Stochastic Optimization
Recalls on convex analysis

V. Leclère

November 24 2021

École des Ponts ParisTech

UNIVERSITÉ PARIS-EST

Objective of the course

- Uncertainty is present in most optimization problem, sometimes taken into account.
- Two major way of taking uncertainty into account :
  - Robust approach: assuming that uncertainty belongs in some set C, and will be chosen adversarily.
  - Stochastic approach: assuming that uncertainty is a random variable with known law.
- We will take the stochastic approach, considering the multi-stage approach : a first decision is taken, then part of the uncertainty is revealed, before taking a second decision and so on.

Syllabus

1st course: Convex toolbox
2nd course: Probability toolbox
3rd course: two-stage stochastic programm
4th course: Bellman operators and Dynamic Programming
5th course: Decomposition methods for two stage SP
6th course: Stochastic Dual Dynamic Programming

Validation

- The stochastic optimization course is in two part
- Evaluation have 2 components :
  - Practical works to be done in between classes and sent to vincent.leclere@enpc.fr
  - Written exam ith theoretical and modelling questions
- Practical work will be done in Julia (www.julialang.com)using jupyter notebook
- Instructions for installing julia / jupyter and using the library can be found at https://github.com/leclere/TP-Saclay
- Practical work will be posted there

Presentation Outline

Convex sets

- C is a convex set iff
$$\forall x_1, x_2 \in C, \quad [x_1, x_2] \subset C.$$
- If for all $i \in I$, C_i is convex, then so is $\cap_{i \in I} C_i$
- $C_1 + C_2$, and $C_1 \times C_2$ are convex
- For any set X the convex hull of X is the smallest convex set containing X,
$$\mathrm{conv}(X) := \Big\{ tx_1 + (1-t)x_2 \quad | \quad x_1, x_2 \in C, \quad t \in [0,1] \Big\}.$$
- The closed convex hull of X is the intersection of all half-spaces containing X.

Separation

Let X be a Banach space, and X^* its topological dual (i.e. the set of all continuous linear form on X).

Theorem (Simple separation)

Let A and B be convex non-empty, disjunct subsets of X. Assume that, $int(A) \neq \emptyset$, then there exists a separating hyperplane $(x^*, \alpha) \in X^* \times \mathbb{R}$ such that
$$\langle x^*, a \rangle \leq \alpha \leq \langle x^*, b \rangle \qquad \forall a, b \in A \times B.$$

Theorem (Strong separation)

Let A and B be convex non-empty, disjunct subsets of X. Assume that, A is closed, and B is compact (e.g. a point), then there exists a strict separating hyperplane $(x^*, \alpha) \in X^* \times \mathbb{R}$ such that, there exists $\varepsilon > 0$,
$$\langle x^*, a \rangle + \varepsilon \leq \alpha \leq \langle x^*, b \rangle - \varepsilon \qquad \forall a, b \in A \times B.$$

Convex functions : basic properties

- A function $f : X \to \bar{\mathbb{R}}$ is convex if its epigraph is convex.
- $f : X \to \mathbb{R} \cup \{+\infty\}$ is convex iff
$$\forall t \in [0,1], \quad \forall x, y \in X, \qquad f(tx + (1-t)y) \leq tf(x) + (1-t)f(y).$$
- If f, g convex, $\lambda > 0$, then $\lambda f + g$ is convex.
- If f convex non-decreasing, g convex, then $f \circ g$ convex.
- If f convex and a affine, then $f \circ a$ is convex.
- If $(f_i)_{i \in I}$ is a family of convex functions, then $\sup_{i \in I} f_i$ is convex.

Convex functions : further definitions and properties

- The domain of a convex function is $\mathrm{dom}(f) = \{x \in X \mid f(x) < +\infty\}$.
- The level set of a convex function is $\mathrm{lev}_\alpha(f) = \{x \in X \mid f(x) \leq \alpha\}$
- A function is lower semi continuous (lsc) iff for all $\alpha \in \mathbb{R}$, lev_α is closed.
- The domain and the level sets of a convex function are convex.
- A convex function is proper if it never takes $-\infty$, and $\mathrm{dom}(f) \neq \emptyset$.
- A function is coercive if $\lim_{\|x\| \to \infty} f(x) = +\infty$.

Convex functions : polyhedral functions

- A polyhedra is a finite intersection of half-spaces, thus convex.
- A polyhedral function is a function whose epigraph is a polyhedra.
- Finite intersection, cartesian product and sum of polyhedra is polyhedra.
- In particular a polyhedral function is convex lsc, with polyhedral domain and level sets.
- If $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is polyhedral, then it can be written as

$$\begin{aligned} f(x) = \min_{\theta} \quad & \theta \\ s.t. \quad & \alpha_\kappa^\top x + \beta_\kappa \leq \theta && \forall \kappa \leq k \\ & \gamma_\kappa{}^\top x + \delta_\kappa \leq 0 && \forall \kappa \leq k' \end{aligned}$$

Convex functions : polyhedral approximations

- f is convex iff it is above all its tangeant.
- Let $\{x_\kappa, g_\kappa\}_{\kappa \leq k}$ be a collection of (sub-)gradient, that is such that $f \geq \langle g_\kappa, \cdot - x_\kappa \rangle + f(x_\kappa)$, then

$$\underline{f}_k : x \mapsto \max_{\kappa \leq k} \langle g_\kappa, x - x_\kappa \rangle + f(x_\kappa)$$

  is a polyhedral outer-approximation of f.
- Let $\{x_\kappa\}_{\kappa \leq k}$ be a collection of point in $dom(f)$. Then,

$$\bar{f}_k : x \mapsto \min_{\sigma \in \Delta_k} \Big\{ \sum_{\kappa=1}^{k} \sigma_\kappa f(x_\kappa) \;\Big|\; \sum_{\kappa=1}^{k} \sigma_\kappa x_\kappa = x \Big\}$$

  is a polyhedral inner-approximation of f.

Convex functions : strict and strong convexity

- $f : X \to \mathbb{R} \cup \{+\infty\}$ is strictly convex iff

$$\forall t \in]0,1[, \quad \forall x, y \in X, \qquad f(tx + (1-t)y) < tf(x) + (1-t)f(y).$$

- $f : X \to \mathbb{R} \cup \{+\infty\}$ is α-convex iff $\forall x, y \in X$

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\alpha}{2}\|y - x\|^2.$$

- If $f \in C^1(\mathbb{R}^n)$
  - $\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq 0$ iff f convex
  - if strict inequality holds, then f strictly convex
- If $f \in C^2(\mathbb{R}^n)$,
  - $\nabla^2 f \succcurlyeq 0$ iff f convex
  - if $\nabla^2 f \succ 0$ then f strictly convex
  - if $\nabla^2 f \succcurlyeq \alpha I$ then f is α-convex

Presentation Outline

Convex optimization problem

$$\min_{x \in C} \quad f(x)$$

Where C is closed convex and f convex finite valued, is a convex optimization problem.

- If C is compact and f proper lsc, then there exists an optimal solution.
- If f proper lsc and coercive, then there exists an optimal solution.
- The set of optimal solutions is convex.
- If f is strictly convex the minimum (if it exists) is unique.
- If f is α-convex the minimum exists and is unique.

Constraints and infinite values

A very standard trick in optimization consists in replacing constraints by infinite value of the cost function.

$$\min_{x \in C \subset X} f(x) = \min_{x \in X} f(x) + \mathbb{I}_C(x).$$

where

$$\mathbb{I}_C(x) = \begin{cases} 0 & \text{if } x \in C \\ +\infty & \text{otherwise} \end{cases}$$

- If f is lsc and C is closed, then $f + \mathbb{I}_C$ is lsc.
- If f is proper and C is bounded, then $f + \mathbb{I}_C$ is coercive.
- Thus, from a theoretical point of view, we do not need to explicitely write constraint in a problem.

Presentation Outline

Subdifferential of convex function

Let X be a Banach space, $f : X \to \bar{\mathbb{R}}$.

- X^* is the topological dual of X, that is the set of continuous linear form on X.
- The subdifferential of f at $x \in \mathrm{dom}(f)$ is the set of slopes of all affine minorants of f exact at x:

$$\partial f(x) := \Big\{ x^* \in X^* \quad | \quad f(\cdot) \geq \langle x^*, \cdot - x \rangle + f(x) \Big\}.$$

- If f is convex and derivable at x then

$$\partial f(x) = \{\nabla f(x)\}.$$

Partial infimum

Let $f : X \times Y \to \bar{\mathbb{R}}$ be a jointly convex and proper function, and define

$$v(x) = \inf_{y \in Y} f(x, y)$$

then v is convex.
If v is proper, and $v(x) = f(x, y^\sharp(x))$ then

$$\partial v(x) = \left\{ g \in X^* \quad | \quad \begin{pmatrix} g \\ 0 \end{pmatrix} \in \partial f(x, y^\sharp(x)) \right\}$$

proof:

$$\begin{aligned}
g \in \partial v(x) \quad &\Leftrightarrow \quad \forall x', \qquad v(x') \geq v(x) + \langle g, x' - x \rangle \\
&\Leftrightarrow \quad \forall x', y' \quad f(x', y') \geq f(x, y^\sharp(x)) + \left\langle \begin{pmatrix} g \\ 0 \end{pmatrix}, \begin{pmatrix} x' \\ y' \end{pmatrix} - \begin{pmatrix} x \\ y^\sharp(x) \end{pmatrix} \right\rangle \\
&\Leftrightarrow \quad \begin{pmatrix} g \\ 0 \end{pmatrix} \in \partial f(x, y^\sharp(x))
\end{aligned}$$

Convex function : regularity

- Assume f convex, then f is continuous on the relative interior of its domain, and Lipschtiz on any compact contained in the relative interior of its domain.
- A proper convex function is subdifferentiable on the relative interior of its domain
- Assume $f : X \to \bar{\mathbb{R}}$ is convex, and consider $A \subset X$.
  - If f is L-Lipschitz on A then $\partial f(x) \subset B(0, L), \quad \forall x \in ri(A)$
  - If $\partial f(x) \subset B(0, L), \quad \forall x \in A + \varepsilon B(0, 1)$ then f is L-Lipschitz on A then

Fenchel transform

Let X be a Banach space, $f : X \to \bar{\mathbb{R}}$ convex proper.

- The Fenchel transform of f, is $f^* : X^* \to \bar{\mathbb{R}}$ with

$$f^*(x^*) := \sup_{x \in X} \langle x^*, x \rangle - f(x).$$

- f^* is convex lsc as the supremum of affine functions.
- $f \leq g$ implies that $f^* \geq g^*$.
- If f is proper convex lsc, then $f^{**} = f$, otherwise $f^{**} \leq f$.

Fenchel transform and subdifferential

- By definition $f^*(x^*) \geq \langle x^*, x\rangle - f(x)$ for all x,
- thus we always have (Fenchel-Young) $f(x) + f^*(x^*) \geq \langle x^*, x\rangle$.
- Recall that $x^* \in \partial f(x)$ iff for all x', $f(x') \geq f(x) + \langle x^*, x' - x\rangle$ iff

$$\langle x^*, x\rangle - f(x) \geq \langle x^*, x'\rangle - f(x') \qquad \forall x'$$

that is

$$x^* \in \partial f(x) \Leftrightarrow x \in \arg\max_{x' \in X} \{\langle x^*, x'\rangle - f(x')\} \Leftrightarrow f(x) + f^*(x^*) = \langle x^*, x\rangle$$

- From Fenchel-Young equality we have

$$\partial v^{**}(x) \neq \emptyset \quad \Longrightarrow \quad \partial v^{**}(x) = \partial v(x) \text{ and } v^{**}(x) = v(x).$$

- If f proper convex lsc

$$x^* \in \partial f(x) \quad \Longleftrightarrow \quad x \in \partial f^*(x^*).$$

Presentation Outline

Weak duality

The problem

$$\begin{aligned} (P) \quad & \min_{x \in \mathbb{R}^n} && f(x) \\ & s.t. && c_i(x) = 0 && \forall i \in [\![1, n_E]\!] \\ & && c_j(x) \leq 0 && \forall j \in [\![n_E + 1, n_E + n_I]\!] \end{aligned}$$

can be written

$$\min_{x \in \mathbb{R}^n} \max_{\lambda \in \mathbb{R}^{n_E}, \mu \in \mathbb{R}^{n_I}_+} \mathcal{L}(x, \lambda, \mu)$$

where

$$\mathcal{L}(x, \lambda, \mu) := f(x) + \sum_{i=1}^{n_E + n_I} \lambda_i c_i(x)$$

The dual problem is

$$(D) \qquad \max_{\lambda \in \mathbb{R}^{n_E} \times \mathbb{R}^{n_I}_+} \min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \mu)$$

and we have, without assumption

$$v_D \leq v_P.$$

Linear Programming duality

$$\begin{aligned} & \min_{x \geq 0} && c^\top x \\ & s.t. && Ax = b \end{aligned}$$

is equivalent to

$$\min_{x \geq 0} \max_{\lambda} (c - A^\top \lambda)^\top x + b^\top \lambda$$

and the dual problem is

$$\begin{aligned} & \max_{\lambda} && b^\top \lambda \\ & s.t. && A^\top \lambda \leq c \end{aligned}$$

with equality between both problem except if there is neither primal nor dual admissible solution.

Strong duality

The duality gap is the difference between the primal value and dual value of a problem.
Consider problem

$$
\begin{aligned}
(P) \quad & \min_{x\in\mathbb{R}^n} \quad f(x) \\
& s.t. \quad c_i(x) = 0 && \forall i \in [\![1, n_E]\!] \\
& \qquad\quad c_j(x) \leq 0 && \forall j \in [\![n_E+1, n_E+n_I]\!]
\end{aligned}
$$

with (P) convex in the sense that f is convex, c_I is convex lsc, c_I is affine.
If further the constraints are qualified, then there is no duality gap.

Recall KKT

Assume that f, g_i and h_j are differentiable. Assume that $x^\sharp$ is an optimal solution of (P), and that the constraints are qualified in $x^\sharp$. Then we have

$$
\begin{cases}
\nabla_x \mathcal{L}(x^\sharp, \lambda^\sharp) = \nabla f(x^\sharp) + \sum_{i=1}^{n_E+n_I} \lambda_i^\sharp \nabla c_i(x^\sharp) = 0 \\
c_E(x^\sharp) = 0 \\
0 \leq \lambda_I \perp c_I(x^\sharp) \leq 0
\end{cases}
$$

Presentation Outline

Perturbed problem

Consider the perturbed problem

$$
\begin{aligned}
(P_p) \quad & \min_{x\in\mathbb{R}^n} \quad f(x) \\
& s.t. \quad c_i(x) + p_i = 0 && \forall i \in [\![1, n_E]\!] \\
& \qquad\quad c_j(x) + p_j \leq 0 && \forall j \in [\![n_E+1, n_I+n_E]\!]
\end{aligned}
$$

with value $v(p)$, and optimal multiplier (for $p=0$) λ_0.

Linear programming case

$$v(p) := \min_{x \geq 0} \quad c^\top x \qquad s.t. \quad Ax + p = b$$

by LP duality (assuming at least one admissible primal solution) we have

$$v(p) = \max_{\lambda} \quad -b^\top \lambda + p^\top \lambda \qquad s.t. \quad A^\top \lambda \leq c$$

Note λ_0 the optimal multiplier for (P_0), note that it is admissible for (D_p), hence $v(p) \geq -b^\top \lambda_0 + p^\top \lambda_0$. By strong duality we have $v(0) = -b^\top \lambda_0$, hence

$$v(p) \geq v(0) + \lambda_0^\top p$$

or

$$\lambda_0 \in \partial v(0).$$

Optimality condition by saddle point

Let $\Lambda := \mathbb{R}^{n_E} \times \mathbb{R}^{n_I}_+$. $(x^\sharp, \lambda^\sharp)$ is a saddle-point of $\mathcal{L}$ on $\mathbb{R}^n \times \Lambda$ iff

$$\forall \lambda \in \Lambda, \quad \mathcal{L}(x^\sharp, \lambda) \leq \mathcal{L}(x^\sharp, \lambda^\sharp) \leq \mathcal{L}(x, \lambda^\sharp), \quad \forall x \in \mathbb{R}^n$$

Consider $(\bar{x}, \bar{\lambda}) \in \mathbb{R}^n \times \Lambda$. Then $\bar{\lambda} \in \arg\max_{\lambda \in \Lambda} \mathcal{L}(\bar{x}, \lambda)$ iff $c_E(\bar{x}) = 0$ and $0 \leq \bar{\lambda}_I \perp c_I(\bar{x}) \leq 0$.

Theorem

If $(x^\sharp, \lambda^\sharp)$ is a saddle-point of $\mathcal{L}$ on $\mathbb{R}^n \times \Lambda$, then $x^\sharp$ is an optimal solution of (P).

Note that we need no assumption for this result.

Convex case

If (P) is convex in the sense that f is convex, c_I is convex and c_E is affine, then v is convex.

Theorem

Assume that v is convex, then

$$\partial v(0) = \{\lambda \in \Lambda \quad | \quad (x, \lambda) \text{ is a saddle point of } \mathcal{L}\}$$

In particular, $\partial v(0) \neq \emptyset$ iff there exists a saddle point of $\mathcal{L}$.

Theorem (Slater's qualification condition)

Consider a convex optimisation problem. Assume that c_E' is onto, and there exists $x \in rint(\mathrm{dom}(f))$ with $c_I(x) < 0$, and c_I continuous at x, then if x^* is an optimal solution, there exists λ^* such that (x^*, λ^*) is a saddle-point of the Lagrangian. Further, v is locally Lipschitz around 0.

Presentation Outline

Duality by abstract perturbation

Let $\mathbb{X}$ and $\mathbb{Y}$ be Banach spaces. There exists an abstract duality framework for $\min_{x\in\mathbb{X}} f(x)$ by considering a perturbation function $\Phi : \mathbb{X}\times\mathbb{Y} \to \mathbb{R}\cup\{+\infty\}$ (with $\Phi(\cdot,0)=f$).

$$(\mathcal{P}_y) \qquad v(y) := \inf_{x\in\mathbb{X}} \Phi(x,y).$$

We have

$$\begin{aligned} v^*(y^*) &= \sup_{y\in\mathbb{Y}} \langle y^*, y\rangle - v(y) \\ &= \sup_{x,y} \langle y^*, y\rangle - \Phi(x,y) = \Phi^*(0,y^*) \end{aligned}$$

Thus we have

$$(\mathcal{D}_y) \qquad v^{**}(y) = \sup_{y^*\in\mathbb{Y}^*} \langle y^*, y\rangle - \Phi^*(0,y^*)$$

Generically

$$\mathrm{val}(\mathcal{D}_y) = v^{**}(y) \le v(y) = \mathrm{val}(\mathcal{P}_y)$$

Solution of the dual as subgradient

Note that the set of solution of the dual is $S(\mathcal{D}_y) = \partial v^{**}(y)$.
Recall that, for v proper convex,

$$\partial v^{**}(x) \neq \emptyset \implies \partial v^{**}(x) = \partial v(x) \text{ and } v^{**}(x) = v(x)$$

Thus, if v is proper convex and subdifferentiable at y (or equivalently if $S(\mathcal{D}_y) \neq \emptyset$), then,

$$\begin{aligned} \mathrm{val}(\mathcal{D}_y) &= \mathrm{val}(\mathcal{P}_y) \\ S(\mathcal{D}_y) &= \partial v(y) \end{aligned}$$

Finally, as a convex function is subdifferentiable on the relative interior of its domain, a sufficient qualification condition (to have a zero dual gap and existence of multipliers), is that

$$0 \in rint(\mathrm{dom}(v)).$$

Recovering the Lagrangian dual

Problem $(\mathcal{P}_y)$ can be written

$$\begin{aligned} \min_{x,z} \quad & \Phi(x,z) \\ s.t. \quad & z = y \end{aligned}$$

with Lagrangian dual

$$\max_{y^*\in Y^*} \inf_{x,z\in X\times Y} \Phi(x,z) + \langle y^*, y-z\rangle = \max_{y^*\in Y^*} \langle y^*, y\rangle - \underbrace{\sup_{x,z\in X\times Y} \Big\{\langle y^*, z\rangle - \Phi(x,z)\Big\}}_{\Phi^*(0,y^*)}$$

Hence, we recover the Fenchel dual from the Lagrangian dual.

For next week

- Install Julia / Jupyter / JuMP (see instructions `https://github.com/leclere/TP-Saclay`)
- Run the CrashCourse notebook to get used with those tools (there are other resources available on the web as well)
- Contact me vincent.leclere@enpc.fr in case of trouble

Stochastic Optimization
Recalls on probability

V. Leclère

December 1st 2021

École des Ponts ParisTech

UNIVERSITÉ PARIS-EST

Presentation Outline

Probability space

- Let Ω be a set.
- A σ-algebra $\mathcal{F}$ of Ω is a collection of subset of Ω such that
  - $\Omega \in \mathcal{F}$
  - $\mathcal{F}$ is closed under complementation
  - $\mathcal{F}$ is closed under countable union
- A measure $\mathbb{P} : \mathcal{F} \to [0, 1]$ is a probability if
  - $\mathbb{P}(\Omega) = 1$
  - $\mathbb{P}(\cup_{i\in\mathbb{N}} A_i) = \sum_{i\in\mathbb{N}} \mathbb{P}(A_i)$ where $\{A_i\}_{i\in\mathbb{N}}$ is a collection of pairwise disjoint sets of $\mathcal{F}$
- $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space.
- $A \in \mathcal{F}$ is $\mathbb{P}$-almost-sure if $\mathbb{P}(A) = 1$, and negligible if $\mathbb{P}(A) = 0$.
- $(\Omega, \mathcal{F}, \mathbb{P})$ is complete if all subset of a negligible set is measurable.

