Risk-averse stochastic programming and distributionally robust optimization via operator splitting

Welington de Oliveira
www.oliveira.mat.br

MINES ParisTech, CMA - Centre de Mathématiques Appliquées

Workshop on robust and stochastic optimization methods
CERMICS, November 19th, 2021

We are interested in convex optimization problems of the form

$$\min_{x\in\mathbb{R}^{nS}} \ \rho\left(\mathfrak{f}_1(x_1),\ldots,\mathfrak{f}_S(x_S)\right) \quad \text{s.t.} \quad x\in\mathcal{N} \qquad (\texttt{Pbm})$$

- $x_s \in \mathbb{R}^n$ is the decision vector related to scenario $s = 1,\ldots,S$
- $x = (x_1,\ldots,x_S) \in \mathbb{R}^{nS}$ comprises all the decision variables: decision policy
- $\mathcal{N} \subset \mathbb{R}^{nS}$ is a linear space of nonanticipative policies
- $\mathfrak{f}_s : \mathbb{R}^n \to \mathbb{R}\cup+\infty$ is the scenario cost function, which we assume lsc and convex (e.g. $\mathfrak{f}_s(\cdot) := c_s(\cdot) + \mathbf{i}_{X_s}(\cdot)$)
- $\rho : \mathbb{R}^S \to \mathbb{R}$ is a convex and monotonically non-decreasing function, that is,

$$\rho(\lambda v + (1-\lambda)v') \leq \lambda\rho(v) + (1-\lambda)\rho(v') \quad \forall v, v' \in \mathbb{R}^S,\ \lambda\in[0,1], \quad \text{and}$$

$$\rho(v) \geq \rho(v') \quad \text{whenever} \quad v \geq v'$$

Centre de Mathématiques Appliquées

- Risk-neutral setting: given a probability vector $p = (p_1, \ldots, p_S)$,

$$\rho\left(\mathfrak{f}_1(x_1), \ldots, \mathfrak{f}_S(x_S)\right) = \mathbb{E}_p[\mathfrak{f}_s(x_s)] = \sum_{s=1}^{S} p_s \mathfrak{f}_s(x_s)$$

Centre de Mathématiques Appliquées

- Risk-neutral setting: given a probability vector $p = (p_1, \dots, p_S)$,

$$\rho\left(\mathfrak{f}_1(x_1), \dots, \mathfrak{f}_S(x_S)\right) = \mathbb{E}_p[\mathfrak{f}_s(x_s)] = \sum_{s=1}^{S} p_s \mathfrak{f}_s(x_s)$$

- Risk-averse setting: ρ is a risk measure. Example: given a regret function $Z : \mathbb{R}^S \to \mathbb{R}$ which is convex, monotone and positively homogeneous

$$\rho(v) = \min_{\tau \in \mathbb{R}} \tau + Z(v - \tau)$$

  Expectation case: $Z(v) = \sum_{s=1}^{S} p_s \zeta(v_s)$ and $\zeta : \mathbb{R} \to \mathbb{R}$ is a convex and non-decreasing function

  - Conditional Value-at-Risk: $\zeta(\cdot) = \frac{1}{1-\alpha}[\cdot]_+$ and $\alpha \in (0,1)$
  - Log-Exponential: $\zeta(\cdot) = \exp(\cdot) - 1 \quad (\Rightarrow \rho(v) = \log(\sum_{s=1}^{S} p_s[\exp(v_s)]))$

- Risk-neutral setting: given a probability vector $p = (p_1, \dots, p_S)$,

$$\rho\left(\mathfrak{f}_1(x_1), \dots, \mathfrak{f}_S(x_S)\right) = \mathbb{E}_p[\mathfrak{f}_s(x_s)] = \sum_{s=1}^{S} p_s \mathfrak{f}_s(x_s)$$

- Risk-averse setting: ρ is a risk measure. Example: given a regret function $Z : \mathbb{R}^S \to \mathbb{R}$ which is convex, monotone and positively homogeneous

$$\rho(v) = \min_{\tau \in \mathbb{R}} \tau + Z(v - \tau)$$

  Expectation case: $Z(v) = \sum_{s=1}^{S} p_s \zeta(v_s)$ and $\zeta : \mathbb{R} \to \mathbb{R}$ is a convex and non-decreasing function

  - Conditional Value-at-Risk: $\zeta(\cdot) = \frac{1}{1-\alpha}[\cdot]_+$ and $\alpha \in (0, 1)$
  - Log-Exponential: $\zeta(\cdot) = \exp(\cdot) - 1 \quad (\Rightarrow \rho(v) = \log(\sum_{s=1}^{S} p_s[\exp(v_s)]))$

