, \cdot)$ is \mathcal{F} -measurable for all $x \in \mathbb{R}^n$.

Then, the limit of the iterates x^k when $k \rightarrow \infty$ are stationary points of $\mathbf{E}_\xi(h_\alpha(x, t, \xi))$, with probability one.

The numerical results are conducted on 20 test problems from the literature. The key settings are :

- Each problem test is a combination of a noisy objective function and of an uncertainty set.
- The uncertainty set is compact in order to compare the three algorithms.
- The size of the problems is from 2 to 100 for the dimension along to x and from 2 to 198 for the dimension along to ξ .
- The results are presented under data profiles where the convergence test is for an accuracy of $\tau > 0$ and an $\alpha \in (0, 1]$ is :

$$t_{\mathbf{x}^0}^*(\alpha) - t_{\mathbf{x}^\varepsilon}^*(\alpha) \geq (1 - \tau)(t_{\mathbf{x}^0}^*(\alpha) - t_{\mathbf{x}^*}^*(\alpha))$$

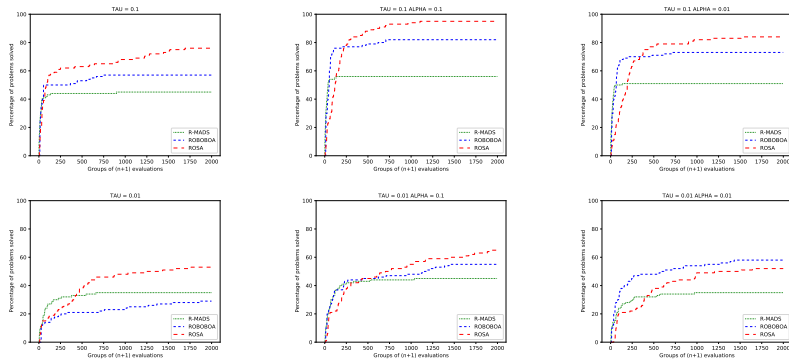


Figure – Results for test problem such that $n \leq 10$ for $\tau = 0.1$ (above) and $\tau = 0.01$ (below)

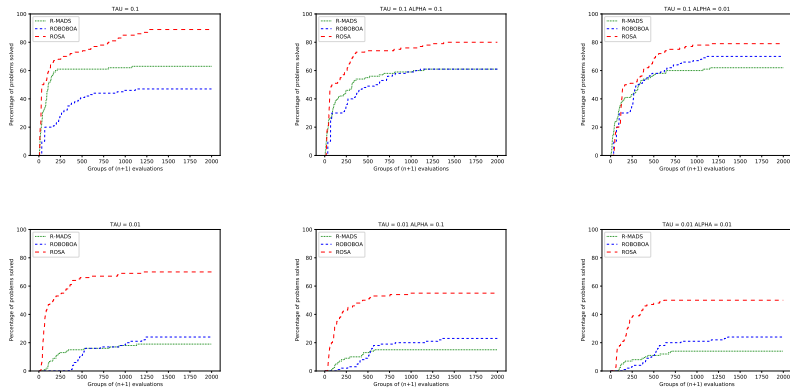


Figure – Results for problem tests such that $n > 10$ for $\tau = 0.1$ (above) and $\tau = 0.01$ (below)

- [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] M. Menickelly and S. M. Wild. Derivative-free robust optimization by outer approximations. *Mathematical Programming*, 179(1) :157–193, 2020.
- [3] C. Audet, J. Bigeon, R. Couderc, and M. Kokkolaras. Risk-adverse optimization by conditional value-at-risk and stochastic approximation. Technical report, Groupe d'études et de recherche en analyse des décisions, GERAD, Montréal QC H3T 2A7, Canada, February 2022.
- [4] S Bhatnagar, H Prasad, and L Prashanth. Stochastic approximation algorithms. In *Stochastic Recursive Algorithms for Optimization*, pages 17–28. Springer, 2013.