Measurability and representation

- Let $\mathcal{F}$ be a σ-algebra on Ω.
- A σ-algebra is generated by a collection of sets if it is the smallest containing the collection.
- A function $X : \Omega \to \mathbb{R}^n$ is $\mathcal{F}$-measurable if $X^{-1}(I) \in \mathcal{F}$ for all boxes I of $\mathbb{R}^n$, we note $X \preceq \mathcal{F}$.
- A σ-algebra $\sigma(X)$ is generated by a function $X : \Omega \to \mathbb{R}^n$ sets if it is generated by $\{X^{-1}(I) \mid I \text{ boxes of } \mathbb{R}^n\}$.
- The σ-algebra generated by all boxes is called the Borel σ-algebra.

Theorem (Doob-Dynkin)

Let $X : \Omega \to \mathbb{R}^n$, $Y : \Omega \to \mathbb{R}^p$ be two $\mathcal{F}$-measurable functions. Then $Y \preceq \sigma(X)$ iff there exists a Borel measurable function $f : \mathbb{R}^n \to \mathbb{R}^p$ such that $Y = f(X)$.

Random variables

- Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space.
- Define the equivalence class over the $\mathcal{L}^0(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n)$

$$X \sim Y \quad \iff \quad \mathbb{P}\Big(\{\omega \in \Omega \mid X(\omega) = Y(\omega)\}\Big) = 1$$

- A random variable $\boldsymbol{X}$ is an element of $L^0(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n) := \mathcal{L}^0(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n)/\sim$.
- In other word a random variable is a measurable function from Ω to $\mathbb{R}^n$ defined up to negligeable set.

Expectation and variance

- We recall that $\mathbb{E}\big[\boldsymbol{X}\big] := \int_\Omega \boldsymbol{X}(\omega)\mathbb{P}(d\omega)$.
- If $\mathbb{P}$ is discrete, we have $\mathbb{E}\big[\boldsymbol{X}\big] = \sum_{\omega=1}^{|\Omega|} X(\omega)p_\omega$.
- If $\boldsymbol{X}$ admit a density function f we have $\mathbb{E}\big[\boldsymbol{X}\big] = \int_\mathbb{R} xf(x)dx$.
- We define the variance of $\boldsymbol{X}$

$$var(\boldsymbol{X}) := \mathbb{E}\Big[\Big(\boldsymbol{X} - \mathbb{E}\big[\boldsymbol{X}\big]\Big)^2\Big] = \mathbb{E}\big[\boldsymbol{X}^2\big] - \big(\mathbb{E}\big[\boldsymbol{X}\big]\big)^2$$

- and the standard deviation

$$std(\boldsymbol{X}) := \sqrt{var(\boldsymbol{X})}$$

- the covariance is given by

$$cov(\boldsymbol{X}, \boldsymbol{Y}) = \mathbb{E}\big[\boldsymbol{X}\boldsymbol{Y}\big] - \mathbb{E}\big[\boldsymbol{X}\big]\mathbb{E}\big[\boldsymbol{Y}\big]$$

Random variables spaces

- $L^0(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n)$ is the set of rv
- $L^1(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n)$ is the set of rv such that $\mathbb{E}\big[|\boldsymbol{X}|\big] < +\infty$
- $L^p(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n)$ is the set of rv such that $\mathbb{E}\big[|\boldsymbol{X}|^p\big] < +\infty$
- $L^\infty(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n)$ is the set of rv that is almost surely bounded
- $L^p(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n)$, for $p \in]1, +\infty[$ is a reflexive Banach space, with dual L^q, where $\dfrac{1}{p} + \dfrac{1}{q} = 1$
- $L^1(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n)$ is a non-reflexive Banach space with dual L^∞
- $L^2(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n)$ is a Hilbert space
- $L^\infty(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n)$ is a non-reflexive Banach space

Independence

- The cumulative distribution function (cdf) of a random variable $\boldsymbol{X}$ is

$$F_X(x) := \mathbb{P}(\boldsymbol{X} \leq x)$$

- Two random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent iff (one of the following)
  - $F_{X,Y}(a, b) = F_X(a)F_Y(b)$ for all a, b
  - $\mathbb{P}(\boldsymbol{X} \in A, \boldsymbol{Y} \in B) = \mathbb{P}(\boldsymbol{X} \in A)\mathbb{P}(\boldsymbol{Y} \in B)$ for all Borel sets A and B
  - $\mathbb{E}\big[f(\boldsymbol{X})g(\boldsymbol{Y})\big] = \mathbb{E}\big[f(\boldsymbol{X})\big]\mathbb{E}\big[g(\boldsymbol{Y})\big]$ for all Borel functions f and g
- A sequence of identically distributed indenpendent variables is denoted iid.

Inequalities

- (Markov) $\mathbb{P}(|\boldsymbol{X}| \geq a) \leq \frac{\mathbb{E}\left[|\boldsymbol{X}|\right]}{a}$, for $a > 0$.
- (Chernoff) $\mathbb{P}(\boldsymbol{X} \geq a) \leq \frac{\mathbb{E}\left[e^{t\boldsymbol{X}}\right]}{e^{ta}}$, for $t, a > 0$.
- (Chebyshev) $\mathbb{P}(|\boldsymbol{X} - \mathbb{E}[X]| \geq a) \leq \frac{var(\boldsymbol{X})}{a^2}$, for $a > 0$.
- (Jensen) $\mathbb{E}\left[f(\boldsymbol{X})\right] \geq f(\mathbb{E}\left[\boldsymbol{X}\right])$ for f convex
- (Cauchy-Schwartz) $\mathbb{E}\left[|\boldsymbol{XY}|\right] \leq \|\boldsymbol{X}\|_2 \|\boldsymbol{Y}\|_2$
- (Hölder) $\mathbb{E}\left[|\boldsymbol{XY}|\right] \leq \|\boldsymbol{X}\|_p \|\boldsymbol{Y}\|_q$ for $\frac{1}{p} + \frac{1}{q} = 1$
- (Hoeffding) $\mathbb{P}\Big(\boldsymbol{M}_n - \mathbb{E}\left[\boldsymbol{M}_n\right] \geq t\Big) \leq \exp\left(\frac{-2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right)$ where $\{\boldsymbol{X}_i\}_{i\in\mathbb{N}}$ is a sequence of bounded independent rv with $a_i \leq \boldsymbol{X}_i \leq b_i$.

Limits of random variable

Let $\{\boldsymbol{X}_n\}_{n\in\mathbb{N}}$ be a sequence of random variables.

- We say that $\{\boldsymbol{X}_n\}_{n\in\mathbb{N}}$ converges almost surely toward $\boldsymbol{X}$ if
$$\mathbb{P}\Big(\lim_n (\boldsymbol{X}_n - \boldsymbol{X}) = 0\Big) = 1.$$
- We say that $\{\boldsymbol{X}_n\}_{n\in\mathbb{N}}$ converges in probability toward $\boldsymbol{X}$ if
$$\forall \varepsilon > 0, \qquad \mathbb{P}(|\boldsymbol{X}_n - \boldsymbol{X}| > \varepsilon) \to 0.$$
- We say that $\{\boldsymbol{X}_n\}_{n\in\mathbb{N}}$ converges in L^p toward $\boldsymbol{X}$ if
$$\|\boldsymbol{X}_n - \boldsymbol{X}\|_p = \mathbb{E}\Big[|\boldsymbol{X}_n - \boldsymbol{X}|^p\Big] \to 0.$$
- We say that $\{\boldsymbol{X}_n\}_{n\in\mathbb{N}}$ converges in law toward $\boldsymbol{X}$ if
$$\mathbb{E}\left[f(\boldsymbol{X}_n)\right] \to \mathbb{E}\left[f(\boldsymbol{X})\right] \quad \text{for all bounded Lipschitz } f$$

Conditional expectation

- $\mathbb{P}(A|B) = \mathbb{P}(A \cap B)/\mathbb{P}(B)$
- If $(\boldsymbol{X}, \boldsymbol{Y})$ has density $f_{X,Y}$, then the conditional law $(\boldsymbol{X}|\boldsymbol{Y})$ has density $f_{X|Y}(x|y) = f_{X,Y}(x,y)/f_Y(y)$.
- In the continuous case we have
$$\mathbb{E}\left[\boldsymbol{X}|\boldsymbol{Y} = y\right] = \int_{\mathbb{R}} x f_{X|Y}(x|y)dx.$$
- More generally if $\mathcal{G}$ is a sub-sigma-algebra of $\mathcal{F}$, the conditional expectation of $\boldsymbol{X} \in L^1(\Omega, \mathcal{F}, \mathbb{P})$ w.r.t $\mathcal{G}$ is the $\mathcal{G}$-measurable random variable $\boldsymbol{Y}$ satisfying
$$\mathbb{E}\left[\boldsymbol{Y}\mathbb{1}_G\right] = \mathbb{E}\left[\boldsymbol{X}\mathbb{1}_G\right], \qquad \forall G \in \mathcal{G}$$
- Finally, we always have
$$\mathbb{E}\Big[\mathbb{E}\left[\boldsymbol{X}|\boldsymbol{Y}\right]\Big] = \mathbb{E}\left[\boldsymbol{X}\right]$$

Presentation Outline

Monotone and dominated convergence

Theorem (Monotone convergence)

Let $\{X_n\}_{n\in\mathbb{N}}$ be a sequence of random variables such that

- $X_{n+1} \geq X_n$ $\mathbb{P}$-a.s.
- $X_n \to X_\infty$ $\mathbb{P}$-a.s.

then $\lim_{n\to\infty} \mathbb{E}\big[X_n\big] = \mathbb{E}\big[\lim_n X_n\big]$

Theorem (Dominated convergence)

Let $\{X_n\}_{n\in\mathbb{N}}$ be a sequence of random variables, and Y such that

- $|X_n| \leq Y$ $\mathbb{P}$-a.s. *with* $\mathbb{E}\big[|Y|\big] < +\infty$
- $X_n \to X_\infty$ $\mathbb{P}$-a.s.

then $\lim_{n\to\infty} \mathbb{E}\big[X_n\big] = \mathbb{E}\big[\lim_n X_n\big]$

Measurability of multi-valued function

Consider a measurable space $(\Omega, \mathcal{F})$.

- A function $f : \Omega \to \mathbb{R}$ is $\mathcal{F}$-measurable if $f^{-1}(I) \in \mathcal{F}$ for all interval I of $\mathbb{R}$.
- A multi-function $\mathcal{G} : \Omega \rightrightarrows \mathbb{R}^n$ is $\mathcal{F}$-measurable if

$$\forall A \subset \mathbb{R}^n \text{closed}, \quad \mathcal{G}^{-1}(A) := \big\{\omega \in \Omega \mid \mathcal{G}(\omega) \cap A \neq \emptyset\big\} \in \mathcal{F}.$$

- A closed valued multi-function $\mathcal{G} : \Omega \rightrightarrows \mathbb{R}^n$ is $\mathcal{F}$-measurable iff $d_x(\omega) := \mathrm{dist}(x, \mathcal{G}(\omega))$ is $\mathcal{F}$-measurable.

Theorem (Measurable selection theorem)

If $\mathcal{G} : \Omega \rightrightarrows \mathbb{R}^n$ is a closed valued measurable multifunction, then there exists a measurable selection of $\mathcal{G}$, that is a measurable function $\pi : \mathrm{dom}(\mathcal{G}) \subset \Omega \to \mathbb{R}^n$ such that $\pi(\omega) \in \mathcal{G}(\omega)$ for all $\omega \in \mathrm{dom}(\mathcal{G})$.

Normal integrand

Assume that $\mathcal{F}$ is $\mathbb{P}$-complete.

Definition (Caratheodory function)

$f : \mathbb{R}^n \times \Omega \to \mathbb{R}$ is a Carathéodory function if

- $f(\cdot, \omega)$ is continuous for a.a. $\omega \in \Omega$
- $f(x, \cdot)$ is measurable for all $x \in \mathbb{R}^n$

Definition (Normal integrand)

$f : \mathbb{R}^n \times \Omega \to \bar{\mathbb{R}}$ is a normal integrand (aka random lowersemicontinuous function) if

- $f(\cdot, \omega)$ is lsc for a.a. $\omega \in \Omega$
- $f(\cdot, \cdot)$ is measurable

f is a convex normal integrand if in addition it is convex in x for a.a. $\omega \in \Omega$.

Measurability of minimum and argmin

Theorem (Measurability of minimum)

Let $f : \mathbb{R}^n \times \Omega \to \bar{\mathbb{R}}$ be a normal integrand and define

$$\vartheta(\omega) := \inf_x f(x, \omega) \qquad X^*(\omega) := \arg\min_x f(x, \omega).$$

Then, ϑ and X^* are measurable.

Theorem (Pointwise minimization)

Let $f : \mathbb{R}^n \times \Omega \to \bar{\mathbb{R}}$ be a normal convex integrand then

$$\inf_{\boldsymbol{U} \in L^0, \boldsymbol{U} \in U} \mathbb{E}\big[f(\boldsymbol{U}(\omega), \omega)\big] = \mathbb{E}\Big[\inf_{u \in U(\omega)} f(u, \omega)\Big]$$

Continuity and derivation under expectation

Let $f : \mathbb{R}^n \times \Omega$ be a random function (i.e. measurable in ω for all x). We say that f is dominated on X if, for all $x \in X$, there exists an integrable random variable $\boldsymbol{Y}$ such that $f(x, \cdot) \leq \boldsymbol{Y}$ almost surely. If f is dominated on $X \subset \mathbb{R}^n$, we define $F(x) := \mathbb{E}\big[f(x, \omega)\big]$.

- If f is lsc in x and dominated on X, then F is lsc.
- If f is continuous in x and dominated on X, then F is continuous.
- If f is Lispchitz in x, with $\mathbb{E}\big[\mathrm{lip}(f(\cdot, \omega))\big] < +\infty$, then F in Lipschitz continous. Moreover if f is differentiable in x, we have

$$\nabla F(x) = \mathbb{E}\big[\nabla_x f(x, \omega)\big].$$

- If f is a convex normal integrand, and $x_0 \in \mathrm{int}(\mathrm{dom}(F))$, then

$$\partial F(x_0) = \mathbb{E}\big[\partial f(x_0, \omega)\big]$$

Presentation Outline

Strong Law of large number

- We consider a function $f : \mathbb{R}^n \times \Xi \to \mathbb{R}$, and a random variable $\boldsymbol{\xi}$ which takes values in Ξ, and define $F(x) := \mathbb{E}\big[f(x, \boldsymbol{\xi})\big]$.
- We consider a sequence of random variables $\{\boldsymbol{\xi}_i\}_{i \in \mathbb{N}}$.
- We define the average function

$$\hat{F}_N(x) := \frac{1}{N} \sum_{i=1}^{N} f(x, \boldsymbol{\xi}_i)$$

- We say that we have a Law of Large Number (LLN) if,

$$\forall x \in \mathbb{R}^n, \qquad \mathbb{P}\Big(\lim_n \hat{F}_n(x) = F(x)\Big) = 1$$

- The strong LLN state that LLN holds if $f(x, \boldsymbol{\xi})$ is integrable, and $\{\boldsymbol{\xi}_i\}_{i \in \mathbb{N}}$ is a iid (with same law as $\boldsymbol{\xi}$).

Uniform Law of large number

- Having LLN means that, for all $\varepsilon > 0$ (and almost all sample),

$$\forall x, \quad \exists N_\varepsilon \in \mathbb{N}, \qquad n \geq N \implies |\hat{F}_N(x) - F(x)| \leq \varepsilon$$

- We say that we have ULLN if for all $\varepsilon > 0$ (and almost all sample),

$$\exists N_\varepsilon \in \mathbb{N}, \quad \forall x, \qquad n \geq N \implies |\hat{F}_N(x) - F(x)| \leq \varepsilon$$

or equivalently

$$\exists N \in \mathbb{N} \qquad n \geq N \implies \sup_x |\hat{F}_N(x) - F(x)| \leq \varepsilon$$

Theorem

If f is a dominated Caratheodory function on X compact and the sample is iid then we have ULLN on X.

Central Limit Theorem

Theorem

Let $\{\boldsymbol{X}_i\}_{i\in\mathbb{N}}$ be a sequence of rv iid, with finite second order moments. Then we have

$$\sqrt{n}\Big(\underbrace{\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_i}_{M_n} - \mathbb{E}\big[\boldsymbol{X}\big]\Big) \to \mathcal{N}(0, std(\boldsymbol{X}))$$

where the convergence is in law.

Monte-Carlo method

- Let $\{\boldsymbol{X}_i\}_{i\in\mathbb{N}}$ be a sequence of rv iid with finite variance.
- We have $\mathbb{P}\Big(M_N \in \Big[\mathbb{E}\big[\boldsymbol{X}\big] \pm \frac{\Phi^{-1}(p) std(\boldsymbol{X})}{\sqrt{N}}\Big]\Big) \approx p$
- In order to estimate the expectation $\mathbb{E}\big[\boldsymbol{X}\big]$, we can
  - sample N independent realizations of $\boldsymbol{X}$, $\{X_i\}_{i\in[\![1,N]\!]}$
  - compute the empirical mean $M_N = \frac{\sum_{i=1}^N X_i}{N}$, and standard-deviation s_N
  - choose an error level p (e.g. 5%) and compute $\Phi^{-1}(1-p/2)$ (1.96)
  - and we know that, asymptotically, the expectation $\mathbb{E}\big[\boldsymbol{X}\big]$ is in $\Big[M_N \pm \frac{\Phi^{-1}(p)s_N}{\sqrt{N}}\Big]$ with probability (on the sample) $1-p$
- In the case of bounded independent variable we can use Hoeffding

$$\mathbb{P}\Big(\mathbb{E}\big[\boldsymbol{X}\big] \in [M_n \pm t]\Big) \geq 2e^{-\frac{2nt^2}{b-a}}$$

Presentation Outline

The (deterministic) newsboy problem

In the 50's a boy would buy a stock u of newspapers each morning at a cost c, and sell them all day long for a price p. The number of people interested in buying a paper during the day is d. We assume that $0 < c < p$.

How shall we model this ?

- Control $u \in \mathbb{R}^+$
- Cost $L(u) = cu - p\min(u,d)$

Leading to

$$\begin{aligned} \min_u \quad & cu - p\min(u,d) \\ s.t. \quad & u \geq 0 \end{aligned}$$

The (stochastic) newsboy problem

Demand d is unknown at time of purchasing. We model it as a random variable $\boldsymbol{d}$ with known law. Note that

- the control $u \in \mathbb{R}^+$ is deterministic
- the cost is a random variable (depending of $\boldsymbol{d}$). We choose to minimize its expectation.

We consider the following problem

$$\begin{aligned} \min_u \quad & \mathbb{E}\big[cu - p\min(u, \boldsymbol{d})\big] \\ s.t. \quad & u \geq 0 \end{aligned}$$

How can we justify the expectation ?
By law of large number: the Newsboy is going to sell newspaper again and again. Then optimizing the sum over time of its gains is closely related to optimizing the expected gains.

Solving the stochastic newsboy problem

For simplicity assume that the demand $\boldsymbol{d}$ has a continuous density f. Define $J(u)$ the expected "loss" of the newsboy if he bought u newspaper. We have

$$\begin{aligned} J(u) &= \mathbb{E}\big[cu - p\min(u, \boldsymbol{d})\big] \\ &= (c-p)u - p\mathbb{E}\big[\min(0, \boldsymbol{d} - u)\big] \\ &= (c-p)u - p\int_{-\infty}^{u} (x-u)f(x)dx \\ &= (c-p)u - p\Big(\int_{-\infty}^{u} xf(x)dx - u\int_{-\infty}^{u} f(x)dx\Big) \end{aligned}$$

Thus,

$$\begin{aligned} J'(u) &= (c-p) - p\Big(uf(u) - \int_{-\infty}^{u} f(x)dx - uf(u)\Big) \\ &= c - p + pF(u) \end{aligned}$$

where F is the cumulative distribution function (cdf) of $\boldsymbol{d}$. F being non

Newsvendor problem (continued)

We assume that the demand can take value $\{d_i\}_{i\in[\![1,n]\!]}$ with probabilities $\{p_i\}_{i\in[\![1,n]\!]}$.
In this case the stochastic newsvendor problem reads

$$\begin{aligned} \min_u \quad & \sum_{i=1}^{n} p_i\Big(cu - p\min(u, d_i)\Big) \\ s.t. \quad & u \geq 0 \end{aligned}$$

Two-stage newsvendor problem I

We can represent the newsvendor problem in a 2-stage framework.

- Let u_0 be the number of newspaper bought in the morning.
  $\rightsquigarrow$ first stage control
- let u_1 be the number of newspaper sold during the day.
  $\rightsquigarrow$ second stage control

The problem reads

$$\begin{aligned} \min_{u_0, \boldsymbol{u}_1} \quad & \mathbb{E}\Big[cu_0 - p\boldsymbol{u}_1\Big] \\ s.t. \quad & u_0 \geq 0 \\ & \boldsymbol{u}_1 \leq u_0 && \mathbb{P}-as \\ & \boldsymbol{u}_1 \leq \boldsymbol{d} && \mathbb{P}-as \\ & \boldsymbol{u}_1 \preceq \boldsymbol{d} \end{aligned}$$

Two-stage newsvendor problem II

In extensive formulation the problem reads

$$\begin{aligned}
\min_{u_0, \{u_1^i\}_{i \in [\![1,n]\!]}} \quad & \sum_{i=1}^{n} p_i (c u_0 - p u_1^i) \\
s.t. \quad & u_0 \geq 0 \\
& u_1^i \leq u_0 && \forall i \in [\![1,n]\!] \\
& u_1^i \leq d_i && \forall i \in [\![1,n]\!]
\end{aligned}$$

Note that there are as many second-stage control u_1^i as there are possible realization of the demand $\boldsymbol{d}$, but only one first-stage control u_0.

Practical work

- Using julia we are going to model and work around the Newsvendor problem
- Download the files at `https://github.com/leclere/TP-Saclay`
- Start working on the "Newsvendor Problem" up to question 3.