- Worst-case scenario (robust optimization):

$$\rho(v) = \max_{s=1,\dots,S} v_s$$

- Risk-neutral setting: given a probability vector $p = (p_1, \dots, p_S)$,

$$\rho\left(\mathfrak{f}_1(x_1), \dots, \mathfrak{f}_S(x_S)\right) = \mathbb{E}_p[\mathfrak{f}_s(x_s)] = \sum_{s=1}^{S} p_s \mathfrak{f}_s(x_s)$$

- Risk-averse setting: ρ is a risk measure. Example: given a regret function $Z : \mathbb{R}^S \to \mathbb{R}$ which is convex, monotone and positively homogeneous

$$\rho(v) = \min_{\tau \in \mathbb{R}} \tau + Z(v - \tau)$$

Expectation case: $Z(v) = \sum_{s=1}^{S} p_s \zeta(v_s)$ and $\zeta : \mathbb{R} \to \mathbb{R}$ is a convex and non-decreasing function

  - Conditional Value-at-Risk: $\zeta(\cdot) = \frac{1}{1-\alpha}[\cdot]_+$ and $\alpha \in (0, 1)$
  - Log-Exponential: $\zeta(\cdot) = \exp(\cdot) - 1$ $\quad (\Rightarrow \rho(v) = \log(\sum_{s=1}^{S} p_s[\exp(v_s)]))$

- Worst-case scenario (robust optimization):

$$\rho(v) = \max_{s=1,\dots,S} v_s$$

- Distributionally robust optimization (fixed support):

$$\rho(v) = \max_{p \in \mathcal{P}} \mathbb{E}_p[v] = \max_{p \in \mathcal{P}} \langle p, v \rangle$$

Centre de Mathématiques Appliquées

Distributionally robust optimization

$$\rho(v) = \max_{p \in \mathcal{P}} \mathbb{E}_p[v]$$

- If $\mathcal{P}$ is a singleton, then we are back to the risk-neutral setting
- If the ambiguity set is the simplex, i.e. $\mathcal{P} = \Delta_S := \{y \in \mathbb{R}^S_+ : \sum_{s=1}^S y_s = 1\}$, then ρ above boils down to the worst-case scenario setting
- The connection between DRO and stochastic programs with coherent-risk measures is made by the Fenchel conjugate function

$$\rho(v) = \max_{y \in \mathcal{D}\mathrm{om}(\rho^*)} \langle y, v \rangle$$

- For the CVaR_α function

$$\rho(v) = \min_{\tau \in \mathbb{R}} \tau + \sum_{s=1}^{S} p_s \frac{1}{1-\alpha} [v_s - \tau]_+,$$

we have that $\mathcal{P}_{\mathrm{CVaR}_\alpha} = \mathcal{D}\mathrm{om}(\rho^*) = \left\{ q \in \mathbb{R}^S_+ : \sum_{s=1}^S q_s = 1,\ q \leq \frac{p}{(1-\alpha)} \right\}$ and thus

$$\rho(v) = \max_{p \in \mathcal{P}_{\mathrm{CVaR}_\alpha}} \mathbb{E}_p[v]$$

Function ρ is decomposable over scenarios

$$\min_{x \in \mathbb{R}^{nS}} \rho\left(\mathfrak{f}_1(x_1), \ldots, \mathfrak{f}_S(x_S)\right) \quad \text{s.t.} \quad x \in \mathcal{N} \qquad \texttt{(Pbm)}$$

- A well-known optimization tool for solving **(Pbm)** in the risk-neutral setting

$$\rho(v) = \rho\left(\mathfrak{f}_1(x_1), \ldots, \mathfrak{f}_S(x_S)\right) = \sum_{s=1}^{S} p_s \mathfrak{f}_s(x_s)$$

  is the Progressive Hedging Algorithm (PHA)[1]

- The PHA has been extended recently to handle risk-averse problems with ρ fitting the expectation setting[2]

$$\rho(v) = \min_{\tau \in \mathbb{R}} \tau + \sum_{s=1}^{S} p_s [\zeta(v_s - \tau)]$$

- In these two settings, the objective function of **(Pbm)** has an additive structure over scenarios and only the nonanticipativity constraint ($x \in \mathcal{N}$) couples the variables

[1] Rockafellar, R.T., Wets, R.J.B.: Scenarios and policy aggregation in optimization under uncertainty. MOR, 1991

[2] Rockafellar, R.T.: Solving stochastic programming problems with risk measures by progressive hedging. SVAA, 2018

General setting

$$\min_{x \in \mathbb{R}^{nS}} \rho\left(\mathfrak{f}_1(x_1), \ldots, \mathfrak{f}_S(x_S)\right) \quad \text{s.t.} \quad x \in \mathcal{N} \qquad \texttt{(Pbm)}$$

- If ρ **does not** have an additive structure over scenarios, then `(Pbm)` is not only coupled by the nonanticipativity constraint but also by the objective function
- PHA does not apply in such a more general setting...
- How can we decompose the problem?