Two-stage stochastic program

V. Leclère

December 8 2021

École des Ponts
ParisTech

Presentation Outline

Presentation Outline

1. Optimization under uncertainty
   - Some considerations on dealing with uncertainty
   - Evaluating a solution
2. Stochastic Programming Approach
   - One-stage Problems
   - Two-stage Problems
   - Recourse assumptions
3. Information and discretization
   - Information Frameworks
   - Sample Average Approximation

A standard optimization problem

$$
\begin{aligned}
\min_{u_0} \quad & L(u_0) \\
s.t. \quad & g(u_0) \leq 0
\end{aligned}
$$

where

- u_0 is the control, or decision.
- L is the cost or objective function.
- $g(u_0) \leq 0$ represent the constraint(s).

The (deterministic) newsboy problem

In the 50's a boy would buy a stock u of newspapers each morning at a cost c, and sell them all day long for a price p. The number of people interested in buying a paper during the day is d. We assume that $0 < c < p$.

How shall we model this ?

- Control $u \in \mathbb{R}^+$
- Cost $L(u) = cu - p\min(u, d)$

Leading to

$$\begin{aligned} \min_{u} \quad & cu - p\min(u, d) \\ s.t. \quad & u \geq 0 \end{aligned}$$

An optimization problem with uncertainty

Adding uncertainty ξ in the mix

$$\begin{aligned} \min_{u_0} \quad & L(u_0, \xi) \\ s.t. \quad & g(u_0, \xi) \leq 0 \end{aligned}$$

Remarks:

- ξ is unknown. Two main ways of modelling it:
  - $\xi \in \Xi$ with a known uncertainty set Ξ, and a pessimistic approach. This is the robust optimization approach (RO).
  - ξ is a random variable with known probability law. This is the Stochastic Programming approach (SP).
- Cost is not well defined.
  - RO : $\max_{\xi \in \Xi} L(u, \xi)$.
  - SP : $\mathbb{E}\big[L(u, \boldsymbol{\xi})\big]$.
- Constraints are not well defined.
  - RO : $g(u, \xi) \leq 0, \qquad \forall \xi \in \Xi$.
  - SP : $g(u, \boldsymbol{\xi}) \leq 0, \qquad \mathbb{P} - a.s.$.

The (stochastic) newsboy problem

Demand d is unknown at time of purchasing. We model it as a random variable $\boldsymbol{d}$ with known law. Note that

- the control $u \in \mathbb{R}^+$ is deterministic
- the cost is a random variable (depending of $\boldsymbol{d}$). We choose to minimize its expectation.

We consider the following problem

$$\begin{aligned} \min_{u} \quad & \mathbb{E}\big[cu - p\min(u, \boldsymbol{d})\big] \\ s.t. \quad & u \geq 0 \end{aligned}$$

How can we justify the expectation ?
By law of large number: the Newsboy is going to sell newspaper again and again. Then optimizing the sum over time of its gains is closely related to optimizing the expected gains.

Solving the stochastic newsboy problem

For simplicity assume that the demand $\boldsymbol{d}$ has a continuous density f. Define $J(u)$ the expected "loss" of the newsboy if he bought u newspaper. We have

$$\begin{aligned} J(u) &= \mathbb{E}\big[cu - p\min(u, \boldsymbol{d})\big] \\ &= (c-p)u - p\mathbb{E}\big[\min(0, \boldsymbol{d} - u)\big] \\ &= (c-p)u - p\int_{-\infty}^{u} (x-u)f(x)dx \\ &= (c-p)u - p\Big(\int_{-\infty}^{u} xf(x)dx - u\int_{-\infty}^{u} f(x)dx\Big) \end{aligned}$$

Thus,

$$J'(u) = (c-p) - p\Big(uf(u) - \int_{-\infty}^{u} f(x)dx - uf(u)\Big) = c - p + pF(u)$$

where F is the cumulative distribution function (cdf) of $\boldsymbol{d}$. F being non decreasing, the optimum control u^* is such that $J'(u^*) = 0$, which is

$$u^* \in F^{-1}\Big(\frac{p-c}{p}\Big)$$

The robust newsboy problem

Demand d is unknown at time of purchasing. We assume that it will be in the set $[\underline{d}, \overline{d}]$.
The robust problem consist in solving

$$\begin{aligned} \min_u \quad & \max_{d \in [\underline{d}, \overline{d}]} cu - p\min(u,d) \\ s.t. \quad & u \geq 0 \end{aligned}$$

By monotonicity it is equivalent to

$$\begin{aligned} \min_u \quad & cu - p\min(u,\underline{d}) \\ s.t. \quad & u \geq 0 \end{aligned}$$

Alternative cost functions I

- When the cost $L(u,\xi)$ is random it might be natural to want to minimize its expectation $\mathbb{E}\big[L(u,\xi)\big]$.
- This is even justified if the same problem is solved a large number of time (Law of Large Number).
- In some cases the expectation is not really representative of your risk attitude. Lets consider two examples:
  - Are you ready to pay \$1000 to have one chance over ten to win \$10000 ?
  - You need to be at the airport in 1 hour or you miss your flight, you have the choice between two mean of transport, one of them take surely 50', the other take 40' four times out of five, and 70' one time out of five.

Alternative cost functions II

Here are some cost functions you might consider

- Probability of reaching a given level of cost : $\mathbb{P}(L(u,\xi) \leq 0)$
- Value-at-Risk of costs $V@R_\alpha(L(u,\xi))$, where for any real valued random variable $\boldsymbol{X}$,

$$V@R_\alpha(\boldsymbol{X}) := \inf_{t\in\mathbb{R}} \Big\{\mathbb{P}(\boldsymbol{X} \geq t) \leq \alpha\Big\}.$$

  In other word there is only a probability of α of obtaining a cost worse than $V@R_\alpha(\boldsymbol{X})$.
- Average Value-at-Risk of costs $AV@R_\alpha(L(u,\xi))$, which is the expected cost over the α worst outcomes.

Alternative constraints I

- The natural extension of the deterministic constraint $g(u,\xi) \leq 0$ to $g(u,\xi) \leq 0 \ \mathbb{P} - as$ can be extremely conservative, and even often without any admissible solutions.
- For example, if u is a level of production that need to be greater than the demand. In a deterministic setting the realized demand is equal to the forecast. In a stochastic setting we add an error to the forecast. If the error is unbouded (e.g. Gaussian) no control u is admissible.

Alternative constraints II

Here are a few possible constraints

- $\mathbb{E}\big[g(u,\xi)\big] \leq 0$, for quality of service like constraint.
- $\mathbb{P}\big(g(u,\xi) \leq 0\big) \geq 1-\alpha$ for chance constraint. Chance constraint is easy to present, but might lead to misconception as nothing is said on the event where the constraint is not satisfied.
- $AV@R_\alpha\big(g(u,\xi)\big) \leq 0$

Presentation Outline

Computing expectation

- Computing an expectation $\mathbb{E}\big[L(u,\xi)\big]$ for a given u is costly.
- If ξ is a r.v. with known law admitting a density, $\mathbb{E}\big[L(u,\xi)\big]$ is a (multidimensional) integral.
- If ξ is a r.v. with known discrete law, $\mathbb{E}\big[L(u,\xi)\big]$ is a sum over all possible realizations of ξ, which can be huge.
- If ξ is a r.v. that can be simulated but with unknown law, $\mathbb{E}\big[L(u,\xi)\big]$ cannot be computed exactly.

Solution : use Law of Large Number (LLN) and Central Limit Theorem (CLT).

- Draw $N \simeq 1000$ realization of ξ.
- Compute the sample average $\frac{1}{N}\sum_{s=1}^{N} L(u,\xi_s)$.
- Use CLT to give an asymptotic confidence interval of the expectation.

This is known as the Monte-Carlo method.

Consequence : evaluating a solution is difficult

- In stochastic optimization even evaluating the value of a solution can be difficult an require approximate methods.
- The same holds true for checking admissibility of a candidate solution.
- It is even more difficult to obtain first order informations (gradient).

Standard solution : sampling and solving the sampled problem (Sample Average Approximation).

Recall on CLT

- Let $\{C_i\}_{i\in\mathbb{N}}$ be a sequence of identically distributed random variables with finite variance.
- Then the Central Limit Theorem ensures that
$$\sqrt{N}\Big(\frac{\sum_{i=1}^N \boldsymbol{C}_i}{N} - \mathbb{E}[\boldsymbol{C}_1]\Big) \Longrightarrow G \sim \mathcal{N}(0, Var[\boldsymbol{C}_1]) ,$$
where the convergence is in law.
- In practice it is often used in the following way. Asymptotically,
$$\mathbb{P}\Big(\mathbb{E}[C_1] \in \Big[\bar{\boldsymbol{C}}_N - \frac{1.96\boldsymbol{\sigma}_N}{\sqrt{N}}, \bar{\boldsymbol{C}}_N + \frac{1.96\boldsymbol{\sigma}_N}{\sqrt{N}}\Big]\Big) \simeq 95\% ,$$
where $\bar{\boldsymbol{C}}_N = \frac{\sum_{i=1}^N \boldsymbol{C}_i}{N}$ is the empirical mean and $\boldsymbol{\sigma}_N = \sqrt{\frac{\sum_{i=1}^N (\boldsymbol{C}_i - \bar{\boldsymbol{C}}_N)^2}{N-1}}$ the empirical standard deviation.

Optimization problem and simulator

- Generally speaking stochastic optimization problem are not well posed and often need to be approximated before solving them.
- Good practice consists in defining a simulator, i.e. a representation of the "real problem" on which solution can be tested.
- Then find a candidate solution by solving an (or multiple) approximated problem.
- Finally evaluate the candidate solutions on the simulator. The comparison can be done on more than one dimension (e.g. constraints, risk...)

Conclusion

When addressing an optimization problem under uncertain one has to consider carefully

- How to model uncertainty ? (random variable or uncertainty set)
- How to represent your attitude toward risk ? (expectation, probability level,...)
- How to include constraints ?
- What is your information stucture ? (More on that later)
- Set up a simulator and evaluate your solutions.

Presentation Outline

One-Stage Problems

Assume that ξ has a discrete distribution [1], with $\mathbb{P}(\xi = \xi_s) = \pi^s > 0$ for $s \in [\![1, S]\!]$. Then, the one-stage problem

$$\begin{aligned} \min_{u_0} \quad & \mathbb{E}\Big[L(u_0, \xi)\Big] \\ s.t. \quad & g(u_0, \xi) \leq 0, \qquad \mathbb{P} - a.s \end{aligned}$$

can be written

$$\begin{aligned} \min_{u_0} \quad & \sum_{s=1}^{S} \pi^s L(u_0, \xi_s) \\ s.t \quad & g(u_0, \xi_s) \leq 0, \qquad \forall s \in [\![1, S]\!]. \end{aligned}$$

[1] If the distribution is continuous we can sample and work on the sampled distribution, this is called the Sample Average Approximation approach with lots of guarantee and results

Newsvendor problem (continued)

We assume that the demand can take value $\{d^s\}_{s\in[\![1,S]\!]}$ with probabilities $\{\pi^s\}_{s\in[\![1,S]\!]}$.
In this case the stochastic newsvendor problem reads

$$\begin{aligned} \min_{u} \quad & \sum_{s=1}^{S} \pi^s \Big(cu - p \min(u, d^s)\Big) \\ s.t. \quad & u \geq 0 \end{aligned}$$

Presentation Outline

Recourse Variable

In most problem we can make a correction u_1 once the uncertainty is known:

$$u_0 \rightsquigarrow \xi_1 \rightsquigarrow u_1.$$

As the recourse control u_1 is a function of ξ it is a random variable. The two-stage optimization problem then reads

$$\begin{aligned} \min_{u_0, \boldsymbol{u}_1} \quad & \mathbb{E}\Big[L(u_0, \xi, \boldsymbol{u}_1)\Big] \\ s.t. \quad & g(u_0, \xi, \boldsymbol{u}_1) \leq 0, \qquad \mathbb{P} - a.s \\ & \boldsymbol{u}_1 \preceq \xi \end{aligned}$$

- u_0 is called a first stage control
- u_1 is called a second stage (or recourse) control

Two-stage Problem

The extensive formulation of

$$\begin{aligned} \min_{u_0,\boldsymbol{u}_1} \quad & \mathbb{E}\Big[L(u_0,\boldsymbol{\xi},\boldsymbol{u}_1)\Big] \\ s.t. \quad & g(u_0,\boldsymbol{\xi},\boldsymbol{u}_1) \le 0, \qquad \mathbb{P}-a.s \\ & \boldsymbol{u}_1 \preceq \boldsymbol{\xi} \end{aligned}$$

is

$$\begin{aligned} \min_{u_0,\{u_1^s\}_{s\in[\![1,S]\!]}} \quad & \sum_{s=1}^{S} p^s L(u_0,\xi^s,u_1^s) \\ s.t \quad & g(u_0,\xi^s,u_1^s) \le 0, \qquad \forall s \in [\![1,S]\!]. \end{aligned}$$

It is a deterministic problem that can be solved with standard tools or specific methods.

Two-stage newsvendor problem I

We can represent the newsvendor problem in a 2-stage framework.

- Let u_0 be the number of newspaper bought in the morning.
  $\rightsquigarrow$ first stage control
- let u_1 be the number of newspaper sold during the day.
  $\rightsquigarrow$ second stage control

The problem reads

$$\begin{aligned} \min_{u_0,\boldsymbol{u}_1} \quad & \mathbb{E}\Big[cu_0 - p\boldsymbol{u}_1\Big] \\ s.t. \quad & u_0 \ge 0 \\ & \boldsymbol{u}_1 \le u_0 && \mathbb{P}-as \\ & \boldsymbol{u}_1 \le \boldsymbol{d} && \mathbb{P}-as \\ & \boldsymbol{u}_1 \preceq \boldsymbol{d} \end{aligned}$$

Two-stage newsvendor problem II

In extensive formulation the problem reads

$$\begin{aligned} \min_{u_0,\{u_1^s\}_{s\in[\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s (cu_0 - pu_1^s) \\ s.t. \quad & u_0 \ge 0 \\ & u_1^s \le u_0 && \forall s \in [\![1,S]\!] \\ & u_1^s \le d^s && \forall s \in [\![1,S]\!] \end{aligned}$$

Note that there are as many second-stage control u_1^s as there are possible realization of the demand $\boldsymbol{d}$, but only one first-stage control u_0.

Presentation Outline

Time decomposition of the problem

We presented the generic two-stage problem as

$$\begin{aligned} \min_{u_0, \boldsymbol{u}_1} \quad & \mathbb{E}\Big[L(u_0, \boldsymbol{\xi}, \boldsymbol{u}_1)\Big] \\ s.t. \quad & g(u_0, \boldsymbol{\xi}, \boldsymbol{u}_1) \leq 0, \qquad \mathbb{P}-a.s \\ & \boldsymbol{u}_1 \preceq \boldsymbol{\xi} \end{aligned}$$

With $L(u_0, \xi, u_1) = L_0(u_0) + L_1(u_0, \xi, u_1)$, it can also be written as

$$\begin{aligned} \min_{u_0} \quad & L_0(u_0) + \mathbb{E}\Big[\tilde{Q}(u_0, \boldsymbol{\xi})\Big] \qquad \text{first stage problem} \\ s.t. \quad & g_0(u_0) \leq 0 \end{aligned}$$

where

$$\begin{aligned} \tilde{Q}(u_0, \xi) := \min_{u_1} \quad & L_1(u_0, \xi, u_1) \qquad \text{second stage problem} \\ s.t. \quad & g_1(u_0, \xi, u_1) \leq 0 \end{aligned}$$

The reformulation always exists, but is not unique

Admissible set

For a given decomposition, we set

$$\begin{aligned} U_0 :=& \{u_0 \in \mathbb{R}^{n_0} \quad | \quad g_0(u_0) \leq 0\} \\ \widetilde{U}_1(u_0, \xi) :=& \{u_1 \in \mathbb{R}^{n_1} \quad | \quad g_1(u_0, \xi, u_1) \leq 0\} \end{aligned}$$

Note that

- $\widetilde{U}_1(u_0, \xi)$ is the set of admissible solutions of the second stage problem
- U_0 contains the set of admissible solutions of the first stage problem

- We say that we are in a complete recourse framework, if for all $u_0 \in U_0$, and almost-all possible outcome $\boldsymbol{\xi}$, every control $\boldsymbol{u}_1$ is admissible, i.e.,
$$\mathbb{P}\big(\widetilde{U}_1(u_0, \boldsymbol{\xi}) = \mathbb{R}^{n_1}\big) = 1, \quad \forall u_0 \in U_0.$$
- We say that we are in a relatively complete recourse framework, if for all $u_0 \in U_0$, and almost-all possible outcome ξ, there exists a control $\boldsymbol{u}_1$ that is admissible, i.e.,
$$\mathbb{P}\big(\widetilde{U}_1(u_0, \boldsymbol{\xi}) \neq \emptyset\big) = 1, \quad \forall u_0 \in U_0.$$
- We say that we are in an extended relatively complete recourse framework, if there exists $\varepsilon > 0$ such that, for all $u_0 \in U_0 + \varepsilon B$, and almost-all possible outcome ξ, there exists a control $\boldsymbol{u}_1$ that is admissible, i.e.,
$$\mathbb{P}\big(\widetilde{U}_1(u_0, \boldsymbol{\xi}) \neq \emptyset\big) = 1, \quad \forall u_0 \in U_0 + \varepsilon B.$$

Obtaining relatively complete recourse

Assume that the two-stage program is given by

$$\min_{u_0 \in U_0} \Big\{L_0(u_0) + \mathbb{E}\big[\tilde{Q}(u_0, \boldsymbol{\xi})\big]\Big\} \quad \text{and} \quad \tilde{Q}(u_0, \boldsymbol{\xi}) := \min_{u_1 \in \widetilde{U}_1(u_0, \xi)} L_1(u_0, \xi, u_1)$$

with finite value, but not necessarily relatively complete recourse.
Then the program is equivalent to

$$\min_{u_0 \in U_0 \cap U_0^{ind}} \Big\{L_0(u_0) + \mathbb{E}\big[\tilde{Q}(u_0, \boldsymbol{\xi})\big]\Big\} \quad \text{and} \quad \tilde{Q}(u_0, \boldsymbol{\xi}) := \min_{u_1 \in \widetilde{U}_1(u_0, \xi)} L_1(u_0, \xi, u_1)$$

where U_0^{ind} is the set of induced constraints given by

$$U_0^{ind} = \Big\{u_0 \in \mathbb{R}^{n_0} \quad | \quad \mathbb{P}\big(\widetilde{U}_1(u_0, \boldsymbol{\xi}) \neq \emptyset\big) = 1\Big\},$$

and with this formulation we are in a relatively complete recourse framework.

Presentation Outline

Two-stage framework : three information models

Consider the problem

$$\min_{\boldsymbol{u}_0, \boldsymbol{u}_1} \mathbb{E}\big[L(\boldsymbol{u}_0, \boldsymbol{\xi}, \boldsymbol{u}_1)\big]$$

- Open-Loop case : $\boldsymbol{u}_0$ and $\boldsymbol{u}_1$ are deterministic. In this case both controls are choosen without any knowledge of the alea $\boldsymbol{\xi}$. The set of control is small, and an optimal control can be found through specific method (e.g. Stochastic Gradient).
- Two-Stage case : $\boldsymbol{u}_0$ is deterministic and $\boldsymbol{u}_1$ is measurable with respect to $\boldsymbol{\xi}$. This is the problem tackled by the Stochastic Programming case.
- Anticipative case : $\boldsymbol{u}_0$ and $\boldsymbol{u}_1$ are measurable with respect to $\boldsymbol{\xi}$. This case consists in solving one deterministic problem per possible outcome of the alea, and taking the expectation of the value of this problems.

Splitted formulation

The extended formulation (in a compact way)

$$\begin{aligned} \min_{u_0, \{u_1^s\}_{s\in[\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s L(u_0, \xi^s, u_1^s) \\ s.t \quad & g(u_0, \xi^s, u_1^s) \leq 0, \qquad && \forall s \in [\![1,S]\!]. \end{aligned}$$

Can be written in a splitted formulation

$$\begin{aligned} \min_{\bar{u}_0, u_0^s, \{u_1^s\}_{s\in[\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s L(u_0^s, \xi^s, u_1^s) \\ s.t \quad & g(u_0^s, \xi^s, u_1^s) \leq 0, \qquad && \forall s \in [\![1,S]\!] \\ & u_0^s = u_0^{s'} && \forall s, s' \end{aligned}$$

Splitted formulation

The extended formulation (in a compact way)

$$\begin{aligned} \min_{u_0, \{u_1^s\}_{s\in[\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s L(u_0, \xi^s, u_1^s) \\ s.t \quad & g(u_0, \xi^s, u_1^s) \leq 0, \qquad && \forall s \in [\![1,S]\!]. \end{aligned}$$

Can be written in a splitted formulation

$$\begin{aligned} \min_{\bar{u}_0, u_0^s, \{u_1^s\}_{s\in[\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s L(u_0^s, \xi^s, u_1^s) \\ s.t \quad & g(u_0^s, \xi^s, u_1^s) \leq 0, \qquad && \forall s \in [\![1,S]\!] \\ & u_0^s = \sum_{s'} \pi^{s'} u_0^{s'} && \forall s \end{aligned}$$

Information models for the Newsvendor I

Open-loop :

$$\begin{aligned}
\min_{u_0,u_1} \quad & \sum_{s=1}^{S} \pi^s (cu_0 - pu_1) \\
s.t. \quad & u_0 \geq 0 \\
& u_1 \leq u_0 \\
& u_1 \leq d^s & \forall s \in [\![1, S]\!]
\end{aligned}$$

Information models for the Newsvendor II

Two-stage :

$$\begin{aligned}
\min_{u_0,\{u_1^s\}_{s\in[\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s (cu_0 - pu_1^s) \\
s.t. \quad & u_0 \geq 0 \\
& u_1^s \leq u_0 & \forall s \in [\![1, S]\!] \\
& u_1^s \leq d_s & \forall s \in [\![1, S]\!]
\end{aligned}$$

Information models for the Newsvendor III

Anticipative :

$$\begin{aligned}
\min_{\{u_0^s,u_1^s\}_{s\in[\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s (cu_0^s - pu_1^s) \\
s.t. \quad & u_0^s \geq 0 & \forall s \in [\![1, S]\!] \\
& u_1^s \leq u_0 & \forall s \in [\![1, S]\!] \\
& u_1^s \leq d^s & \forall s \in [\![1, S]\!]
\end{aligned}$$

Comparing the information models

The three information models can be written this way :

$$\begin{aligned}
\min_{\{u_0^s,u_1^s\}_{s\in[\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s (cu_0^s - pu_1^s) \\
s.t. \quad & u_0^s \geq 0 & \forall s \in [\![1, S]\!] \\
& u_1^s \leq u_0 & \forall i \in [\![1, S]\!] \\
& u_1^s \leq d^s & \forall i \in [\![1, s]\!] \\
& u_0^s = u_0^{s'} & \forall s, s' \\
& u_1^s = u_1^{s'} & \forall s, s'
\end{aligned}$$

Hence, by simple comparison of constraints we have

$$V^{anticipative} \leq V^{2-stage} \leq V^{OL}.$$

Value of information

- The Expected Value of Perfect Information (EVPI) is defined as
$$EVPI = v^{2-stage} - v^{anticipative} \geq 0.$$
- Its the maximum amount of money you can gain by getting more information (e.g. incorporating better statistical model in your problem)
- The Value of Stochastic Solution is defined as
$$VSS = v^{OL} - v^{2-stage} \geq 0.$$
- The expected value problem is the value of the deterministic problem where the randomness is replaced by its expectation
$$v^{EV} = \min_{u_0, u_1} L(u_0, \mathbb{E}[\xi], u_1).$$
- If (u_0^{EV}, u_1^{EV}) is the solution of the EV problem, then $\mathbb{E}\left[L(u_0^{EV}, \xi, u_1^{EV})\right]$, is known as Expected Value of Expected Value problem v^{EEV}.

Comparison and convexity

- Without assumption we have
$$v^{EEV} \geq v^{OL} \geq v^{2-stage} \geq v^{anticipative}$$
- If additionally L is jointly convex we have
$$\begin{aligned} v^{anticipative} &= \mathbb{E}\left[L(u_0^{\xi}, \xi, u_1^{\xi})\right] \\ &\geq L(\mathbb{E}[u_0^{\xi}], \mathbb{E}[\xi], \mathbb{E}[u_1^{\xi}]) \\ &\geq L(u_0^{EV}, \mathbb{E}[\xi], u_1^{EV}) = v^{EV} \end{aligned}$$
- Hence, under convexity we have,
$$v^{EEV} \geq v^{OL} \geq v^{2-stage} \geq v^{anticipative} \geq v^{EV}$$

Solving the problems

- The solution of v^{EEV} is easy to find (one deterministic problem), and its value is obtained by Monte-Carlo.
- v^{OL} can be approximated through specific methods (e.g. SG).
- $v^{2-stage}$ is obtained through Stochastic Programming specific methods. There are two main approaches:
  - Lagrangian decomposition methods (like Progressive-Hedging algorithm).
  - Benders decomposition methods (like L-shaped or nested-decomposition methods).
- $v^{anticipative}$ is difficult to compute exactly but can be estimated through Monte-Carlo approach by drawing a reasonable number of realizations of ξ, solving the deterministic problem for each realization ξ_i and taking the means of the value of the deterministic problem.
- v^{EV} is easy to compute, but is usefull only in the convex case.

Presentation Outline

How to deal with continuous distributions ?

Recall that if $\boldsymbol{\xi}$ as finite support we rewrite the 2-stage problem

$$\begin{aligned} \min_{u_0,\boldsymbol{u}_1} \quad & \mathbb{E}\Big[L(u_0,\boldsymbol{\xi},\boldsymbol{u}_1)\Big] \\ s.t. \quad & g(u_0,\boldsymbol{\xi},\boldsymbol{u}_1) \leq 0, \qquad \mathbb{P}-a.s \end{aligned}$$

as

$$\begin{aligned} \min_{u_0,\{u_1^s\}_{s\in[\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s L(u_0,\xi^s,u_1^s) \\ s.t \quad & g(u_0,\xi^s,u_1^s) \leq 0, \qquad \forall s \in [\![1,S]\!]. \end{aligned}$$

If we consider a continuous distribution (e.g. a Gaussian), we would need an infinite number of recourse variables to obtain an extensive formulation.

Simplest idea: sample $\boldsymbol{\xi}$

First consider the one-stage problem

$$\min_{u\in U} \mathbb{E}\big[L(u,\boldsymbol{\xi})\big] \qquad (\mathcal{P})$$

- Draw a sample $(\xi^1,\dots,\xi^N)$ (in a i.i.d setting with law $\boldsymbol{\xi}$).
- Consider the empirical probability $\hat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^N \delta_{\xi^i}$.
- Replace $\mathbb{P}$ by $\hat{\mathbb{P}}_N$ to obtain a finite-dimensional problem that can be solved.
- This means solving

$$\min_{u\in U} \frac{1}{N}\sum_{i=1}^{N} L(u,\xi^i) \qquad (\mathcal{P}_N)$$

- We denote by $\hat{\boldsymbol{v}}_N$ (resp. v^*) the value of $(\mathcal{P}_N)$ (resp. $(\mathcal{P})$), and $\boldsymbol{S}_n$ the set of optimal solutions (resp. S^*).

Biased estimator

Generically speaking the estimators of the minimum are biased

$$\mathbb{E}\big[\hat{\boldsymbol{v}}_N\big] \leq \mathbb{E}\big[\hat{\boldsymbol{v}}_{N+1}\big] \leq v^*$$

proof :

- Let $(\boldsymbol{\xi}_i)_{i\in\mathbb{N}}$ be a sequence of iid copies of $\boldsymbol{\xi}$
- Set $J(u) := \mathbb{E}\big[L(u,\boldsymbol{\xi})\big]$, $\boldsymbol{J}_N(u) := \frac{1}{N}\sum_{i=1}^N L(u,\boldsymbol{\xi}_i)$
- We have, for every $u'\in U$, $\boldsymbol{J}_N(u') \geq \inf_{u\in U} \boldsymbol{J}_N(u)$.
- Taking the expectation yields,

$$J(u') = \mathbb{E}\Big[\boldsymbol{J}_N(u')\Big] \geq \mathbb{E}\Big[\inf_{u\in U} \boldsymbol{J}_N(u)\Big] = \mathbb{E}\big[\hat{\boldsymbol{v}}_N\big].$$

- We now take the infimum over $u'\in U$, to obtain

$$v^* = \inf_{u'\in U} J(u') \geq \mathbb{E}\big[\hat{\boldsymbol{v}}_N\big].$$

Decreasing bias

We now show that the bias is monotonically decreasing. Notice that

$$\boldsymbol{J}_{N+1}(u) = \frac{1}{N+1}\sum_{i=1}^{N+1}\Big[\frac{1}{N}\sum_{j\neq i} L(u,\boldsymbol{\xi}_j)\Big].$$

Hence,

$$\begin{aligned} \mathbb{E}\big[\hat{\boldsymbol{v}}_{N+1}\big] &= \mathbb{E}\Big[\inf_{u\in U} \boldsymbol{J}_{N+1}(u)\Big] = \mathbb{E}\Big[\inf_{u\in U} \frac{1}{N+1}\sum_{i=1}^{N+1}\Big[\frac{1}{N}\sum_{j\neq i} L(u,\boldsymbol{\xi}_j)\Big]\Big] \\ &\geq \mathbb{E}\Big[\frac{1}{N+1}\sum_{i=1}^{N+1}\inf_{u_i\in U}\Big[\frac{1}{N}\sum_{j\neq i} L(u_i,\boldsymbol{\xi}_j)\Big]\Big] \\ &= \frac{1}{N+1}\sum_{i=1}^{N+1}\mathbb{E}\Big[\inf_{u_i\in U}\Big[\frac{1}{N}\sum_{j\neq i} L(u_i,\boldsymbol{\xi}_j)\Big]\Big] \\ &= \frac{1}{N+1}\sum_{i=1}^{N+1}\mathbb{E}\big[\hat{\boldsymbol{v}}_N\big] = \mathbb{E}\big[\hat{\boldsymbol{v}}_N\big] \end{aligned}$$

Consistency of estimator

Definition

Let $\{\boldsymbol{f}_N\}_{N\in\mathbb{N}}$ be a sequence of random functions mapping X into $\mathbb{R}$. We say that $\boldsymbol{f}_N$ converges almost surely toward $f : X \mapsto \mathbb{R}$ uniformly on X, if

$$\forall \varepsilon > 0, \quad \exists N \in \mathbb{N}, \quad \forall n \geq N, \qquad \mathbb{P}\Big(\sup_{x\in X} |\boldsymbol{f}_n(x) - f(x)| \leq \varepsilon \Big) = 1.$$

Theorem (Consistency of SAA)

If $\boldsymbol{J}_{N+1}$ converges almost surely toward J uniformly on U, then $\hat{v}_N$ converges almost surely toward $v^\sharp$.

Theorem (Convergence in the compact case)

Assume that

1. *U is compact non empty,*
2. *$\boldsymbol{J}_N$ converges uniformly on U toward J,*
3. *$\boldsymbol{U}_N^\sharp$ in non-empty,*
4. *J is continuous on U.*

Then,

- $\boldsymbol{v}_N^\sharp \to v^\sharp \quad \mathbb{P}^N$*-a.s.,*
- $\mathbb{D}\big(\boldsymbol{U}_n^\sharp, U^\sharp\big) \to 0 \quad \mathbb{P}^N$*-a.s.*

1. can be relaxed in a compact set containing optimal solution
2. usually comes from the uniform law of large number
3. can be obtained if J_N is lower semi-continuous with some non-empty but uniformly bounded level set
4. often rely on a domination theorem.

Theorem (Convergence in the convex case)

Assume that

1. *j is a.s. convex l.s.c.*
2. *U is closed convex*
3. *J is l.s.c, and there exists $u \in U$ such that a neighboorhoud of u is contained in $\mathrm{dom}(J)$*
4. *$S \neq \emptyset$ is bounded*
5. *the LLN holds*

Then,

- $\boldsymbol{v}_N^\sharp \to v^\sharp \quad \mathbb{P}^N$*-a.s.,*
- $\mathbb{D}\big(\boldsymbol{U}_n^\sharp, U^\sharp\big) \to 0 \quad \mathbb{P}^N$*-a.s.*

Theorem (Convergence speed)

Assume that,

- $\mathbb{E}\big[j(u,\boldsymbol{\xi})^2\big] < \infty$,
- $u \mapsto j(u,\xi)$ *is Lipschitz-continuous with constant* $L(\xi)$ *with* $\mathbb{E}\big[L(\boldsymbol{\xi})^2\big] < \infty$,
- U *is compact,* $U^\sharp = \{u^\sharp\}$.

Then,

- $\boldsymbol{v}_N^\sharp = \boldsymbol{J}_N(u^\sharp) + o(\frac{1}{\sqrt{N}})$,
- $\sqrt{N}\big(\boldsymbol{v}_N^\sharp - v^\sharp\big) \Rightarrow \mathcal{N}(0, \sigma^2(u^\sharp))$,

where $\sigma^2(u) := \mathbb{E}\Big[\big(j(u,\boldsymbol{\xi}) - \mathbb{E}\big[j(u,\boldsymbol{\xi})\big]\big)^2\Big]$.

The unicity of solution assumption can be relaxed.
Good reference for precise results : Lectures on Stochastic Programming (Dentcheva, Ruszczynski, Shapiro) chap. 5.

Stochastic Dynamic Programming
Bellman Operators

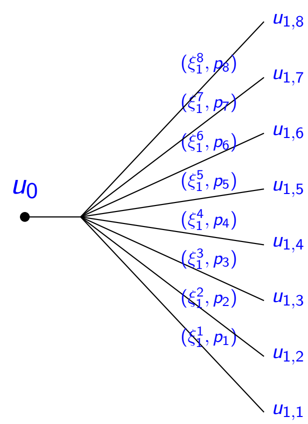
V. Leclère

December 15, 2021

Contents

Contents

Where do we come from: two-stage programming



- We take decisions in two stages

$$u_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow \boldsymbol{u}_1 \ ,$$

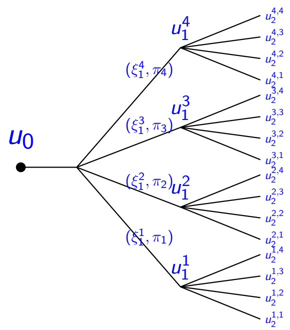
with $\boldsymbol{u}_1$: recourse decision .

- On a tree, it resumes to solve the extensive formulation:

$$\min_{u_0, u_{1,s}} \sum_{s \in \mathbb{S}} \pi^s \Big[\big\langle c_s \, , u_0 \big\rangle + \big\langle p_s \, , u_{1,s} \big\rangle \Big] \ .$$

We have as many $u_{1,s}$ as scenarios!

Extending two-stage to multistage programming



$$\min_{\boldsymbol{u}} \mathbb{E}\big(j(\boldsymbol{u}, \boldsymbol{\xi})\big)$$

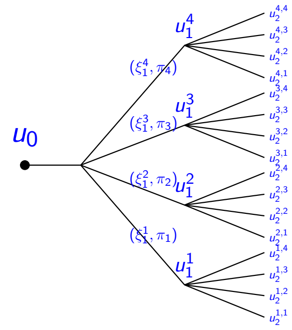
$$\boldsymbol{U} = (\boldsymbol{u}_0, \cdots, \boldsymbol{U}_T)$$

$$\boldsymbol{\xi} = (\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_T)$$

We take decisions in T stages

$$\boldsymbol{\xi}_0 \rightsquigarrow \boldsymbol{u}_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow \boldsymbol{u}_1 \rightsquigarrow \cdots \rightsquigarrow \boldsymbol{\xi}_T \rightsquigarrow \boldsymbol{u}_T .$$

Multistage extensive formulation approach



Assume that $\xi_t \in \mathbb{R}^{n_\xi}$ can take n_ξ values and that $U_t(x) \subset \mathbb{R}^{n_u}$.

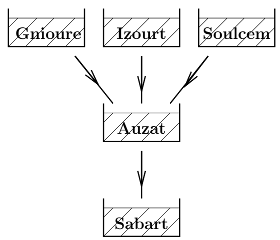
Then, considering the extensive formulation approach, we have

- n_ξ^T scenarios.
- $(n_\xi^{T+1} - 1)/(n_\xi - 1)$ nodes in the tree.
- Number of variables in the optimization problem is roughly $n_u \times (n_\xi^{T+1} - 1)/(n_\xi - 1) \approx n_u n_\xi^T$.

The complexity grows exponentially with the number of stage. :-(

A way to overcome this issue is to compress information!

Illustrating extensive formulation with the damsvalley example



- 5 interconnected dams
- 5 controls per timesteps
- 52 timesteps (one per week, over one year)
- $n_\xi = 10$ noises for each timestep

We obtain 10^{52} scenarios, and $\approx 5.10^{52}$ constraints in the extensive formulation ...

Estimated storage capacity of the Internet: 10^{24} bytes.

Contents

Optimization Problem

We want to solve the following optimization problem

$$\min \quad \mathbb{E}\Big[\sum_{t=0}^{T-1} L_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1}) + K(\boldsymbol{x}_T)\Big] \tag{1a}$$

$$s.t. \quad \boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1}), \qquad \boldsymbol{x}_0 = \boldsymbol{\xi}_0 \tag{1b}$$

$$\boldsymbol{u}_t \in \mathcal{U}_t(\boldsymbol{x}_t) \tag{1c}$$

$$\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_t := \sigma(\boldsymbol{\xi}_0, \cdots, \boldsymbol{\xi}_t) \tag{1d}$$

Where

- constraint (1b) is the dynamic of the system ;
- constraint (1c) refer to the constraint on the controls;
- constraint (1d) is the information constraint : $\boldsymbol{u}_t$ is choosen knowing the realisation of the noises $\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_t$ but without knowing the realisation of the noises $\boldsymbol{\xi}_{t+1}, \ldots, \boldsymbol{\xi}_{T-1}$.

Information structure I

In Problem (1), constraint (1d) is the information constraint.
There are different possible information structure.

- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_0$, the problem is open-loop, as the controls are choosen without knowledge of the realisation of any noise.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_t$, the problem is said to be in decision-hazard structure as decision $\boldsymbol{u}_t$ is chosen without knowing $\boldsymbol{\xi}_{t+1}$.
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_{t+1}$, the problem is said to be in hazard-decision structure as decision $\boldsymbol{u}_t$ is chosen with knowledge of $\boldsymbol{\xi}_{t+1}$ (in which case we have $\boldsymbol{u}_t \in \mathcal{U}_t(\boldsymbol{x}_t, \boldsymbol{\xi}_{t+1})$)
- If constraint (1d) reads $\sigma(\boldsymbol{u}_t) \subset \mathcal{F}_{T-1}$, the problem is said to be anticipative as decision $\boldsymbol{u}_t$ is chosen with knowledge of all the noises.

Information structure II

Be careful when modeling your information structure:

- Open-loop information structure might happen in practice (you have to decide on a planning and stick to it). If the problem does not require an open-loop solution then it might be largely suboptimal (imagine driving a car eyes closed...). In any case it yields an upper-bound of the problem.
- In some cases decision-hazard and hazard-decision are both approximation of the reality. Hazard-decision yield a lower value then decision-hazard.
- Anticipative structure is never an accurate modelization of the reality. However it can yield a lower-bound of your optimization problem relying on deterministic optimization and Monte-Carlo.

We are going to assume Hazard-Decision structure

Contents

Bounds and heuristics

- Due to the size of the extensive formulation of multistage programm we cannot hope to numerically solve them without further assumptions on the problem.
- However, there are a few ideas we can use to get
  - heuristics policies (that is non-optimal but "reasonable" solution), and thus upper bounds (estimated by Monte Carlo)
  - lower bounds to guarantee quality of heuristics
- We can get these through:
  - deterministic approximation
  - two-stage approximations
  - linear decision rules
  - ...

Anticipative lower bound

- If we relax the measurability constraint by assuming that u_t is measurable w.r.t $\sigma(\xi_0, \dots, \xi_T)$, that is knows the whole scenario we get the anticipative solution :

$$\mathbb{E}\Big[\min_{\boldsymbol{u}} \sum_{t=0}^{T} L_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1}) + K(x_T)\Big]$$

- This can be computed by solving $|\Omega|$ deterministic optimization problems.
- As $|\Omega|$ is often too large, this lower bound is estimated by Monte-Carlo :
  - draw N scenarios (e.g. $N = 1000$)
  - solve each deterministic problem
  - average their value to estimate the lower bound

Deterministic heuristic

- A natural heuristic consists in looking for a deterministic solution (we stick to the plan).
- The first heuristic consists in simply replacing $\boldsymbol{\xi}_{t+1}$ by an estimation (often its expectation $\mathbb{E}[\boldsymbol{\xi}_{t+1}]$), and solve a deterministic problem.
- A more advanced heuristic consists in looking for optimal open-loop solution (e.g. by using Stochastic Gradient algorithms).

Model Predictive Control

- A very classical heuristic, often very efficient if the stochasticity is not too important is the so-called Model Predictive Control (MPC).
- MPC works in the following way :
  - at time t_0, being in x_0, solve the deterministic problem

$$\begin{aligned}
\min \quad & \sum_{t=t_0}^{T-1} L_t\big(x_t, u_t, \hat{\xi}_{t+1}\big) + K\big(x_T\big) \\
s.t. \quad & x_{t+1} = f_t\big(x_t, u_t, \hat{\xi}_{t+1}\big), \qquad x_{t_0} = x_0 \\
& u_t \in \mathcal{U}_t(x_t)
\end{aligned}$$

    where $\hat{\xi}_t$ is your best estimate of $\boldsymbol{\xi}_t$ (its expectation by default)
  - apply u_{t_0} and get x_{t_0+1}
  - update your estimation of $\boldsymbol{\xi}$, set $x_0 = x_{t_0+1}$ and $t_0 = t_0 + 1$

Two-stage lower-bound

- We can refine the anticipative lower bound by relaxing all measurability constraint except the one on u_0.
- We thus obtain a two-stage programm u_0 being the first stage control, and all the other u_t knowing the whole scenario are second-stage variable.
- We thus have a 2-stage program with $|\Omega|$ second stage (vector) variables whose value is a lower-bound to the original problem.
- This value can be approximated by SAA :
  - draw N scenarios
  - write a 2-stage programm with these scenarios, with u_0 as first stage control and $(u_1, \dots, u_{T-1})$ as recourse
  - its value is an estimation of the 2-stage lower-bound

2-stage repeated heuristic

- We can adapt the MPC approach by solving two-stage programm instead of deterministic one.
- The procedure goes as follows:
  - at time t_0 in stage x_0, draw N scenarios
  - approximate the problem on $[t_0, T]$ by a two-stage programm with u_{t_0} as first stage variable, and $(u_{t_0+1}, \dots, u_{T-1})$ as recourse
  - apply u_{t_0} and get x_{t_0+1}
  - set $x_0 = x_{t_0+1}$ and $t_0 = t_0 + 1$

Linear Decision Rules

- Another way of getting heuristics consists in looking for solution $\boldsymbol{u}_t = \Phi_t(\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_{t+1})$ where Φ is in a specific class of function.
- Classically we can look for Φ_t in the class of affine functions.
- In which case, a multistage linear stochastic programm turns into a large one-stage stochastic linear programm, which can be approximated by SAA to get a reasonable LP.
- Don't forget to evaluate the obtained heuristic by Monte Carlo on new scenarios.

Contents

Stochastic Controlled Dynamic System

A discrete time controlled stochastic dynamic system is defined by its *dynamic*

$$\boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1})$$

and initial state

$$\boldsymbol{x}_0 = \boldsymbol{\xi}_0$$

The variables

- $\boldsymbol{x}_t$ is the *state* of the system,
- $\boldsymbol{u}_t$ is the *control* applied to the system at time t,
- $\boldsymbol{\xi}_t$ is an exogeneous noise.