Centre de Mathématiques Appliquées

Reformulating the problem

$$\min_{x \in \mathbb{R}^{nS}} \rho\left(\mathfrak{f}_1(x_1), \ldots, \mathfrak{f}_S(x_S)\right) \quad \text{s.t.} \quad x \in \mathcal{N} \qquad \texttt{(Pbm)}$$

- With the help of an auxiliary vector $u \in \mathbb{R}^n$, let us move the random cost functions to the constraints

$$\begin{cases} \min\limits_{x,u} & \rho(u) \\ \text{s.t.} & \mathfrak{f}_s(x_s) \leq u_s \quad \forall\, s = 1, \ldots, S \\ & x \in \mathcal{N} \end{cases}$$

This does not help much because u_s links f_s to ρ...

Reformulating the problem

$$\min_{x \in \mathbb{R}^{nS}} \rho\left(\mathfrak{f}_1(x_1), \ldots, \mathfrak{f}_S(x_S)\right) \quad \text{s.t.} \quad x \in \mathcal{N} \qquad \texttt{(Pbm)}$$

- With the help of an auxiliary vector $u \in \mathbb{R}^n$, let us move the random cost functions to the constraints

$$\begin{cases} \min\limits_{x,u} & \rho(u) \\ \text{s.t.} & \mathfrak{f}_s(x_s) \leq u_s \quad \forall\, s = 1, \ldots, S \\ & x \in \mathcal{N} \end{cases}$$

  This does not help much because u_s links f_s to ρ...

- Let us add another auxiliary vector $v \in \mathbb{R}^n$

$$\begin{cases} \min\limits_{x,u,v} & \rho(\color{red}{v}) \\ \text{s.t.} & \mathfrak{f}_s(x_s) \leq u_s \quad \forall\, s = 1, \ldots, S \\ & x \in \mathcal{N}, \ \color{red}{u = v} \end{cases}$$

Reformulating the problem

$$\min_{x \in \mathbb{R}^{nS}} \rho\left(\mathfrak{f}_1(x_1), \ldots, \mathfrak{f}_S(x_S)\right) \quad \text{s.t.} \quad x \in \mathcal{N} \qquad (\texttt{Pbm})$$

- With the help of an auxiliary vector $u \in \mathbb{R}^n$, let us move the random cost functions to the constraints

$$\begin{cases} \min_{x,u} & \rho(u) \\ \text{s.t.} & \mathfrak{f}_s(x_s) \leq u_s \quad \forall\, s = 1, \ldots, S \\ & x \in \mathcal{N} \end{cases}$$

  This does not help much because u_s links f_s to ρ...

- Let us add another auxiliary vector $v \in \mathbb{R}^n$

$$\begin{cases} \min_{x,u,v} & \rho(v) \\ \text{s.t.} & (x_s, u_s) \in \texttt{epi}\, \mathfrak{f}_s \quad \forall\, s = 1, \ldots, S \\ & x \in \mathcal{N},\ u = v \end{cases}$$

Reformulating the problem

$$\min_{x \in \mathbb{R}^{nS}} \ \rho\left(\mathfrak{f}_1(x_1), \dots, \mathfrak{f}_S(x_S)\right) \quad \text{s.t.} \quad x \in \mathcal{N} \qquad \texttt{(Pbm)}$$

By writing `(Pbm)` as

$$\begin{cases} \min\limits_{x,u,v} & \rho(v) \\ \text{s.t.} & (x_s, u_s) \in \mathbf{epi}\, \mathfrak{f}_s \quad \forall\, s = 1, \dots, S \\ & x \in \mathcal{N},\ u = v, \end{cases}$$

we can go further and obtain the following equivalent problem

$$\min_{\mathbf{x} \in \mathfrak{L}} G(\mathbf{x})$$

- dimension $\mathbf{n} := (n+2)S$
- $\mathbf{x} := ((x_1, u_1), \dots, (x_S, u_S), v_1, \dots, v_S) \in \mathbb{R}^{\mathbf{n}}$
- $\mathfrak{L} := \{\mathbf{x} \in \mathbb{R}^{\mathbf{n}} : \ x \in \mathcal{N}, \ \text{ and } \ u = v\}$ is a linear subspace of $\mathbb{R}^{\mathbf{n}}$
- $G(\mathbf{x}) := \sum_{s=1}^{S} \mathbf{i}_{\mathbf{epi}\, \mathfrak{f}_s}(x_s, u_s) + \rho(v)$ is a convex function

$\boxed{\text{The new objective function is now decomposable!}}$

Centre de Mathématiques Appliquées

The Douglas-Rachford splitting method

$$\text{(Pbm)} \qquad \min_{x \in \mathcal{N}} \rho\left(\mathfrak{f}_1(x_1), \ldots, \mathfrak{f}_S(x_S)\right) \quad \equiv \quad \min_{\mathbf{x} \in \mathfrak{L}} G(\mathbf{x})$$

Optimality condition (under a constraint qualification)

$$\text{find } \bar{\mathbf{x}} \text{ s.t. } 0 \in \partial G(\bar{\mathbf{x}}) + \partial \mathbf{i}_{\mathfrak{L}}(\bar{\mathbf{x}})$$

- Solving (Pbm) amounts to finding a zero of the sum of two maximal monotone operators
- Such a task can be accomplished by the well-known Douglas-Rachford splitting method (DR)[3]
- Given $r > 0$ and $\mathbf{z}^0 \in \mathbb{R}^{\mathbf{n}}$, set $k = 0$ and perform the following steps[4]:

$$\begin{cases} \mathbf{x}^k & = & \mathtt{Proj}_{\mathfrak{L}}(\mathbf{z}^k) \\ \hat{\mathbf{x}}^{k+1} & = & \mathtt{Prox}_{\frac{G}{r}}(2\mathbf{x}^k - \mathbf{z}^k) \\ \mathbf{z}^{k+1} & = & \mathbf{z}^k + [\hat{\mathbf{x}}^{k+1} - \mathbf{x}^k] \end{cases}$$

[3] Douglas, J., Rachford, H.H.: On the numerical solution of heat conduction problems in two and three space variables. Trans. Am. Math. (1956)

[4] $\mathtt{Proj}_X(y) := \arg\min_{x \in X} \|x - y\|^2$ and $\mathtt{Prox}_f(y) := \arg\min_{x \in \mathbb{R}^n} f(x) + \frac{1}{2}\|x - y\|^2$

CMA Centre de Mathématiques Appliquées

Initialization. Let $z_x^0 \in \mathbb{R}^{nS}$, $z_u^0 \in \mathbb{R}^S$, $z_v^0 \in \mathbb{R}^S$, and $r > 0$ be given. Set $k := 0$

Step 1. Define

$$x^k := \mathtt{Proj}_{\mathcal{N}}(z_x^k) \qquad \text{and} \qquad u^k := \frac{z_u^k + z_v^k}{2}$$

Step 2. Compute (in parallel) the auxiliary vectors

$$\hat{v}^{k+1} := \mathtt{Prox}_{\frac{\rho}{r}}(2u^k - z_v^k)$$

$$(\hat{x}_s^{k+1}, \hat{u}_s^{k+1}) := \mathtt{Proj}_{\mathtt{epi}\,\mathfrak{f}_s}\left[\left(2x_s^k - z_{x_s}^k, 2u_s^k - z_{u_s}^k\right)\right] \ \forall\, s = 1, \dots, S$$

Step 3. Update

$$z_x^{k+1} := z_x^k + \hat{x}^{k+1} - x^k$$
$$z_u^{k+1} := z_u^k + \hat{u}^{k+1} - u^k$$
$$z_v^{k+1} := z_v^k + \hat{v}^{k+1} - u^k$$

Set $k := k + 1$ and go back to Step 1

Step 1 computes the projection onto the nonanticipativity space $\mathcal{N}$, which is a straightforward operation

To see that, let

$$\Lambda(s,t) := \{j \in \{1,\dots,S\} : \xi^j_{[t]} = \xi^s_{[t]}\}, \quad \text{for all} \ \ t = 1,\dots,T \ \ \text{and} \ \ s = 1,\dots,S,$$

be the index set of all scenarios sharing the same history $\xi_{[t]} = (\xi^s_1, \dots, \xi^s_t)$

Then

$$x^k = (x^k_{1,1}, \dots, x^k_{T,1}, \dots, x^k_{1,S}, \dots, x^k_{T,S}) = \texttt{Proj}_{\mathcal{N}}(z^k_x)$$

is given by

$$x^k_{t,s} := \frac{1}{|\Lambda(s,t)|} \sum_{j \in \Lambda(s,t)} z^k_{x_{t,j}}, \quad \text{for all} \ \ t = 1,\dots,T \ \ \text{and} \ \ s = 1,\dots,S$$