Usually, $\boldsymbol{x}_t \in \mathbb{X}_t$ and $\boldsymbol{u}_t$ beglongs to a set depending upon the state: $\boldsymbol{u}_t \in U_t(\boldsymbol{x}_t)$.

Examples

- Stock of water in a dam:
  - $\boldsymbol{x}_t$ is the amount of water in the dam at time t,
  - $\boldsymbol{u}_t$ is the amount of water turbined at time t,
  - $\boldsymbol{\xi}_{t+1}$ is the inflow of water in $[t, t+1[$.
- Boat in the ocean:
  - $\boldsymbol{x}_t$ is the position of the boat at time t,
  - $\boldsymbol{u}_t$ is the direction and speed chosen for $[t, t+1[$,
  - $\boldsymbol{\xi}_{t+1}$ is the wind and current for $[t, t+1[$.
- Subway network:
  - $\boldsymbol{x}_t$ is the position and speed of each train at time t,
  - $\boldsymbol{u}_t$ is the acceleration chosen at time t,
  - $\boldsymbol{\xi}_{t+1}$ is the delay due to passengers and incident on the network for $[t, t+1[$.

More considerations about the state

- Physical state: the physical value of the controlled system. e.g. amount of water in your dam, position of your boat...
- Information state: physical state and information you have over noises. e.g.: amount of water and weather forecast...
- Knowledge state: your current belief over the actual information state (in case of noisy observations). Represented as a distribution law over information states.

The state in the Dynamic Programming sense is the information required to define an optimal solution.

Optimization Problem

We want to solve the following optimization problem

$$\begin{aligned} \min_{\boldsymbol{u}^{\Phi}} \quad & \mathbb{E}\Big[\sum_{t=0}^{T-1} L_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1}) + K(\boldsymbol{x}_T)\Big] \\ s.t. \quad & \boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1}), \qquad \boldsymbol{x}_0 = \boldsymbol{\xi}_0 \\ & \boldsymbol{u}_t \in \mathcal{U}_t(\boldsymbol{x}_t, \boldsymbol{\xi}_{t+1}) \\ & \sigma(\boldsymbol{u}_t) \subset \sigma(\boldsymbol{\xi}_0, \cdots, \boldsymbol{\xi}_{t+1}) \boldsymbol{u}_t = \Phi(\boldsymbol{\xi}_0, \cdots, \boldsymbol{\xi}_{t+1}) \end{aligned}$$

1. We want to minimize the expectation of the sum of costs.
2. The system follows a dynamic given by the function f_t.
3. There are constraints on the controls.
4. The controls are functions of the past noises (= non-anticipativity).

Optimization Problem with independence of noises

If noises at time independent, the optimization problem is equivalent to

$$\begin{aligned} \min_{\pi} \quad & \mathbb{E}\Big[\sum_{t=0}^{T-1} L_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1}) + K(\boldsymbol{x}_T)\Big] \\ s.t. \quad & \boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1}), \qquad \boldsymbol{x}_0 = \boldsymbol{\xi}_0 \\ & \boldsymbol{u}_t \in \mathcal{U}_t(\boldsymbol{x}_t, \boldsymbol{\xi}_{t+1}) \\ & \boldsymbol{u}_t = \pi_t(\boldsymbol{x}_t, \boldsymbol{\xi}_{t+1}) \end{aligned}$$

Keeping only the state

For notational ease, we want to formulate Problem (1) only with states.
Let $\mathcal{X}_t(x_t, \xi_{t+1})$ be the reachable states, i.e.,

$$\mathcal{X}_t(x_t, \xi_{t+1}) := \Big\{ x_{t+1} \in \mathbb{X}_{t+1} \quad | \quad \exists u_t \in \mathcal{U}_t(x_t, \xi_{t+1}), \quad x_{t+1} = f_t(x_t, u_t, \xi_{t+1}) \Big\}.$$

And $c_t(x_t, x_{t+1}, \xi_{t+1})$ the transition cost from x_t to x_{t+1}, i.e.,

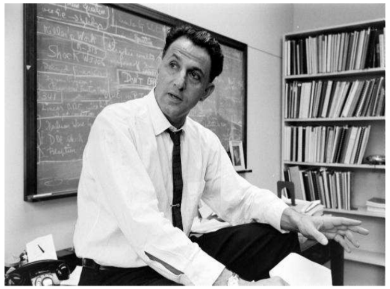
$$c_t(x_t, x_{t+1}, \xi_{t+1}) := \min_{u_t \in \mathcal{U}_t(x_t, \xi_{t+1})} \Big\{ L_t(x_t, u_t, \xi_{t+1}) \quad | \quad x_{t+1} = f_t(x_t, u_t, \xi_{t+1}) \Big\}.$$

Then, under independance of noises, the optimization problem reads

$$\begin{aligned} \min_{\psi} \quad & \mathbb{E}\Big[\sum_{t=0}^{T-1} c_t(x_t, x_{t+1}, \boldsymbol{\xi}_{t+1}) + K(x_T)\Big] \\ s.t. \quad & \boldsymbol{x}_{t+1} \in \mathcal{X}_t(\boldsymbol{x}_t, \boldsymbol{\xi}_{t+1}), \qquad x_0 = \xi_0 \\ & \boldsymbol{x}_{t+1} = \psi_t(\boldsymbol{x}_t, \boldsymbol{\xi}_{t+1}) \end{aligned}$$

Contents

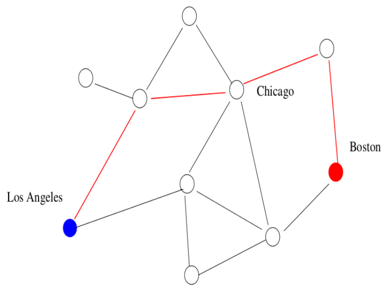
Bellman's Principle of Optimality



Richard Ernest Bellman (August 26, 1920 – March 19, 1984)

An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision (Richard Bellman)

The shortest path on a graph illustrates Bellman's Principle of Optimality



For an auto travel analogy, suppose that the fastest route from Los Angeles to Boston passes through **Chicago**. *The principle of optimality translates to obvious fact that the Chicago to Boston portion of the route is also the fastest route for a trip that starts from Chicago and ends in Boston. (Dimitri P. Bertsekas)*

Idea behind dynamic programming

If noises are time independent, then

1. The cost to go at time t depends only upon the current state.
2. We can compute recursively the cost to go for each position, starting from the terminal state and computing optimal trajectories backward.

Optimal cost-to-go of being in state x at time t is:
At time t, V_{t+1} gives the cost of the future. Dynamic

Programming is a time decomposition method.

Dynamic Programming Principle

Assume that the noises $\boldsymbol{\xi}_t$ are time-independent and exogeneous.
The Bellman's equation writes

$$\begin{cases} V_T(x) &= K(x) \\ \hat{V}_t(x,\xi) &= \min\limits_{y \in \mathcal{X}_t(x,\xi)} c_t(x,y,\xi_{t+1}) + V_{t+1}(y) \\ V_t(x) &= \mathbb{E}\Big[\hat{V}_t(x,\boldsymbol{\xi}_{t+1})\Big] \end{cases}$$

An optimal state trajectory is obtained by $\boldsymbol{x}_{t+1} = \psi_t^V(\boldsymbol{x}_t)$, with

$$\psi_t^V(x,\xi) \in \mathop{\arg\min}_{y \in \mathcal{X}_t(x,\xi)} \underbrace{c_t(x,y,\xi)}_{\text{current cost}} + \underbrace{V_{t+1}(y)}_{\text{future costs}},$$

Interpretation of Bellman Value Function

The Bellman's value function $V_{t_0}(x)$ can be interpreted as the value of the problem starting at time t_0 from the state x.
More precisely we have

$$\begin{aligned} V_{t_0}(x) = \min \quad & \mathbb{E}\Big[\sum_{t=t_0}^{T-1} L_t(\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{\xi}_{t+1}) + K(\boldsymbol{x}_T)\Big] \\ s.t. \quad & \boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{\xi}_{t+1}), \qquad \boldsymbol{x}_{t_0} = x \\ & \boldsymbol{u}_t \in \mathcal{U}_t(\boldsymbol{x}_t,\boldsymbol{\xi}_{t+1}) \\ & \sigma(\boldsymbol{u}_t) \subset \sigma(\boldsymbol{\xi}_0,\cdots,\boldsymbol{\xi}_{t+1}) \end{aligned}$$

or

$$\begin{aligned} \min_{\psi} \quad & \mathbb{E}\Big[\sum_{t=t_0}^{T-1} c_t(x_t,x_{t+1},\boldsymbol{\xi}_{t+1}) + K(x_T)\Big] \\ s.t. \quad & \boldsymbol{x}_{t+1} \in \mathcal{X}_t(\boldsymbol{x}_t,\boldsymbol{\xi}_{t+1}), \qquad x_{t_0} = x \\ & \boldsymbol{x}_{t+1} = \psi_t(\boldsymbol{x}_t) \end{aligned}$$

Contents

Optimization Problem

Recall that we want to solve the following optimization problem

$$\begin{aligned} \min_{\psi} \quad & \mathbb{E}\Big[\sum_{t=0}^{T-1} c_t(x_t, x_{t+1}, \boldsymbol{\xi}_{t+1}) + K(x_T)\Big] \\ s.t. \quad & \boldsymbol{x}_{t+1} \in \mathcal{X}_t(\boldsymbol{x}_t, \boldsymbol{\xi}_{t+1}), \qquad x_0 = \boldsymbol{\xi}_0 \\ & \boldsymbol{x}_{t+1} = \psi_t(\boldsymbol{x}_t) \end{aligned}$$

With Bellman's equation reading

$$\begin{cases} V_T(x) & = K(x) \\ \hat{V}_t(x,\xi) & = \min_{y \in \mathcal{X}_t(x,\xi)} \quad c_t(x,y,\xi) + V_{t+1}(y) \\ V_t(x) & = \mathbb{E}\big[\hat{V}_t(x, \boldsymbol{\xi}_{t+1})\big] \end{cases}$$

Bellman operator

For any time t, and any function R mapping the set of states and noises $\mathbb{X} \times \Xi$ into $\mathbb{R}$, we define

$$\begin{cases} \hat{\mathcal{B}}_t(R)(x,\xi) & := \min_{y \in \mathcal{X}_t(x,\xi)} \quad c_t(x,y,\xi) + R(y) \\ \mathcal{B}_t(R)(x) & := \mathbb{E}\Big(\hat{\mathcal{B}}_t(R)(x, \boldsymbol{\xi}_{t+1})\Big) \end{cases}$$

Thus the Bellman equation simply reads

$$\begin{cases} V_T & = & K \\ V_t & = & \mathcal{B}_t(V_{t+1}) \end{cases}$$

Further, any estimation R of the value functions yields an admissible trajectory given by

$$\psi_t^R(x,\xi) \in \arg\min_{y \in \mathcal{X}(x,\xi)} c_t(x,y,\xi) + R_{t+1}(y)$$

optimal if $R_t = V_t$.

Properties of the Bellman operator

Assume that $\boldsymbol{\xi}_t$ are finitely supported

- Monotonicity:
  $$R \leq \overline{R} \quad \Rightarrow \mathcal{B}_t(R) \leq \mathcal{B}_t(\overline{R})$$
- Convexity: if c_t is jointly convex in (x,y) for all ξ, R is convex, $\mathrm{gr}(\mathcal{X}_t)$ is convex then
  $$x \mapsto \mathcal{B}_t(R)(x) \quad \text{is convex}$$
- Polyhedrality: for any polyhedral function R, if c_t is also polyhedral for all ξ, and $\mathrm{gr}(\mathcal{X}_t)$ is polyhedral, then
  $$x \mapsto \mathcal{B}_t(R)(x) \quad \text{is polyhedral}$$

Computing upper bounds

In the convex case we can compute exact upper-bound on the value of the stochastic optimization problem.

- For all $t \leq T$, select points $\{x_t^n\}_{n \leq N}$ in $\mathbb{X}_t$.
- For $t = T$, define $v_T^n = K(x_t^n)$.
- Iteratively backward for $t = T..1$:
  - $\bar{V}_t(x) := \min_{\alpha \in \Delta_n} \left\{ \sum_{n=1}^N \alpha^n v_t^n \;\middle|\; \sum_{n=1}^N \alpha^n x_t^n = x \right\}$
  - where $\Delta_n = \{\alpha \in \mathbb{R}^n \mid \sum_n \alpha_n = 1, \ \alpha_n \geq 0\}$.
  - Compute $v_{t-1}^n = \mathcal{B}_{t-1}(\bar{V}_t)(x_{t-1}^n)$
- For all t, $\bar{V}_t \geq V_t$, and in particular $\mathcal{B}_0\big(\bar{V}_1\big)(x_0)$ is an upper bound on the value of our problem.

Contents

Dynamic Programming Algorithm - Discrete Case

Data: Problem parameters
Result: optimal trajectory and value;
$V_T \equiv K$; $V_t \equiv 0$
for $t : T-1 \to 0$ **do**
  for $x \in \mathbb{X}_t$ **do**
    for $\xi \in \Xi_t$ **do**
      $\hat{V}_t(x, \xi) = \infty$;
      for $y \in \mathcal{X}_t(x, \xi)$ **do**
        $v_y = c_t(x, y, \xi) + V_{t+1}(y)$;
        if $v_y < \hat{V}_t(x, \xi)$ **then**
          $\hat{V}_t(x, \xi) = v_y$;
          $\psi_t(x, \xi) = y$;
    $V_t(x) = V_t(x) + \mathbb{P}(\xi)\hat{V}_t(x, \xi)$

Algorithm 1: Classical stochastic dynamic programming algorithm

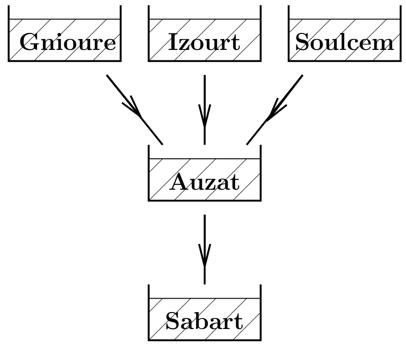
3 curses of dimensionality

Complexity $= O(T \times |\mathbb{X}_t| \times |\mathcal{X}_t| \times |\Xi_t|)$
Linear in the number of time steps, but we have 3 curses of dimensionality :

1. State. Complexity is exponential in the dimension of $\mathbb{X}_t$
   e.g. 3 independent states each taking 10 values leads to a loop over 1000 points.
2. Decision. Complexity is exponential in the dimension of $\mathcal{X}_t$.
   $\rightsquigarrow$ due to exhaustive minimization of inner problem. Can be accelerated using faster method (e.g. MILP solver).
3. Expectation. Complexity is exponential in the dimension of Ξ_t.
   $\rightsquigarrow$ due to expectation computation. Can be accelerated through Monte-Carlo approximation (still at least 1000 points)

In practice DP is not used for state of dimension more than 5.

Illustrating dynamic programming with the damsvalley example



Illustrating the curse of dimensionality

We are in dimension 5 (not so high in the world of big data!) with 52 timesteps (common in energy management) plus 5 controls and 5 independent noises.

1. We discretize each state's dimension in 100 values: $|\mathbb{X}_t| = 100^5 = 10^{10}$
2. We discretize each control's dimension in 100 values: $|U_t| = 100^5 = 10^{10}$
3. We use optimal quantization to discretize the noises' space in 10 values: $|\Xi_t| = 10$

Number of flops: $\mathcal{O}(52 \times 10^{10} \times 10^{10} \times 10) \approx \mathcal{O}(10^{23})$.
In the TOP500, the best computer computes 10^{17} flops/s.
Even with the most powerful computer, it takes at least 12 days to solve this problem.

Contents

Computing a decision online

Algorithm: Offline value functions precomputation + Online open loop reoptimization

Offline: We produce value functions with Bellman equation:

$$V_t(x) = \mathbb{E}\Big[\min_{y \in \mathcal{X}_t(x,\xi_{t+1})} c_t(x, y, \xi_{t+1}) + V_{t+1}(y)\Big]$$

Online: At time t, knowing x_t and ξ_{t+1} we plug the computed value function V_{t+1} as future cost

$$x_{t+1} \in \operatorname*{arg\,min}_{y \in \mathcal{X}_t(x_t,\xi_{t+1})} c_t(x_t, y, \xi_{t+1}) + V_{t+1}(y)$$

This can be extended to approximate value function $\tilde{V}_t$ computed in any way.

Dynamic Programming : Discretization-Interpolation

- When the state space is continuous, the DP equation holds :

$$V_t(x) = \mathbb{E}\Big[\min_{y \in \mathcal{X}_t(x,\xi_{t+1})} c_t(x,y,\xi_{t+1}) + V_{t+1}(y)\Big].$$

- But computation is impractical in a continuous space. Simplest solution : discretization and interpolation.
- We choose a finite set $\mathbb{X}_t^D \subset \mathbb{X}_t$ where we will compute (an approximation of) the Bellman value V_t.
- We approximate the Bellman value at time t by interpolating these value.

Dynamic Programming : Discretization-Interpolation

Data: Problem parameters, discretization, one-stage solver, interpolation operator;
Result: approximation of optimal value;
$\tilde{V}_T \equiv K$;
for $t : T-1 \to 0$ **do**
  for $x \in \mathbb{X}_t^D$ **do**
    $\tilde{V}_t(x) := \mathbb{E}\Big[\min_{y \in \mathcal{X}_t(x,\xi_{t+1})} c_t(x,y,\xi_{t+1}) + \tilde{V}_{t+1}(y)\Big]$;
  Define $\tilde{V}_t$ by interpolating $\{\tilde{V}_t(x) \mid x \in \mathbb{X}_t^D\}$;

Algorithm 2: Dynamic Programming Algorithm (Continuous)

Independence of noises

- The Dynamic Programming equation requires only the time-independence of noises.
- This can be relaxed if we consider an extended state.
- Consider a dynamic system driven by an equation

$$\boldsymbol{y}_{t+1} = f_t(\boldsymbol{y}_t, \boldsymbol{u}_t, \boldsymbol{\varepsilon}_{t+1})$$

where the random noise $\boldsymbol{\varepsilon}_t$ is an AR-1 process :

$$\boldsymbol{\varepsilon}_t = \alpha_t \boldsymbol{\varepsilon}_{t-1} + \beta_t + \boldsymbol{\xi}_t,$$

$\{\boldsymbol{\xi}_t\}_{t\in\mathbb{Z}}$ being independent.
- Then $\boldsymbol{y}_t$ is called the physical state of the system and DP can be used with the information state $\boldsymbol{x}_t = (\boldsymbol{y}_t, \boldsymbol{\varepsilon}_t)$.
- Generically speaking, if the noise $\boldsymbol{\xi}_t$ is exogeneous (not affected by decisions $\boldsymbol{u}_t$), then we can always apply Dynamic Programming with the state $(\boldsymbol{x}_t, \boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_t)$.

State augmentation limits

Because of the curse of dimensionality it might be impossible to take into account correlation by augmenting the state variable.

Practitioners often ignore noise dependence for the value functions computation but use dependence information during online reoptimization.

Conclusion

- Multistage stochastic programming fails to handle large number of timesteps.
- Dynamic Programming overcomes this difficulty while compressing information inside a state $\boldsymbol{x}$.
- Dynamic Programming computes backward a set of value functions $\{V_t\}$, corresponding to the optimal cost of being at a given position at time t.
- Numerically, DP is limited by the curse of dimensionality and its performance are deeply related to the discretization of the look-up table used.
- Other methods exist to compute the value functions without look-up table (Approximate Dynamic Programming, SDDP).

Independence of noises: AR-1 case

- Consider a dynamic system driven by an equation $\boldsymbol{y}_{t+1} = f_t(\boldsymbol{y}_t, \boldsymbol{u}_t, \boldsymbol{\varepsilon}_{t+1})$ where the random noise $\boldsymbol{\varepsilon}_t$ is an AR-1 process : $\boldsymbol{\varepsilon}_t = \alpha_t \boldsymbol{\varepsilon}_{t-1} + \beta_t + \boldsymbol{\xi}_{t+1}$, $\{\boldsymbol{\xi}_t\}_{t\in\mathbb{Z}}$ being independent.
- Define the information state $\boldsymbol{x}_t = (\boldsymbol{y}_t, \boldsymbol{\varepsilon}_t)$.
- Then we have

$$\boldsymbol{x}_{t+1} = \begin{pmatrix} f_t(\boldsymbol{y}_t, \boldsymbol{u}_t, \alpha_t\boldsymbol{\varepsilon}_t + \beta_t + \boldsymbol{\xi}_{t+1}) \\ \alpha_t\boldsymbol{\varepsilon}_t + \beta_t + \boldsymbol{\xi}_{t+1} \end{pmatrix} = \tilde{f}_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_{t+1})$$

- And we have the following DP equation

$$V_t\binom{y}{\varepsilon} = \min_{u\in U_t(x)} \mathbb{E}\Big[L_t(y, u, \underbrace{\alpha_t\varepsilon + \beta_t + \boldsymbol{\xi}_{t+1}}_{"\boldsymbol{\varepsilon}_{t+1}"}) + V_{t+1}\circ\underbrace{\tilde{f}_t(x, u, \boldsymbol{\xi}_{t+1})}_{"\boldsymbol{x}_{t+1}"}\Big]$$

DP on a Markov Chain

- Sometimes it is easier to represent a problem as a controlled Markov Chain
- Dynamic Programming equation can be computed directly, without expliciting the control.
- Let's work out an example...