Comments on SDAP - Step 2

Let the random cost mapping be $\mathfrak{f}_s(\cdot) = c_s(\cdot) + \mathbf{i}_{X_s}(\cdot)$ (convex)

The projection subproblem in Step 2

$$(\hat{x}_s^{k+1}, \hat{u}_s^{k+1}) := \texttt{Proj}_{\texttt{epi}\, \mathfrak{f}_s} \left[\left(2x_s^k - z_{x_s}^k, 2u_s^k - z_{u_s}^k \right) \right]$$

reads as

$$(\texttt{SDAP}) \qquad \begin{cases} \min\limits_{x_s, u_s} & \left\| x_s - \left(2x_s^k - z_{x_s}^k\right) \right\|^2 + \left[u_s - \left(2u_s^k - z_{u_s}^k \right) \right]^2 \\ \text{s.t.} & c_s(x_s) \le u_s, \quad x_s \in X_s \end{cases}$$

Comments on SDAP - Step 2

Let the random cost mapping be $\mathfrak{f}_s(\cdot) = c_s(\cdot) + \mathbf{i}_{X_s}(\cdot)$ (convex)

The projection subproblem in Step 2

$$(\hat{x}_s^{k+1}, \hat{u}_s^{k+1}) := \texttt{Proj}_{\texttt{epi}\,\mathfrak{f}_s}\left[\left(2x_s^k - z_{x_s}^k, 2u_s^k - z_{u_s}^k\right)\right]$$

reads as

$$(\texttt{SDAP}) \qquad \begin{cases} \displaystyle\min_{x_s, u_s} & \left\|x_s - (2x_s^k - z_{x_s}^k)\right\|^2 + \left[u_s - (2u_s^k - z_{u_s}^k)\right]^2 \\ \text{s.t.} & c_s(x_s) \le u_s, \quad x_s \in X_s \end{cases}$$

For comparison reasons, if the risk function falls into the expectation category (with $\zeta : \mathbb{R} \to \mathbb{R}$ convex and non-decreasing), then the PHA subproblem becomes

$$(\texttt{PHA}) \qquad \begin{cases} \displaystyle\min_{x_s, u_s} & \zeta(c_s(x_s) - u_s) + u_s + \frac{r}{2}\left\|x_s - \left(x_s^k - \frac{z_{x_s}^k}{r}\right)\right\|^2 + \frac{r}{2}\left[u_s - \left(u_s^k - \frac{z_{u_s}^k}{r}\right)\right]^2 \\ \text{s.t.} & x_s \in X_s \end{cases}$$

In contrast with PHA, the SDAP subproblem is independent of the risk function

Centre de Mathématiques Appliquées

Comments on SDAP - Step 2

- SDAP evaluates the proximal mapping $\hat{v}^{k+1} := \texttt{Prox}_{\frac{\rho}{r}}(2u^k - z_v^k)$ at every iteration
- This procedure, that is independent of the epi-projections, can be easily performed depending on ρ
- In the setting distributionally robust optimization, computing $\hat{v}^{k+1}$ amounts to projecting onto the ambiguity set and performing straightforward operations

Theorem

Let $\mathcal{P} \subset \mathbb{R}_+^S$ be a convex compact set, and suppose that

$$\rho(v) = \max_{p \in \mathcal{P}} \mathbb{E}_p[v]$$

Then

$$\texttt{Prox}_{\frac{\rho}{r}}(\mu) = \mu - \frac{1}{r}\texttt{Proj}_{\mathcal{P}}(r\mu)$$

This result justifies the algorithm's name: each step involve different kind of projections

Special setups

- Risk-neutral stochastic programs. Suppose that the ambiguity set is a singleton $\mathcal{P} = \{p\}$, with $p = (p_1, \ldots, p_S)$. Then $\texttt{Prox}_{\frac{\rho}{r}}(\mu) = \mu - \frac{1}{r}p$ is a straightforward operation

- Worst-case scenario. Let the ambiguity set be the simplex $\mathcal{P} = \Delta_S$: then $\texttt{Prox}_{\frac{\rho}{r}}(\mu)$ results in projecting $r\mu$ onto Δ_S, an easy task

- Conditional value-at-risk. Given a probability vector $p \in \mathbb{R}^S$, let the ambiguity set be

$$\mathcal{P}_{\mathrm{CVaR}_\alpha} = \left\{ q \in \mathbb{R}^S_+ : \sum_{s=1}^{S} q_s = 1,\ q \leq \frac{p}{(1-\alpha)} \right\},$$

which is the domain of the conjugate function of $\rho(\cdot) = \mathrm{CVaR}_\alpha(\cdot)$. Computing $\texttt{Prox}_{\frac{\mathrm{CVaR}_\alpha}{r}}(\mu) = \mu - \frac{1}{r}\texttt{Proj}_{\mathcal{P}_{\mathrm{CVaR}_\alpha}}(r\mu)$ boils down to solving a strictly convex QP problem of dimension S