Controlled Markov Chain

- A controlled Markov Chain is controlled stochastic dynamic system with independent noise $(\boldsymbol{w}_t)_{t\in\mathbb{Z}}$, where the dynamic and the noise are left unexplicited.
- What is given is the *transition probability*

$$\pi_t^u(x, y) := \mathbb{P}\Big(\boldsymbol{x}_{t+1} = y \mid \boldsymbol{x}_t = x, \boldsymbol{u}_t = u\Big).$$

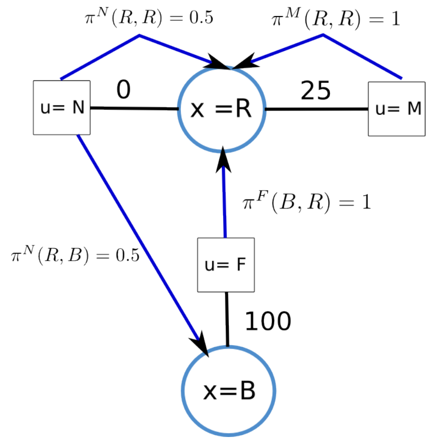
- In this case the cost are given as a function of the current stage, the next stage and the control.
- The Dynamic Programming Equation then reads (assume finite state)

$$V_t(x) = \min_u \sum_{y\in\mathbb{X}_{t+1}} \pi_t^u(x, y)\Big[L_t^u(x, y) + V_{t+1}(y)\Big].$$

Example

Consider a machine that has two states : running (R) and broken (B). If it is broken we need to fix it (F) for a cost of 100. If it is running there are two choices: maintaining it (M), or not maintaining (N). If we maintain, the cost is 25 and the machine stay running with probability $\pi^M(R,R) = 1$; if we do not maintain there is a probability of $\pi^N(R,B) = 0.5$ of breaking it (or keep it running). We need to have it running for 3 periods.

Controlled Markov Chain



	V_0	V_1
R	$\min\left\{25+50, 0+(50+125)/2\right\}75$	$\min\left\{25+25, 0+(25+100)/\right.$
B	$100+50150$	$100+25125$

Stochastic Optimization
Decomposition Methods for Two-stage problems

V. Leclère

January 5th 2022

École des Ponts
ParisTech

Presentation Outline

Two-stage Problem

The extensive formulation of

$$\begin{aligned} \min_{u_0, \boldsymbol{u}_1} \quad & \mathbb{E}\Big[L(u_0, \boldsymbol{\xi}, \boldsymbol{u}_1)\Big] \\ s.t. \quad & g(u_0, \boldsymbol{\xi}, \boldsymbol{u}_1) \leq 0, \qquad \mathbb{P} - a.s \\ & \sigma(\boldsymbol{u}_1) \subset \sigma(\boldsymbol{\xi}) \end{aligned}$$

is

$$\begin{aligned} \min_{u_0, \{u_1^s\}_{s \in [\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s L(u_0, \xi^s, u_1^s) \\ s.t \quad & g(u_0, \xi^s, u_1^s) \leq 0, \qquad \forall s \in [\![1, S]\!]. \end{aligned}$$

It is a deterministic problem that can be solved with standard tools or specific methods.

Splitting variables

The extended Formulation (in a compact formulation)

$$\begin{aligned} \min_{u_0, \{u_1^s\}_{s \in [\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s L(u_0^s, \xi^s, u_1^s) \\ s.t \quad & g(u_0, \xi^s, u_1^s) \leq 0, \qquad \forall s \in [\![1, S]\!]. \end{aligned}$$

Can be written in a splitted formulation

$$\begin{aligned} \min_{\bar{u}_0 \{u_0^s, u_1^s\}_{s \in [\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s L(u_0^s, \xi^s, u_1^s) \\ s.t \quad & g(u_0^s, \xi^s, u_1^s) \leq 0, \qquad \forall s \in [\![1, S]\!] \\ & u_0^s = u_0^{s'} \qquad \forall s, s' \end{aligned}$$

Splitting variables

The extended Formulation (in a compact formulation)

$$\min_{u_0,\{u_1^s\}_{s\in[\![1,S]\!]}} \sum_{s=1}^{S} \pi^s L(u_0^s,\xi^s,u_1^s)$$
$$s.t \quad g(u_0,\xi^s,u_1^s) \leq 0, \qquad \forall s \in [\![1,S]\!].$$

Can be written in a splitted formulation

$$\min_{\bar{u}_0\{u_0^s,u_1^s\}_{s\in[\![1,S]\!]}} \sum_{s=1}^{S} \pi^s L(u_0^s,\xi^s,u_1^s)$$
$$s.t \quad g(u_0^s,\xi^s,u_1^s) \leq 0, \qquad \forall s \in [\![1,S]\!]$$
$$u_0^s = \sum_{s'} \pi^{s'} u_0^{s'} \qquad \forall s$$

Dualizing non-anticipativity constraint I

$$\min_{\{u_0^s,u_1^s\}_{s\in[\![1,S]\!]}} \sum_{s=1}^{S} \pi^s L(u_0^s,\xi^s,u_1^s)$$
$$s.t \quad g(u_0^s,\xi^s,u_1^s) \leq 0, \qquad \forall s \in [\![1,S]\!]$$
$$u_0^s = \sum_{s'} \pi^{s'} u_0^{s'} \qquad \forall s$$

is equivalent to

$$\min_{\{u_0^s,u_1^s\}_{s\in[\![1,S]\!]}} \max_{\{\lambda^s\}_{s\in[\![1,S]\!]}} \sum_{s=1}^{S} \pi^s L(u_0^s,\xi^s,u_1^s) + \pi^s\lambda^s\Big(u_0^s - \sum_{s'} \pi^{s'} u_0^{s'}\Big)$$

$$s.t \quad g(u_0^s,\xi^s,u_1^s) \leq 0, \qquad \forall s \in [\![1,S]\!]$$

Dualizing non-anticipativity constraint I

$$\min_{\{u_0^s,u_1^s\}_{s\in[\![1,S]\!]}} \sum_{s=1}^{S} \pi^s L(u_0^s,\xi^s,u_1^s)$$
$$s.t \quad g(u_0^s,\xi^s,u_1^s) \leq 0, \qquad \forall s \in [\![1,S]\!]$$
$$u_0^s = \sum_{s'} \pi^{s'} u_0^{s'} \qquad \forall s$$

is equivalent to

$$\min_{\{u_0^s,u_1^s\}_{s\in[\![1,S]\!]}} \max_{\{\lambda^s\}_{s\in[\![1,S]\!]}} \sum_{s=1}^{S} \pi^s L(u_0^s,\xi^s,u_1^s)$$
$$+ \sum_{s=1}^{S} \pi^s\lambda^s u_0^s - \sum_{s'} \mathbb{E}\big[\boldsymbol{\lambda}\big]\pi^{s'} u_0^{s'}$$
$$s.t \quad g(u_0^s,\xi^s,u_1^s) \leq 0, \qquad \forall s \in [\![1,S]\!]$$

Dualizing non-anticipativity constraint I

$$\min_{\{u_0^s,u_1^s\}_{s\in[\![1,S]\!]}} \sum_{s=1}^{S} \pi^s L(u_0^s,\xi^s,u_1^s)$$
$$s.t \quad g(u_0^s,\xi^s,u_1^s) \leq 0, \qquad \forall s \in [\![1,S]\!]$$
$$u_0^s = \sum_{s'} \pi^{s'} u_0^{s'} \qquad \forall s$$

is equivalent to

$$\min_{\{u_0^s,u_1^s\}_{s\in[\![1,S]\!]}} \max_{\{\lambda^s\}_{s\in[\![1,S]\!]}} \sum_{s=1}^{S} \pi^s L(u_0^s,\xi^s,u_1^s)$$
$$+ \sum_{s=1}^{S} \pi^s\Big(\lambda^s - \mathbb{E}\big[\boldsymbol{\lambda}\big]\Big) u_0^s$$
$$s.t \quad g(u_0^s,\xi^s,u_1^s) \leq 0, \qquad \forall s \in [\![1,S]\!]$$

Dualizing non-anticipativity constraint II

Thus, the dual problem reads

$$\max_{\boldsymbol{\lambda}:\mathbb{E}[\boldsymbol{\lambda}]=0} \min_{\{u_0^s,u_1^s\}_{s\in[\![1,S]\!]}} \sum_{s=1}^{S} \pi^s \Big(L(u_0^s,\xi^s,u_1^s) + \big(\lambda^s - \mathbb{E}[\boldsymbol{\lambda}]\big) u_0^s \Big)$$
$$s.t \quad g(u_0^s,\xi^s,u_1^s) \le 0, \qquad \forall s \in [\![1,S]\!]$$

The inner minimization problem, for $\boldsymbol{\lambda}$ given, can decompose scenario by scenario, by solving S deterministic problem

$$\min_{\{u_0^s,u_1^s\}} \quad L(u_0^s,\xi^s,u_1^s) + \lambda^s u_0^s$$
$$s.t \quad g(u_0^s,\xi^s,u_1^s) \le 0$$

Price of information

- By weak duality, any $\boldsymbol{\lambda}$ such that $\mathbb{E}[\boldsymbol{\lambda}] = 0$ will give a lower bound on the 2-stage problem, computed as

$$\sum_{s=1}^{S} \pi^s \min_{u_0^s,u_1^s} \quad \Big(L(u_0^s,\xi^s,u_1^s) + \lambda^s u_0^s \Big)$$
$$s.t \quad g(u_0^s,\xi^s,u_1^s) \le 0$$

- $\boldsymbol{\lambda} = 0$ lead to the anticipative lower-bound
- If problem is convex, and under some qualification assumptions, there exists an optimal $\boldsymbol{\lambda}^*$, called the price of information, such that the lower bound is tight.

Progressive Hedging Algorithm

The progressive hedging algorithm build on this decomposition in the following way.

1. Set a price of information $\{\lambda^s\}_{s\in[\![1,S]\!]}$ such that $\mathbb{E}[\boldsymbol{\lambda}] = 0$
2. For each scenario solve

$$\min_{u_0^s,u_1^s} \quad L(u_0^s,\xi^s,u_1^s) + \lambda^s u_0^s + \rho\|u_0^s - \bar{u}_0\|^2$$
$$s.t \quad g(u_0^s,\xi^s,u_1^s) \le 0$$

3. Compute the mean first control $\bar{u}_0 := \sum_{s=1}^{S} \pi^s u_0^s$
4. Update the price of information with

$$\lambda^s := \lambda^s + \rho(u_0^s - \bar{u}_0)$$

5. Go back to 2.

Convergence of Progressive Hedging

Theorem

Assume that L and g are convex lsc in (u_0,u_1) for all ξ, and that, for all $s \in S$, there exists (u_0^s,u_1^s) such that $L(u_0^s,\xi^s,u_1^s) < +\infty$ and $g(u_0^s,\xi^s,u_1^s) < 0$.
Then, the progressive hedging algorithm converges toward an optimal primal solution, and the price of information converges toward an optimal price of information.

Moreover we can show that

$$\varepsilon_k = \sqrt{\|(u_0^k,u_1^k) - (u_0^\sharp,u_1^\sharp)\|_2^2 + \frac{1}{\rho^2}\|\lambda - \lambda^\sharp\|_2^2},$$

is a decreasing sequence.

Bounds in Progressive Hedging

- At any iteration of the PH algorithm, we have a collection of primal solution $\{(u_0^s, u_1^s)\}_{s\in S}$, and a price of information $\{\lambda^s\}_{s\in S}$.
- We have a lower bound on the value of the stochastic programm given by

$$LB^{PH} = \sum_{s\in S} \pi^s \big[L(u_0^s, \xi^s, u_1^s) + \lambda^s u_0^s \big],$$

- and an upper bound given by

$$UB^{PH} = \sum_{s\in S} \pi^s L(\bar{u}_0, \xi^s, u_1^s(u_0)).$$

Presentation Outline

Linear 2-stage stochastic program

Consider the following problem

$$\begin{aligned} \min \quad & \mathbb{E}\Big[c^\top u_0 + \boldsymbol{q}^\top \boldsymbol{u}_1 \Big] \\ s.t. \quad & Au_0 = b, \quad u_0 \geq 0 \\ & \boldsymbol{T} u_0 + \boldsymbol{W} \boldsymbol{u}_1 = \boldsymbol{h}, \quad \boldsymbol{u}_1 \geq 0, \qquad \mathbb{P}-a.s. \\ & u_0 \in \mathbb{R}^n, \quad \sigma(\boldsymbol{u}_1) \subset \sigma(\underbrace{\boldsymbol{q}, \boldsymbol{T}, \boldsymbol{W}, \boldsymbol{h}}_{\boldsymbol{\xi}}) \end{aligned}$$

Which we rewrite

$$\begin{aligned} \min_{u_0 \geq 0} \quad & c^\top u_0 + \mathbb{E}\Big[Q(u_0, \boldsymbol{\xi}) \Big] \\ s.t. \quad & Au_0 = b \end{aligned}$$

with

$$\begin{aligned} Q(u_0, \xi) := \min_{u_1 \geq 0} \quad & q_\xi^\top u_1 \\ s.t. \quad & W_\xi u_1 = h_\xi - T_\xi u_0 \end{aligned}$$

Linear 2-stage stochastic program : Extensive Formulation

The associated extensive formulation read

$$\begin{aligned} \min \quad & c^\top u_0 + \sum_{s=1}^S \pi^s \, q^s \cdot u_1^s \\ s.t. \quad & Au_0 = b, \quad u_0 \geq 0 \\ & T^s u_0 + W^s u_1^s = h^s, \quad u_1^s \geq 0, \forall s \end{aligned}$$

Which we rewrite

$$\begin{aligned} \min_{u_0} \quad & c^\top u_0 + \sum_{s=1}^S \pi^s Q^s(u_0) \\ s.t. \quad & Au_0 = b, \quad u_0 \geq 0 \end{aligned}$$

with

$$\begin{aligned} Q^s(u_0) := \min_{u_1 \geq 0} \quad & q^s \cdot u_1 \\ s.t. \quad & W^s u_1 = h^s - T^s u_0 \end{aligned}$$

Relatively complete recourse

We assume here relatively complete recourse. Without this assumption we would need feasability cuts.
Here, relatively complete recourse means that, for $u_0 \geq 0$:

$$Au_0 = b \quad \Longrightarrow \quad Q_s(u_0) < +\infty, \quad \forall s \in [\![1, S]\!]$$

Decomposition of linear 2-stage stochastic program

We rewrite the extended formulation as

$$\begin{aligned} \min_{u_0,(\theta^s)_{s\in S}} \quad & c^\top u_0 + \sum_s \pi^s \theta^s \\ s.t. \quad & Au_0 = b, \quad u_0 \geq 0 \\ & \theta^s \geq Q^s(u_0)\theta^s \geq \alpha_k^s \cdot u_0 + \beta_k^s \qquad \forall k, \forall s \end{aligned}$$

Note that $Q^s(u_0)$ is a polyhedral function of u_0, hence $\theta^s \geq Q^s(u_0)$ can be rewritten $\theta^s \geq \alpha_k^s \cdot u_0 + \beta_k^s, \forall k$.
The decomposition approach consists in constructing iteratively cut coefficients α_k^s and β_k^s.

Obtaining (optimality) cuts I

Recall that

$$\begin{aligned} Q^s(u_0) := \min_{u_1^s \in \mathbb{R}^n} \quad & q^s \cdot u_1^s \\ s.t. \quad & W^s u_1^s = h^s - T^s u_0, \quad u_1^s \geq 0 \end{aligned}$$

can also be written (through strong duality by relatively complete recourse assumption)

$$\begin{aligned} (D_{u_0}) \quad Q^s(u_0) = \max_{\lambda^s \in \mathbb{R}^m} \quad & \lambda^s \cdot (h^s - T^s u_0) \\ s.t. \quad & (W^s)^\top \lambda^s \leq q^s \end{aligned}$$

Obtaining (optimality) cuts II

$$\begin{aligned} (D_{u_0}) \quad Q^s(u_0) = \max_{\lambda^s \in \mathbb{R}^m} \quad & \lambda^s \cdot (h^s - T^s u_0) \\ s.t. \quad & (W^s)^\top \lambda^s \leq q^s \end{aligned}$$

admits for optimal solution $\lambda_{u_0}^s$.
Consider another control u_0', we have

$$\begin{aligned} (D_{u_0'}) \quad Q^s(u_0') = \max_{\lambda^s \in \mathbb{R}^m} \quad & \lambda^s \cdot (h^s - T^s u_0') \\ s.t. \quad & (W^s)^\top \lambda^s \leq q^s \end{aligned}$$

As $\lambda_{u_0}^s$ is admissible for (D_{u_0}) it is also admissible for $(D_{u_0'})$, hence

$$Q^s(u_0') \geq \lambda_{u_0}^s \cdot (h^s - T^s u_0').$$

Obtaining (optimality) cuts III

To sum up we have seen that, for any admissible first stage solution, we can construct an exact cut for Q^s by solving the dual of the second stage problem.

More precisely, let $u_0^k \geq 0$ be such that $Au_0^k = b$. Let λ_k^s be an optimal dual solution. Then, setting

$$\alpha_k^s := -(T^s)^\top \lambda_k^s \quad \text{and} \quad \beta_k^s := (\lambda_k^s)^\top h^s$$

we have

$$\begin{cases} Q^s(u_0') \geq \alpha_k^s \cdot u_0' + \beta_k^s & \forall u_0' \geq 0, Au_0' = b \\ Q^s(u_0^k) = \alpha_k^s \cdot u_0^k + \beta_k^s \end{cases}$$

L-shaped method (multi-cut version)

1. We have a collection of $K \times S$ cuts, such that $Q^s(u_0) \geq \alpha_k^s \cdot u_0 + \beta_k^s$.
2. Solve the master problem, with optimal primal solution u_0^{K+1}.

$$\begin{aligned} \min_{u_0 \geq 0} \quad & c^\top u_0 + \sum_{s=1}^S \pi^s \theta^s \\ s.t. \quad & Au_0 = b \\ & \theta^s \geq \alpha_k^s u_0 + \beta_k^s \qquad \forall k \in [\![1, K]\!], \quad \forall s \in [\![1, S]\!] \end{aligned}$$

3. Solve S slave problems, with optimal dual solution λ_{K+1}^s

$$\begin{aligned} Q^s(u_0^{K+1}) = \min_{u_1^s \in \mathbb{R}^n} \quad & q^s \cdot u_1^s \\ s.t. \quad & W^s u_1^s = h^s - T^s u_0^{K+1}, \quad u_1^s \geq 0 \end{aligned}$$

$$Q^s(u_0^{K+1}) = \max_{\lambda^s \in \mathbb{R}^m} \quad \lambda^s \cdot (h^s - T^s u_0^{K+1})$$

L-shaped method (multi-cut version) : bounds

- At any iteration of the L-shaped method we can easily determine upper and lower bound over our problem.
- Indeed, u_0^K is an admissible firt stage solution, and $Q^s(u_0^K)$ is the value of a slave problem. Thus the value of admissible solution u_0^k is simply given by

$$UB = c^\top u_0^K + \sum_{s=1}^S \pi^s Q^s(u_0^K).$$

- Furthermore, $Q_K^s(u_0) \geq \max_{k \leq K} \alpha_k^s \cdot u_0 + \beta_k^s$, thus the value of the master problem is always a lower bound over the value of the SP problem :

$$LB = c^\top u_0^K + \sum_{s=1}^S \pi^s \theta_K^s.$$

L-shaped method (single-cut version)

1. We have a collection of K cuts, such that $Q(u_0) := \sum_{s \in S} Q^s(u_0) \geq \alpha_k \cdot u_0 + \beta_k$.
2. Solve the master problem, with optimal primal solution u_0^{K+1}.

$$\begin{aligned} \min_{u_0 \geq 0} \quad & c^\top u_0 + \theta \\ s.t. \quad & Au_0 = b \\ & \theta \geq \alpha_k u_0 + \beta_k \qquad \forall k \in [\![1, K]\!] \end{aligned}$$

3. Solve S slave dual problems, with optimal dual solution λ_{K+1}^s

$$\begin{aligned} \max_{\lambda^s \in \mathbb{R}^m} \quad & \lambda^s \cdot (h^s - T^s u_0^{K+1}) \\ s.t. \quad & W^s \cdot \lambda^s \leq q^s \end{aligned}$$

4. construct new cut with

$$\alpha_{K+1} := -\sum_{s}^{S} \pi^s \, (T^s)^\top \lambda^s, \qquad \beta_{K+1} := \sum^{S} \pi^s \, h^s \cdot \lambda^s.$$

Feasibility cuts

- Without the relatively complete recourse assumption we cannot guarantee that $Q(u_0) < +\infty$, however we still have that Q is polyhedral, thus so is $dom(Q)$.
- Without RCR we need to add feasibility cuts in the following way:
  - If, $Q^s(u_0^k) = +\infty$, then we can find an unbounded ray of the dual problem

$$\begin{aligned} \max_{\lambda^s \in \mathbb{R}^m} \quad & \lambda^s \cdot \left(h^s - T^s u_0^k\right) \\ s.t. \quad & W^s \cdot \lambda^s \le q^s \end{aligned}$$

    more precisely a vector $\bar{\lambda}^k$ such that, for all $t \geq 0$ $W^s \cdot t\bar{\lambda}^k \leq q^s$.
  - Then, for u_0 to be admissible, we need that

$$\bar{\lambda}^k \cdot \left(h^s - T^s u_0\right) \le 0$$

    which is a feasibility cut.

Convergence

Theorem

In the linear case, the L-Shaped algorithm terminates in finitely many steps, yielding the optimal solution.

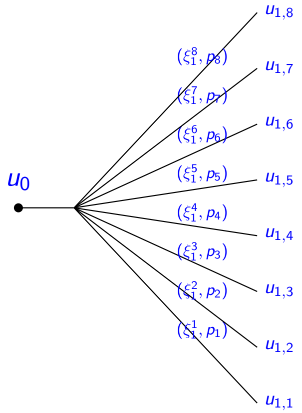
The proof is done by noting that only finitely many cuts can be added, and not being able to add a cut prove that the algorithm has converged.