Centre de Mathématiques Appliquées

Special setups

- **Wasserstein ambiguity set: the fixed support case.** The Wasserstein ambiguity set $\mathcal{P}_W^\epsilon$ is defined for $\{\xi^1, \ldots, \xi^S\}$ and $\epsilon > 0$ as

$$\mathcal{P}_W^\epsilon := \left\{ p \in \mathbb{R}_+^S : \begin{array}{lll} \sum_{s=1}^S \sum_{\ell=1}^L \eta_{s\ell} d(\xi^s, \hat{\xi}^\ell) & \leq \epsilon & \\ \sum_{s=1}^S \eta_{s\ell} & = \hat{p}_\ell, & \ell = 1, \ldots, L \\ \sum_{\ell=1}^L \eta_{s\ell} & = p_s, & s = 1, \ldots, S \\ \sum_{s=1}^S \sum_{\ell=1}^L \eta_{s\ell} & = 1 & \\ \eta & \geq 0 & \end{array} \right\}$$

  For the choice $\rho(v) = \max_{p \in \mathcal{P}_W^\epsilon} \mathbb{E}_p[v]$, computing $\texttt{Proj}_{\frac{\rho}{r}}$ requires solving a convex quadratic program with variables $p \in \mathbb{R}^S$ and $\eta \in \mathbb{R}^{S \times L}$, where L is the number of given reference scenarios $\hat{\xi}^\ell$

- **General setting.** If no additional assumption on $\rho(\cdot)$ (other than convexity and monotonicity) is assumed, then computing

$$\texttt{Prox}_{\frac{\rho}{r}}(\mu) = \arg \min_{y \in \mathbb{R}^{\mathbf{n}}} \ \rho(x) + \frac{r}{2} \|y - \mu\|^2$$

  can be done by off-the-shelf algorithms or bundle methods

Convergence analysis

$$\min_{x \in \mathbb{R}^{nS}} \rho\left(\mathfrak{f}_1(x_1), \ldots, \mathfrak{f}_S(x_S)\right) \quad \text{s.t.} \quad x \in \mathcal{N} \qquad \texttt{(Pbm)}$$

Theorem

Suppose $\rho : \mathbb{R}^S \to \mathbb{R}$ is convex and monotonically non-decreasing, and the convex problem `(Pbm)` *is solvable. Furthermore, assume that at least one of the following constraint qualifications hold:*

i) $\mathcal{N} \cap \mathbf{ri}\,\mathcal{D}\mathrm{om}(\sum_{s=1}^{S} \mathfrak{f}_s) \neq \emptyset$ *ii)* $\mathcal{N} \cap \mathcal{D}\mathrm{om}(\sum_{s=1}^{S} \mathfrak{f}_s) \neq \emptyset$ *and* $\mathfrak{f}_s$ *is polyhedral*

Then the SDAP generates a sequence $\{x^k\}$ that converges to a solution $\bar{x}$ of `(Pbm)`

Related algorithms

- SDAP is equivalent to applying DR to the operators ∂G and $\partial \mathbf{i}_{\mathfrak{L}}$, a primal approach
- A dual strategy consisting in applying DR to the operators $\partial[G^* \circ (-\mathbb{I})]$ and $\partial(\mathbf{i}_{\mathfrak{L}})^*$ yields an implementation of ADMM
- In the risk-neutral setting $f(x) = \sum_{s=1}^{S} p_s \mathfrak{f}_s(x_s)$, the DR applied to the operators $\mathtt{diag}(p)^{-1}\partial f$ and $\mathtt{diag}(p)^{-1}\partial \mathbf{i}_{\mathcal{N}}$ gives rise to PHA

Numerical experiments

We first compare the numerical performance of SDAP and PHA for risk-neutral, risk-averse, and worst-case scenario settings of a multistage production/inventory problem

The optimization problem consists of deciding the volume of production, inventory and external purchase of $\texttt{nprod} = 15$ of products to satisfy, at the minimal cost, a stochastic demand ξ over $T = 4$ stages (weeks)

All stages have fixed production, inventory and external supply costs, respectively given by $c_t^c, c_t^i, c_t^e \in \mathbb{R}^{\texttt{nprod}}$. The cost mapping is independent of scenarios