Comparison of Progressive Hedging and L-shaped

	Progressive Hedging	L-Shaped
problems	convex continuous	linear, 1st stage integer
sol. at it. k	non-admissible splitted solutions	admissible primal solution
Bounds	LB free, UB easy	LB and UB free
Convergence	asymptotic	finite
Complexity	fixed : S deterministic problem	increasing for master problem, fixed for slave problem
Implem.	easy from deterministic solver	built from scratch

Presentation Outline

Where do we come from: two-stage programming



- We take decisions in two stages

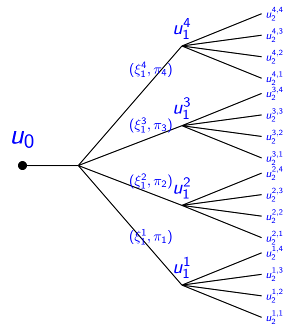
$$u_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow \boldsymbol{u}_1 ,$$

with $\boldsymbol{u}_1$: recourse decision .

- On a tree, it means solving the extensive formulation:

$$\min_{u_0, u_{1,s}} c_0 u_0 + \sum_{s \in \mathbb{S}} p_s \big[\langle c_s \, , u_{1,s} \rangle \big] .$$

Extending two-stage to multistage programming



- We want to minimize $\min_{\boldsymbol{u}} \mathbb{E}\big[c(\boldsymbol{u}, \boldsymbol{\xi})\big]$
- Where we take decisions in T stages

$$\boldsymbol{u}_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow \boldsymbol{u}_1 \rightsquigarrow \cdots \rightsquigarrow \boldsymbol{\xi}_T \rightsquigarrow \boldsymbol{u}_T .$$

- It can be represented on a tree $\mathcal{T}$, where a node n of depth t represent a realisation of $(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_t)$, and to which is attached a probability p_n.
- Then, the extensive formulation reads

$$\min_{\{u_n\}_{n \in \mathcal{T}}} \quad \sum_{n \in \mathcal{T}} p_n c_n(u_n)$$

Compact and splitted extended formulation

- Consider a tree of depth T. A scenario $s = (n_1, \dots, n_T)$ is a sequence of node, where each element is a descendent of the previous one. A scenario $s \in S$ is uniquely defined by its last element, which is a leaf of the tree.
- Let π^s be the probability of the leaf defining scenario s.
- The compact formulation of the multistage problem reads

$$\min_{\{u_n\}_{n \in \mathcal{T}}} \quad \sum_{n \in \mathcal{T}} \pi^n c_n(u_n) = \sum_{s \in S} \pi^s \sum_{n \in S} c_n(u_n)$$

- The splitted extended formulation reads

$$\min_{\{u_{s,t}\}_{s \in S, t \in [\![0,T]\!]}} \quad \sum_{s \in S} \pi^s \sum_{t=0}^{T} c_{s,t}(u_{s,t})$$

$$s.t. \qquad u_{s,t} = u_{s',t} \qquad\qquad \forall t, \forall n \in \mathcal{N}_t, \forall s, s' \ni n$$

where $\mathcal{N}_t$ is the set of nodes of depth t

Introducing the non-anticipativity constraint

We do not know what holds behind the door.

Non-anticipativity

At time t, decisions are taken sequentially, only knowing the past realizations of the perturbations.

Mathematically, this is equivalent to say that at time t, the decision $\boldsymbol{u}_t$ is

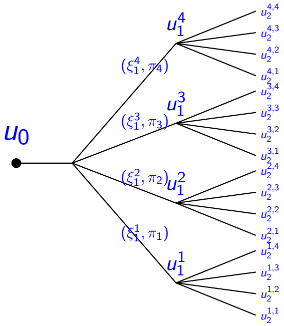
1. a function of past noises

$$\boldsymbol{u}_t = \pi_t(\boldsymbol{\xi}_0, \cdots, \boldsymbol{\xi}_t) ,$$

2. taken knowing the available information,

$$\sigma(\boldsymbol{u}_t) \subset \sigma(\boldsymbol{\xi}_0, \cdots, \boldsymbol{\xi}_t) .$$

Multistage extensive formulation approach



Assume that $\xi_t \in \mathbb{R}^{n_\xi}$ can take n_ξ values and that $U_t(x)$ can take n_u values.

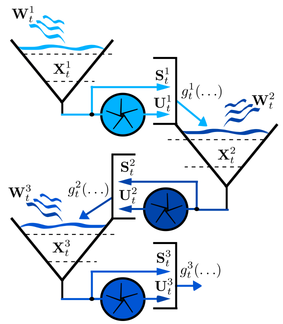
Then, considering the extensive formulation approach, we have

- n_ξ^T scenarios.
- $(n_\xi^{T+1} - 1)/(n_\xi - 1)$ nodes in the tree.
- Number of variables in the optimization problem is roughly $n_u \times (n_\xi^{T+1} - 1)/(n_\xi - 1) \approx n_u n_\xi^T$.

The complexity grows exponentially with the number of stage. :-(

A way to overcome this issue is to compress information!

Illustrating extensive formulation with the damsvalley example



- 5 interconnected dams
- 5 controls per timesteps
- 52 timesteps (one per week, over one year)
- $n_\xi = 10$ noises for each timestep

We obtain 10^{52} scenarios, and $\approx 5.10^{52}$ constraints in the extensive formulation ...
Estimated storage capacity of the Internet: 10^{24} bytes.

2-stage approach

The 2-stage approach consists in approximating the multistage program by a two-stage programm :

- relax all non-anticipativity constraints except the ones on u_0, this turn the tree into a scenario fan (same number of scenario),
- it means that all decision $(u_1, \dots, u_{T-1})$ are anticipative (not u_0).
- reduce the number of scenarios by sampling, and solve the SAA approximation of the 2-stage relaxation.

Denote $v^\sharp$ the value of the multistage problem, v^{2SA} the value of the 2-stage relaxation, and v_m^{2SA} the (random) value of the SAA of the 2-stage relaxation. Then we have

$$v^{2SA} \leq v^\sharp$$
$$v_m^{2SA} \to v^{2SA}$$
$$\mathbb{E}\left[v_m^{2SA}\right] \leq v^{2SA}$$

An Introduction to *Stochastic Dual Dynamic Programming* (SDDP).

V. Leclère (CERMICS, ENPC)

12/01/2022

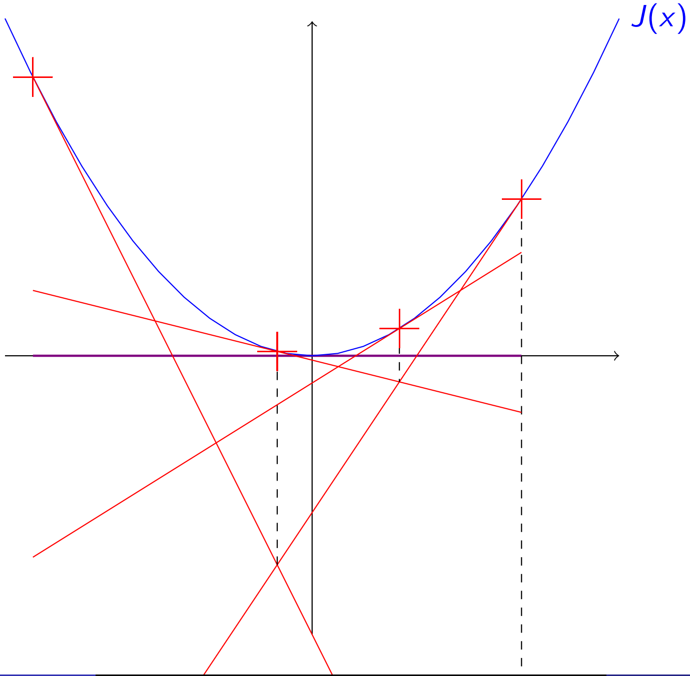
Introduction

- Large scale stochastic optimization problems are hard to solve
- Different ways of attacking such problems:
  - decompose the problem and coordinate solutions
  - construct easily solvable approximations (Linear Programming)
  - find approximate value functions or policies
- Behind the name SDDP, *Stochastic Dual Dynamic Programming*, one finds three different things:
  - a class of algorithms,
    based on specific mathematical assumptions
  - a specific implementation of an algorithm
  - a software implementing this method,
    and developed by the PSR company

Setting

- Multi-stage stochastic optimization problems with finite horizon.
- Continuous, finite dimensional state and control.
- Convex cost, linear dynamic.
- Discrete, stagewise independent noises.

Contents



Kelley algorithm

Data: Convex objective function J, Compact set X, Initial point $x_0 \in X$
Result: Admissible solution $x^{(k)}$, lower-bound $\underline{v}^{(k)}$
Set $J^{(0)} \equiv -\infty$;
for $k \in \mathbb{N}$ **do**
  Compute a subgradient $\alpha^{(k)} \in \partial J(x^{(k)})$;
  Define a cut $\mathcal{C}^{(k)} : x \mapsto J(x^{(k)}) + \langle \alpha^{(k)}, x - x^{(k)} \rangle$;
  Update the lower approximation $J^{(k+1)} = \max\{J^{(k)}, \mathcal{C}^{(k)}\}$;
  Solve $(P^{(k)}) : \quad \min_{x \in X} J^{(k+1)}(x)$;
  Set $\underline{v}^{(k)} = val(P^{(k)})$;
  Select $x^{(k+1)} \in sol(P^{(k)})$;
end

Algorithm 1: Kelley's cutting plane algorithm

Contents

Problem considered

We consider an optimal control problem in discrete time with finite horizon T

$$\begin{aligned} \min_{x \in \mathbb{R}^{nT}} \quad & \sum_{t=0}^{T-1} c_t(x_t, x_{t+1}) + K(x_T) \\ s.t. \quad & (x_t, x_{t+1}) \in P_t, \quad x_0 \text{ given} \\ & x_t \in X_t \end{aligned}$$

- We assume that $P_t \subset \mathbb{R}^n \times X_{t+1}$ is convex, and X_t convex compact
- the transition costs $c_t(x_t, x_{t+1})$ and the final cost $K(x_T)$ are convex

For example, x_t follow a dynamic $x_{t+1} = f_t(x_t, u_t)$, with

- f_t affine, $u_t \in U_t(x_t)$ is convex compact
- $c_t(x_t, x_{t+1}) = \min\{L_t(x_t, u_t) \mid u_t \in U_t(x_t), \; f_t(x_t, u_t) = x_{t+1}\}$, where L_t is a convex instantaneous cost function

Contents

Introducing Bellman's function

We look for solutions as policies, where a policy is a sequence of functions $\pi = (\pi_1, \dots, \pi_{T-1})$ giving for any state x a control u
This problem can be solved by dynamic programming, thanks to the Bellman function that satisfies

$$\begin{cases} V_T(x) &= K(x), \\ \breve{V}_t(x) &= \min_{y:(x,y)\in P_t} \left\{ c_t(x,y) + V_{t+1}(y) \right\} \\ V_t &= \breve{V}_t + \mathbb{I}_{X_t} \end{cases}$$

Indeed, an optimal policy for the original problem is given by

$$\pi_t(x) \in \arg\min_{x_{t+1}} \left\{ c_t(x, x_{t+1}) + V_{t+1}(x_{t+1}) \;\middle|\; (x_t, x_{t+1}) \in P_t \right\}$$

Introducing Bellman's operator

We define the Bellman operator

$$\mathcal{B}_t(A) : x \mapsto \min_{y:(x,y)\in P_t} \left\{ c_t(x,y) + A(y) \right\}$$

With this notation, the Bellman Equation reads

$$\begin{cases} V_T &= K, \\ V_t &= \mathcal{B}_t(V_{t+1}) + \mathbb{I}_{X_t} \end{cases}$$

Any approximate cost function $\breve{V}_{t+1}$ induce an *admissible* policy

$$\pi_t^{\breve{V}_{t+1}} : x \mapsto \arg\min \mathcal{B}_t(\breve{V}_{t+1})(x).$$

By Dynamic Programming, $\pi_t^{V_{t+1}}$ is optimal.

Properties of the Bellman operator

- Monotonicity:

$$V \le \overline{V} \quad \Rightarrow \mathcal{B}_t(V) \le \mathcal{B}_t(\overline{V})$$

- Convexity: if c_t is jointly convex, P and X are closed convex, V is convex then

$$x \mapsto \mathcal{B}_t(V)(x) \quad \text{is convex}$$

- Polyhedrality: for any polyhedral function V, if c_t is also polyhedral, and P_t and X_t are polyhedron, then

$$x \mapsto \mathcal{B}_t(V)(x) \quad \text{is polyhedral}$$

Duality property

- Consider $J : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ jointly convex, and define

$$\varphi(x) = \min_{u \in \mathbb{U}} J(x, u)$$

- Then we can obtain a subgradient $\alpha \in \partial\varphi(x_0)$ as the dual multiplier of

$$\begin{aligned} \min_{x,u} \quad & J(x,u), \\ s.t. \quad & x_0 - x = 0 \qquad [\alpha] \end{aligned}$$

(This is the marginal interpretation of the multiplier)

- In particular, we have that

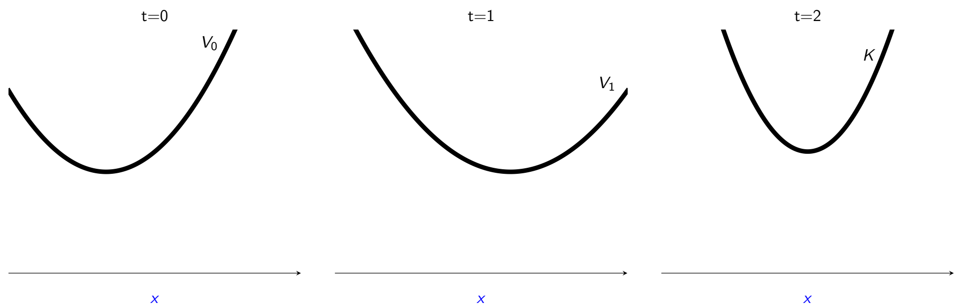
$$\varphi(\cdot) \geq \varphi(x_0) + \langle \alpha, \cdot - x_0 \rangle$$

Contents

General idea

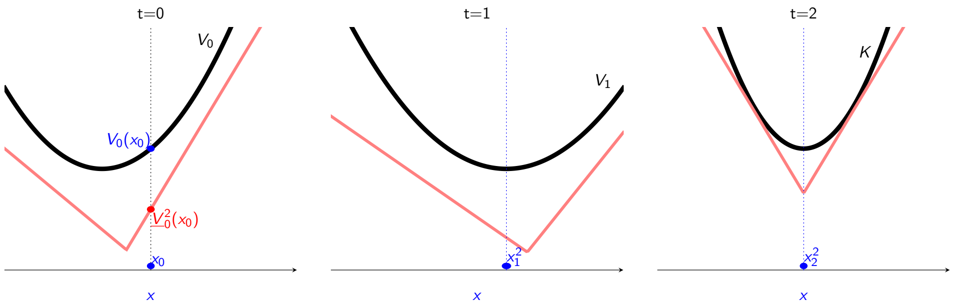
- The SDDP algorithm recursively constructs an approximation of each Bellman function V_t as the supremum of affine functions
- At stage k, we have a lower approximation $\underline{V}_t^{(k)}$ of V_t and we want to construct a better approximation
- We follow an optimal trajectory $(x_t^{(k)})_t$ of the approximated problem, and add a so-called "cut" to improve each Bellman function

Deterministic SDDP



Final Cost $V_2 = K$
Real Bellman function $V_1 = \mathcal{B}_1(V_2)$
Real Bellman function $V_0 = \mathcal{B}_0(V_1)$

Deterministic SDDP

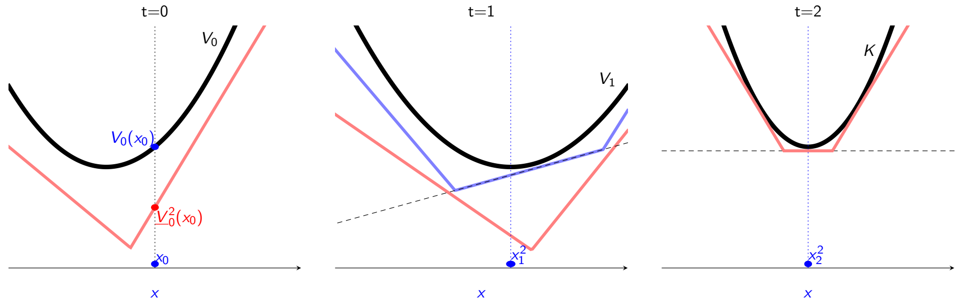


Assume that we have lower polyhedral approximations of V_t

We apply $\pi_0^{\underline{V}_1^{(2)}}$ to x_0 and obtain $x_1^{(2)}$

We apply $\pi_1^{\underline{V}_1^{(2)}}$ to $x_1^{(2)}$ and obtain $x_2^{(2)}$

Deterministic SDDP



Compute a cut for K at $x_2^{(2)}$

Add the cut to $\underline{V}_2^{(2)}$ which gives $\underline{V}_2^{(3)}$

A new lower approximation of V_1 is $\mathcal{B}_1(\underline{V}_2^{(3)})$

We only compute the face active at $x_1^{(2)}$

DDP description

Data: Starting point, initial lower approximation
Result: optimal trajectory and value function;
$V_T \equiv K$;
for $k = 1, 2, \dots$ **do**
  set $x_0^{(k)} = x_0$
  `/* Forward pass : compute trajectory */`
  for $t = 0, \dots, T-1$ **do**
    find $x_{t+1}^{(k)} \in \arg\min \mathcal{B}_t(\underline{V}_{t+1}^{(k)})(x_t^{(k)})$;
  end
  `/* Backward pass : update cuts */`
  for $t = T-1, \dots, 0$ **do**
    Solve $\mathcal{B}_t(\underline{V}_{t+1}^{(k+1)})(x_t^{(k)})$ to compute $\mathcal{C}_t^{(k+1)}$;
    Update lower approximations : $\underline{V}_t^{(k+1)} := \max\{\underline{V}_t^{(k)}, \mathcal{C}_t^{(k+1)}\}$;
  end
end

Algorithm 2: Deterministic Dual Dynamic Programming

Detailing forward pass

- From $t = 0$ to $t = T-1$ we have to solve T one-stage problem of the form

$$x_{t+1}^{(k)} \in \arg\min_{y} \quad c_t(x_t^{(k)}, y) + \underline{V}_{t+1}^{(k)}(y)$$
$$(x_t^{(k)}, y) \in P_t$$

- We only need to keep the trajectory $(x_t^{(k)})_{t \in [\![0,T]\!]}$.

Detailing Backward pass

- From $t = T-1$ to $t = 0$ we have to solve T one-stage problem of the form

$$\begin{aligned} \theta_t^{(k+1)} = \min_{x,y} \quad & c_t(x,y) + \underline{V}_{t+1}^{(k+1)}(y) \\ & (x,y) \in P_t \\ & x = x_t^{(k)} \qquad [\alpha_t^{(k+1)}] \end{aligned}$$

- By construction, we have that

$$\theta_t^{(k+1)} = \mathcal{B}_t\Big(\underline{V}_{t+1}^{(k+1)}\Big)(x_t^{(k)}), \qquad \alpha_t^{(k+1)} \in \partial\mathcal{B}_t\Big(\underline{V}_{t+1}^{(k+1)}\Big)(x_t^{(k)}).$$

- Which means

$$\mathcal{C}_t^{(k+1)} := \theta_t^{(k+1)} + \langle \alpha_t^{(k+1)}, \cdot - x_t^{(k)} \rangle \leq \mathcal{B}_t\Big(\underline{V}_{t+1}^{(k+1)}\Big) \leq \mathcal{B}_t\Big(V_{t+1}\Big) = \tilde{V}_t \leq V_t$$

Contents

Initialization and stopping rule

- To initialize the algorithm, we need a lower bound $\underline{V}_t^{(0)}$ for each value function V_t. This lower bound can be computed backward by arbitrarily choosing a point x_t and using the standard cut computation.
- At any step k we have an admissible, non optimal trajectory $(x_t^{(k)})_t$, with
  - an upper bound

$$\sum_{t=0}^{T-1} c_t\big(x_t^{(k)}, x_{t+1}^{(k)}\big) + K\big(x_T^{(k)}\big)$$

  - a lower bound $\underline{V}_0^{(k)}(x_0)$
- A reasonable stopping rule for the algorithm is given by checking that the (relative) difference between the upper and lower bounds is small enough

Contents

Extended Relatively Complete Recourse

- We say that we are in a relatively complete recourse framework if

$$\forall t, \quad \forall x_t \in X_t, \quad \exists x_{t+1} \in X_{t+1} \quad \text{such that} \quad (x_t, x_{t+1}) \in P_t.$$

- We say that we are in a extended relatively complete recourse framework if there exists $\varepsilon > 0$ such that

$$\forall t, \quad \forall x_t \in X_t + \varepsilon B, \quad \exists x_{t+1} \in X_{t+1} \quad \text{such that} \quad (x_t, x_{t+1}) \in P_t.$$

- RCR is required for the algorithm to run (otherwise we could find non-finite problems, and would require some feasability cuts mechanisms).
- ERCR is required for the convergence proof as the way of ensuring that the multipliers α_t^k remains bounded.

Technical lemmas

Lemma

Let $f : X \to \mathbb{R}$ where X is compact. Let $(f^k)_{k\in\mathbb{N}}$ be a sequence of functions such that

- $f^k \le f^{k+1} \le f$
- *f^k are Lipschitz continuous uniformly in k*

Consider a sequence $(x^k)_{k\in\mathbb{N}}$ of points of X such that $f(x^k) - f^{k+1}(x^k) \to 0$. Then, we also have $f(x^k) - f^k(x^k) \to 0$.

Lemma

Under convexity assumptions, compactness of X_t, and ERCR the SDDP algorithm is well defined and

1. *for all t, V_t is convex and Lipschitz*
2. *for all t, k, and $x \in X_t$, $\underline{V}_t^k \le V_t$*
3. *There exists $L > 0$ such that $\|\alpha_t^k\| \le L$, thus $\underline{V}_t^k$ is L-Lipschitz*

Convergence result

Theorem

Let K and c_t be convex functions, X_t and P_t be closed convex sets, and X_t bounded. Assume that we have extended relatively complete recourse. Then, for every t, we have

$$\lim_k \underline{V}_t^{(k)}(x_t^{(k)}) - V_t(x_t^{(k)}) = 0.$$

Further, the cost associated to $\pi^{\underline{V}_t^{(k)}}$ converges toward the optimal value of the problem.
In other words, the upper and lower bounds are both converging.