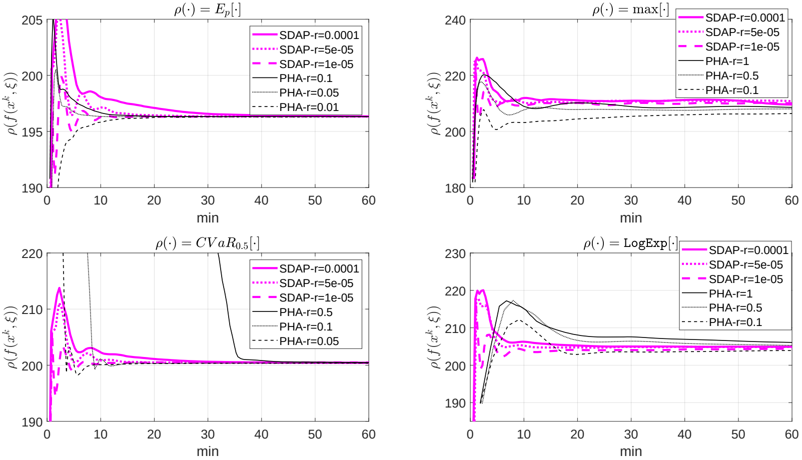
$$f(x, \xi) = \sum_{t=1}^{T} [{c_t^c}^\top x_t^c + {c_t^i}^\top x_t^i + {c_t^e}^\top x_t^e]$$

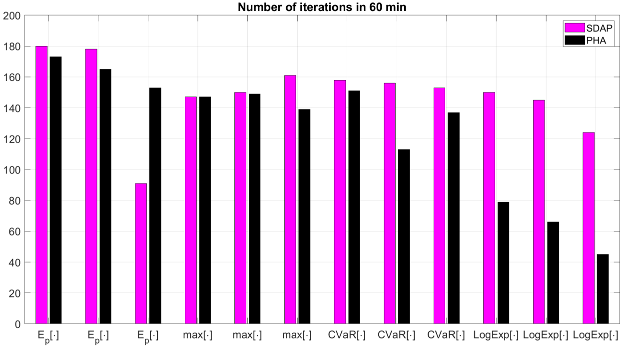
Deciding on how much of each product types to produce/purchase during a particular week forms the decision variables. The problem's constraints are:

- Production capacity: $\sum_{j=1}^{\texttt{nprod}} x_{jt}^c \leq \texttt{ProdCap}$, for $t = 1, \dots, T$
- Inventory capacity: $\sum_{j=1}^{\texttt{nprod}} x_{jt}^i \leq \texttt{InvCap}$, for $t = 1, \dots, T$
- Demand satisfaction: $x_{t-1}^i + x_t^e \geq \xi_t \in \mathbb{R}^{\texttt{nprod}}$, for $t = 1, \dots, T$
- Inventory balance: $x_t^i = x_t^c + [x_{t-1}^i + x_t^e - \xi_t]$, for $t = 1, \dots, T$
- $x_t^c, x_t^i, x_t^e \geq 0$ for $t = 1, \dots, T$

- Since we consider **`nprod`** $= 15$ products and $T = 4$ stages, the number of decision variables in the epi-projection subproblem is $181 = 4 \times 45 + 1$
- The considered scenario tree is composed of $S = 3\,000$ demand scenarios
- All the solvers ran for 60 minutes
- Numerical experiments were conducted on a PC Intel(R) with 32GB of RAM under Windows 10, using MATLAB 2020a in a parallel configuration with 4 workers, corresponding to the PC's 4 cores
- Subproblems were solved by one of the MATLAB's optimization routines **`linprog`**, **`quadprog`**, **`fmincon`** and **`fminunc`**, depending upon the subproblem's structure

SDAP versus PHA



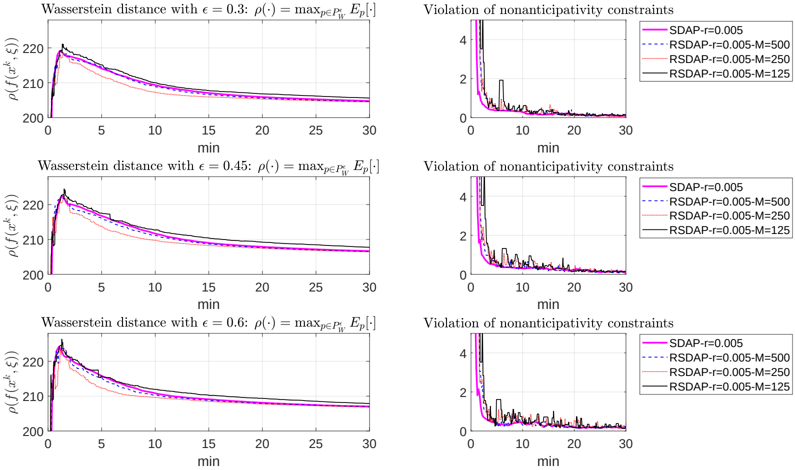


- Number of iterations performed by SDAP and PHA in 60 min of processing: four instances, and three different choices for the prox-parameter $r > 0$
- SDAP performed (within 60 min) twice more iterations than PHA in the challenging Log-Exponential instances
- For the other cases, SDAP and PHA performed more or less the same number of iterations (except for the case $r = \texttt{1e}{-}\texttt{5}$ of the risk-neutral instance, where SDAP performed fewer iteration due to numerical issues in `quadprog`)

- Thanks to the interpretation of SDAP as a variant of DR, it is not difficult to design a randomized variant of SDAP
- We denote such a randomized variant by RSDAP: M is the number of subproblems solved per iteration
- RSDAP is useful to alleviate the computational burden in Wasserstein DRO setting
- Distributionally robust optimization: results for a tree with $S = 1\,000$ scenarios
- In a total, $L = 250$ fixed reference scenarios were employed to construct the Wasserstein ambiguity set

The Wasserstein DRO setting

SDAP versus randomized SDAP



Concluding remarks

- We proposed a new algorithm denoted by SDAP for convex multistage optimization problems under uncertainty
- SDAP handles risk-neutral, distributionally robust, and risk-averse problems without changing the scenario subproblems' structure
- Such a property has a practical appeal because practitioners can solve risk-averse and distributionally robust versions of their problems in a single algorithm
- SDAP copes with the risk measure in an independent and dedicated step. This fact opens the way to deal with risk functions other than those handled by PHA
- Randomized, asynchronous and inexact variants of SDAP follow without much difficulties from the vast theory on the Douglas-Rachford algorithm
- Our randomized variant of SDAP avoids evaluating the risk-function's prox-mapping at every iteration

Thank you!

Reference

- W. de Oliveira. Risk-averse stochastic programming and distributionally robust optimization via operator splitting, Set-Valued and Variational Analysis, 2021. DOI: 10.1007/s11228-021-00600-5

Contact:

✉ welington.oliveira@mines-paristech.fr

www.oliveira.mat.br

We would like to acknowledge financial support from PGMO

Randomized SDAP (RSDAP)

Initialization. Let $z_x^0 \in \mathbb{R}^{nS}$, $z_u^0 \in \mathbb{R}^S$, $z_v^0 \in \mathbb{R}^S$, and $r > 0$ be given. Choose $nb \in \{1, \ldots, S\}$ and consider disjoint bundles $B_i \neq \emptyset$ such that $\bigcup_{i=1}^{nb} B_i = \{1, 2, \ldots, S+1\}$. Set $k := 0$

Step 1. Define

$$x := \mathtt{Proj}_{\mathcal{N}}(z_x^k) \qquad \text{and} \qquad u := \frac{z_u^k + z_v^k}{2}$$

Step 2. Draw an index $i \in \{1, \ldots, nb\}$ with probability $\pi_i > 0$. For all $s_\iota \in B_i$, compute (in parallel)

$$(\hat{x}_{s_\iota}, \hat{u}_{s_\iota}) := \mathtt{Proj}_{\mathtt{epi}\, \mathfrak{f}_{s_\iota}} \left[\left(2x_{s_\iota} - z^k_{x_{s_\iota}}, 2u_{s_\iota} - z^k_{u_{s_\iota}} \right) \right] \quad \text{if } s_\iota < S+1$$

$$\hat{v} := \mathtt{Prox}_{\frac{\rho}{r}}(2u - z_v^k) \quad \text{if } s_\iota = S+1$$

Step 3. For all $s_\iota \in B_i$ set

$$z^{k+1}_{x_{s_\iota}} := z^k_{x_{s_\iota}} + \hat{x}_{s_\iota} - x_{s_\iota} \quad \text{and} \quad z^{k+1}_{u_{s_\iota}} := z^k_{u_{s_\iota}} + \hat{u}_{s_\iota} - u_{s_\iota} \quad \text{if } s_\iota < S+1$$

$$z_v^{k+1} := z_v^k + \hat{v} - u \quad \text{if } s_\iota = S+1$$

For all the remaining subproblems $s \in \{1, \ldots, S+1\} \backslash B_i$, set

$$z^{k+1}_{x_s} = z^k_{x_s} \quad \text{and} \quad z^{k+1}_{u_s} = z^k_{u_s} \quad \text{if } s < S+1$$

$$z_v^{k+1} = z_v^k \quad \text{if } s = S+1$$

Set $k := k+1$ and go back to Step 1