Contents

What's new ?

Now we introduce random variables ξ_t in our problem, which complexifies the algorithm in different ways:

- we need some probabilistic assumptions
- for each stage k we need to do a forward phase, for each sequence of realizations of the random variables, that yields a trajectory $(x_t^{(k)})_t$, and a backward phase that gives a new cut
- we cannot compute an exact upper bound for the problem value

Problem statement

We consider the optimization problem

$$
\begin{aligned}
\min \quad & \mathbb{E}\Big[\sum_{t=0}^{T-1} c_t(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{\xi}_{t+1}) + K(\boldsymbol{x}_T)\Big] \\
s.t. \quad & (\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \in P_t(\boldsymbol{\xi}_{t+1}) \\
& \boldsymbol{x}_t \in X_t, \qquad \boldsymbol{x}_0 = x_0 \\
& \boldsymbol{x}_t \preceq \sigma(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_t)
\end{aligned}
$$

under the crucial assumption that $(\boldsymbol{\xi}_t)_{t\in\{1,\cdots,T\}}$ is a white noise

$\rightsquigarrow$ we are in an hazard-decision framework.

Stochastic Dynamic Programming

By the white noise assumption, this problem can be solved by dynamic programming, where the Bellman functions satisfy

$$
\begin{cases}
V_T & = K \\
\hat{V}_t(x,\xi) & = \min\limits_{(x,y)\in P_t(\xi)} c_t(x,y,\xi) + V_{t+1}(y) \\
\tilde{V}_t(x) & = \mathbb{E}\Big[\hat{V}_t(x,\boldsymbol{\xi}_t)\Big] \\
V_t & = \tilde{V}_t + \mathbb{I}_{X_t}
\end{cases}
$$

Indeed, an optimal policy for this problem is given by

$$
\pi_t(x,\xi) \in \underset{(x,y)\in P_t(\xi)}{\arg\min} \big\{ c_t(x,y,\xi) + V_{t+1}(y) \big\}
$$

Bellman operator

For any time t, and any function A mapping the set of states and noises $\mathbb{X}\times\Xi$ into $\mathbb{R}$, we define

$$
\begin{cases}
\hat{\mathcal{B}}_t(A)(x,\xi) & := \min\limits_{(x,y)\in P_t(\xi)} c_t(x,y,\xi) + A(y) \\
\mathcal{B}_t(A)(x) & := \mathbb{E}\Big[\hat{\mathcal{B}}_t(A)(x,\boldsymbol{\xi}_t)\Big]
\end{cases}
$$

Thus the Bellman equation simply reads

$$
\begin{cases}
V_T & = K \\
V_t & = \underbrace{\mathcal{B}_t(V_{t+1})}_{\tilde{V}_t} + \mathbb{I}_{X_t}
\end{cases}
$$

The Bellman operators have the same properties as in the deterministic case

Contents

Computing cuts (1/2)

Suppose that we have $\underline{V}_{t+1}^{(k+1)} \leq V_{t+1}$

$$
\begin{aligned}
\hat{\theta}_t^{(k+1)}(\xi) = \min_{x,y} \quad & c_t(x,y,\xi) + \underline{V}_{t+1}^{(k+1)}(y) \\
s.t \quad & x = x_t^{(k)} \qquad [\hat{\alpha}_t^{(k+1)}(\xi)] \\
& (x,y) \in P_t(\xi)
\end{aligned}
$$

This can also be written as

$$
\hat{\theta}_t^{(k+1)}(\xi) = \hat{\mathcal{B}}_t\Big[\underline{V}_{t+1}^{(k+1)}\Big](x,\xi)
$$

$$
\hat{\alpha}_t^{(k+1)}(\xi) \in \partial_x \hat{\mathcal{B}}_t\Big[\underline{V}_{t+1}^{(k+1)}\Big](x,\xi)
$$

Thus, for all ξ, $\hat{\mathcal{C}}_t^{(k+1),\xi} : x \mapsto \hat{\theta}_t^{(k+1)}(\xi) + \left\langle \hat{\alpha}_t^{(k+1)}(\xi), x - x_t^{(k)} \right\rangle$ satisfy

$$
\hat{\mathcal{C}}_t^{(k+1),\xi}(x) \leq \hat{\mathcal{B}}_t\Big[\underline{V}_{t+1}^{(k+1)}\Big](x,\xi) \leq \hat{\mathcal{B}}_t\Big[V_{t+1}\Big](x,\xi) = \hat{V}_t(x,\xi)
$$

Computing cuts (2/2)

Thus, we have an affine minorant of $\hat{V}_t(x,\boldsymbol{\xi}_t)$ for each realization of $\boldsymbol{\xi}_t$
Replacing ξ by the random variable $\boldsymbol{\xi}_t$ and taking the expectation yields the following affine minorant

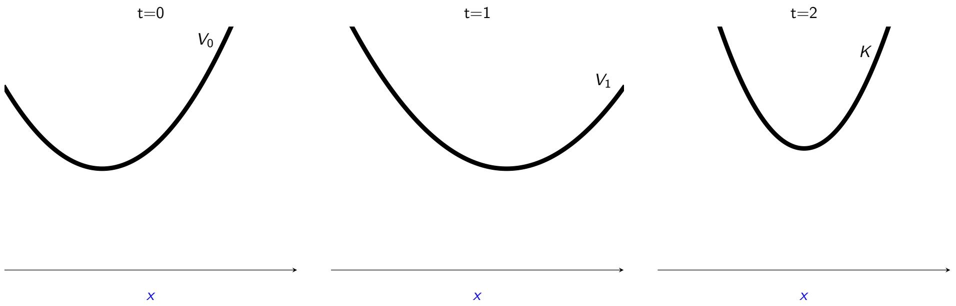
$$
\mathcal{C}^{(k+1)} := \theta_t^{(k+1)} + \left\langle \alpha_t^{(k+1)}, \cdot - x_t^{(k)} \right\rangle \leq V_t
$$

where

$$
\begin{cases}
\theta_t^{(k+1)} & := \quad \mathbb{E}\Big[\hat{\theta}_t^{(k+1)}(\boldsymbol{\xi}_t)\Big] = \mathcal{B}_t\Big[\underline{V}_{t+1}^{(k)}\Big](x) \\
\alpha_t^{(k+1)} & := \quad \mathbb{E}\Big[\hat{\alpha}_t^{(k+1)}(\boldsymbol{\xi}_t)\Big] \in \partial\mathcal{B}_t\Big[\underline{V}_{t+1}^{(k)}\Big](x)
\end{cases}
$$

Contents

Abstract SDDP

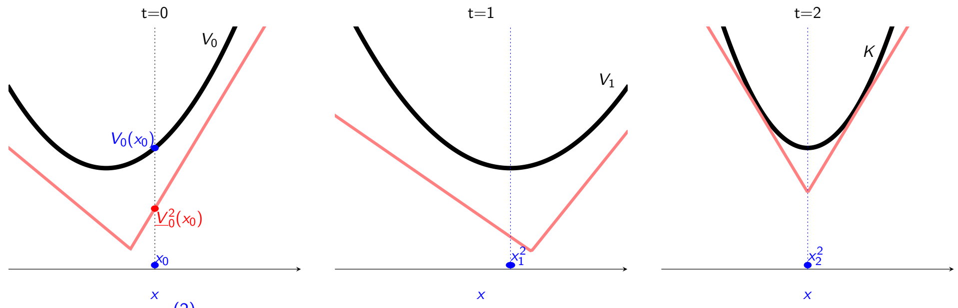


Final Cost $V_2 = K$

Real Bellman function $V_1 = \mathcal{B}_1(V_2)$

Real Bellman function $V_0 = \mathcal{B}_0(V_1)$

Abstract SDDP



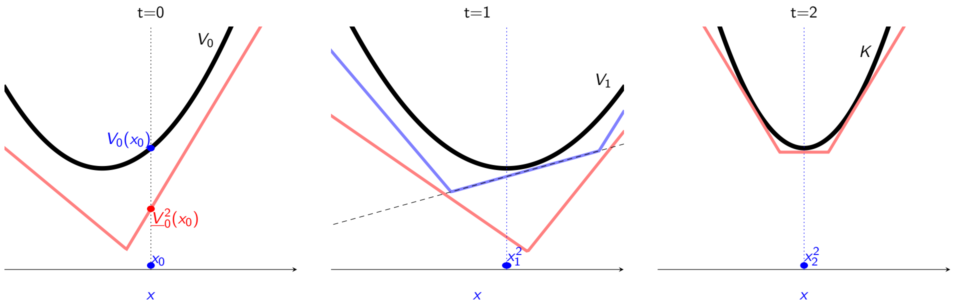
Apply $\pi_0^{\underline{V}_1^{(2)}}$ to x_0 and obtain $\boldsymbol{X}_1^{(2)}$

Draw a random realisation $x_1^{(2)}$ of $\boldsymbol{X}_1^{(2)}$

We apply $\pi_1^{\underline{V}_1^{(2)}}$ to $x_1^{(2)}$ and obtain $\boldsymbol{X}_2^{(2)}$

Draw a random realisation $x_2^{(2)}$ of $\boldsymbol{X}_2^{(2)}$

Abstract SDDP



Compute a cut for K at $x_2^{(2)}$

Add the cut to $\underline{V}_2^{(2)}$ which gives $\underline{V}_2^{(3)}$

A new lower approximation of V_1 is $\mathcal{B}_1(\underline{V}_2^{(3)})$

Compute the face active at $x_1^{(2)}$

SDDP description

for $k = 1, 2, \dots$ **do**

  set $\underline{V}_T^{(k+1)} \equiv K$; $x_0^{(k)} = x_0$;

  draw $(\xi_t^{(k)})_{t \in [\![1,T]\!]}$;

  `/* Forward pass : compute trajectory */`

  for $t = 0, \dots, T-1$ **do**

    find $x_{t+1}^{(k)} \in \arg\min \hat{\mathcal{B}}_t(\underline{V}_{t+1}^{(k)})(x_t^{(k)}, \xi_t^{(k)})$;

  end

  `/* Backward pass : update cuts */`

  for $t = T-1, \dots, 0$ **do**

    for $\xi \in \Xi_t$ **do**

      Solve $\hat{\mathcal{B}}_t(\underline{V}_{t+1}^{(k+1)})(x_t^{(k)}, \xi)$ to compute $\hat{\mathcal{C}}_t^{(k+1),\xi}$;

    end

  end

  Compute averaged cut : $\mathcal{C}_t^{(k+1)}$;

  Update lower approximation : $\underline{V}_t^{(k+1)} := \max\{\underline{V}_t^{(k)}, \mathcal{C}_t^{(k+1)}\}$;

end

Algorithm 3: Stochastic Dual Dynamic Programming

Detailing forward pass

- From $t=0$ to $t=T-1$ we have to solve T one-stage problem of the form

$$x_{t+1}^{(k)} \in \arg\min_{y} \quad c_t(x_t^{(k)}, y, \xi_t^{(k)}) + \underline{V}_{t+1}^{(k)}(y)$$
$$(x_t^{(k)}, y) \in P_t$$

- We only need to keep the trajectory $(x_t^{(k)})_{t\in[\![0,T]\!]}$.

Detailing Backward pass

- For each $t = T-1 \to 0$ we solve Ξ_t one-stage problem

$$\hat\theta_t^{(k+1)}(\xi) = \min_{y} \quad c_t(x_t^{(k)}, y, \xi) + \underline{V}_{t+1}^{(k+1)}(y)$$
$$(x_t^{(k)}, y) \in P_t$$
$$x = x_t^{(k)} \qquad [\hat\alpha_t^{(k+1)}(\xi)]$$

- By construction, we have that

$$\hat\theta_t^{(k+1)}(\xi) = \mathcal{B}_t\Big(\underline{V}_{t+1}^{(k)}\Big)(x_t^{(k)}, \xi), \qquad \hat\alpha_t^{(k+1)}(\xi) \quad \in \partial\mathcal{B}_t\Big(\underline{V}_{t+1}^{(k)}\Big)(x_t^{(k)}, \xi).$$

- We average the coefficients

$$\theta_t^{(k+1)} = \mathbb{E}\big[\hat\theta_t^{(k+1)}(\boldsymbol{\xi})\big], \qquad \alpha_t^{(k+1)} = \mathbb{E}\big[\hat\alpha_t^{(k+1)}(\boldsymbol{\xi})\big]$$

- Which means

$$\mathcal{C}_t^{(k+1)} := \theta_t^{(k+1)} + \langle \alpha_t^{(k+1)}, \cdot - x_t^{(k)}\rangle \le \mathcal{B}_t\Big(\underline{V}_{t+1}^{(k+1)}\Big) \le \mathcal{B}_t\Big(V_{t+1}\Big) = \tilde V_t \le V_t$$

Recall on CLT

- Let $\{C_i\}_{i\in\mathbb{N}}$ be a sequence of identically distributed random variables with finite variance.
- Then the Central Limit Theorem ensures that

$$\sqrt{n}\Big(\frac{\sum_{i=1}^n \boldsymbol{C}_i}{n} - \mathbb{E}[\boldsymbol{C}_1]\Big) \Longrightarrow G \sim \mathcal{N}(0, Var[\boldsymbol{C}_1])\ ,$$

where the convergence is in law.
- In practice it is often used in the following way. Asymptotically,

$$\mathbb{P}\Big(\mathbb{E}\big[C_1\big] \in \Big[\bar{\boldsymbol{C}}_n - \frac{1.96\boldsymbol{\sigma}_n}{\sqrt{n}}, \bar{\boldsymbol{C}}_n + \frac{1.96\boldsymbol{\sigma}_n}{\sqrt{n}}\Big]\Big) \simeq 95\%\ ,$$

where $\bar{\boldsymbol{C}}_n = \frac{\sum_{i=1}^n \boldsymbol{C}_i}{n}$ is the empirical mean and $\boldsymbol{\sigma}_n = \sqrt{\frac{\sum_{i=1}^n(\boldsymbol{C}_i - \bar{\boldsymbol{C}}_n)^2}{n-1}}$ the empirical standard deviation.

Bounds

- Exact lower bound on the value of the problem: $\underline{V}_0^{(k)}(x_0)$.
- Exact upper bound on the value of the problem:

$$\mathbb{E}\Big[\sum_{t=0}^{T-1} c_t(\boldsymbol{x}_t^{(k)}, \boldsymbol{x}_{t+1}^{(k)}, \boldsymbol{\xi}_{t+1}) + K(\boldsymbol{X}_T)\Big]$$

where $\boldsymbol{X}_t^{(k)}$ is the trajectory induced by $\underline{V}_t^{(k)}$.
- This bound cannot be computed exactly, but can be estimated by Monte-Carlo method as follows
  - Draw N scenarios $\{\xi_1^n, \dots, \xi_T^n\}$.
  - Simulate the corresponding N trajectories $x_t^{(k),n}$, and the total cost for each trajectory $C^{(k),n}$.
  - Compute the empirical mean $\bar C^{(k),N}$ and standard dev. $\sigma^{(k),N}$.
  - Then, with confidence 95% the upper bound on the problem is

$$\Big[\bar C^{(k),N} - \frac{1.96\sigma^{(k),N}}{\sqrt{N}}, \underbrace{\bar C^{(k),N} + \frac{1.96\sigma^{(k),N}}{\sqrt{N}}}_{UB_k}\Big]$$

Stopping rule

- One stopping test consist in fixing an a priori relative gap ε, and stopping if
$$\frac{UB_k - V_0^{(k)}(x_0)}{V_0^{(k)}(x_0)} \leq \varepsilon$$
in which case we know that the solution is ε-optimal with probability 97.5%.
- It is not necessary to evaluate the gap at each iteration.
- To alleviate the computational load, we can estimate the upper bound by using the trajectories of the recent forward phases.
- Another more practical stopping rule consists in stopping after a given number of iterations or fixed computation time.

Contents

Non-independent inflows

- In most cases the stagewise independence assumption is not realistic.
- One classical way of modelling dependencies consists in considering that the inflows I_t follow an AR-k process
$$I_t = \alpha_1 I_{t-1} + \cdots + \alpha_k I_{t-k} + \theta_t + \xi_t$$
where ξ_t is the residual, forming an independent sequence.
- The state of the system is now $(X_t, I_{t-1}, \dots, I_{(t-k)})$.

Implementations and numerical tricks

- We can play with the number of forward / backward pass. Classically we do 200 forward passes in parallel, before computing cuts.
- Instead of averaging the cuts, we can keep one cut per alea, for a multicut version. In other word instead of representing V_t we represent $\hat{V}_t$.
- Early forward passes are not really usefull, selecting (randomly or by hand) a few trajectory can save some workload.
- Cut pruning (eliminating useless cuts) is easy to implement and pretty efficient.
- Adding some regularization term in the forward pass has shown some numerical improvement but is not yet fully understood.

Cut Selection methods I

- Let $\underline{V}_t^{(k)}$ be defined as $\max_{\ell \leq k} \mathcal{C}_t^{(\ell)}$
- For $j \leq k$, if

$$\begin{aligned} \min_{x,\alpha} \quad & \alpha - \mathcal{C}_t^{(j)}(x) \\ s.t. \quad & \alpha \geq \mathcal{C}_t^{(\ell)}(x) \qquad \forall \ell \neq j \end{aligned}$$

  is non-negative, then cut j can be discarded without modifying $\underline{V}_t^{(k)}$
- this technique is exact but time-consuming.

Cut Selection methods II

- Instead of comparing a cut everywhere, we can choose to compare it only on the already visited points.
- The Level-1 cut method goes as follow:
  - keep a list of all visited points $x_t^{(\ell)}$ for $\ell \leq k$.
  - for ℓ from 1 to k, tag each cut that is active at $x_t^{(\ell)}$.
  - Discard all non-tagged cut.

Contents

Coherent Risk Measure I

To take into account some risk aversion we can replace the expectation by a *risk measure*. A risk measure is a function giving to a random cost $\boldsymbol{X}$ a deterministic equivalent $\rho(\boldsymbol{X})$ A Coherent Risk Measure $\rho : L^\infty(\Omega, \mathcal{F}, \mathbb{P}) \to \mathbb{R}$ is a functionnal satisfying

- Monotonicity: if $\boldsymbol{X} \geq \boldsymbol{Y}$ then $\rho(\boldsymbol{X}) \geq \rho(\boldsymbol{Y})$,
- Translation equivariance: for $c \in \mathbb{R}$ we have $\rho(\boldsymbol{X} + c) = \rho(\boldsymbol{X}) + c$,
- Convexity: for $t \in [0,1]$, we have

$$\rho(t\boldsymbol{X} + (1-t)\boldsymbol{Y}) \leq t\rho(\boldsymbol{X}) + (1-t)\rho(\boldsymbol{Y}),$$

- Positive homogeneity: for $\alpha \in \mathbb{R}^+$, we have $\rho(\alpha \boldsymbol{X}) = \alpha\rho(\boldsymbol{X})$.

Coherent Risk Measure II

From convex analysis we obtain the main theorem over coherent risk measure.

Theorem

Let ρ be a coherent risk measure, then there exists a (convex) set of probability $\mathcal{P}$ such that

$$\forall \boldsymbol{X}, \qquad \rho(\boldsymbol{X}) = \sup_{\mathbb{Q}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[\boldsymbol{X}].$$

Average Value at Risk I

One of the most practical and used coherent risk measure is the Average Value at Risk at level α. Roughly, it is the expectation of the cost over the α-worst cases. For a random variable $\boldsymbol{X}$ admitting a density, we define de value at risk of level α, as the quantile of level α, that is

$$VaR_\alpha(\boldsymbol{X}) = \inf\Big\{t\in\mathbb{R} \quad | \quad \mathbb{P}(\boldsymbol{X}\geq t)\leq \alpha\Big\}.$$

And the average value at risk is

$$AVaR_\alpha(\boldsymbol{X}) = \mathbb{E}\big[\boldsymbol{X} \;\big|\; \boldsymbol{X}\geq VaR_\alpha(\boldsymbol{X})\big]$$

Average Value at Risk II

One of the best aspect of the AVaR, is the following formula

$$AVaR_\alpha(\boldsymbol{X}) = \min_{t\in\mathbb{R}}\Big\{t + \frac{\mathbb{E}\big[X-t\big]^+}{\alpha}\Big\}.$$

Indeed it allow to linearize the AVaR.

SDDP and risk

- The problem studied was risk neutral
- However a lot of works has been done recently about how to solve risk averse problems
- Most of them are using AVAR, or a mix between AVAR and expectation either as objective or constraint
- Indeed AVAR can be used in a linear framework by adding other variables
- Another easy way is to use "composed risk measures"
- Finally a convergence proof with convex costs (instead of linear costs) exists, although it requires to solve non-linear problems

Contents

Assumptions

- Noises are time-independent, with finite support.
- X_t is convex compact, P_t is closed convex.
- Costs are convex and lower semicontinuous.
- We are in a strong relatively complete recourse framework.

Remark, if we take the tree-view of the algorithm

- stage-independence of noise is not required to have theoretical convergence
- node-selection process should be admissible (e.g. independent, SDDP, CUPPS...)

Convergence result

Theorem

With the preceding assumption, we have that the upper and lower bound are almost surely converging toward the optimal value, and we can obtain an $\varepsilon-$optimal strategy for any $\varepsilon > 0$.
More precisely, if we call $\underline{V}_t^{(k)}$ the outer approximation of the Bellman function V_t at step k of the algorithm, and $\pi_t^{(k)}$ the corresponding strategy, we have

$$\underline{V}_0^{(k)}(x_0) \to_k V_0(x_0)$$

and

$$\mathbb{E}\Big[c_t\big(\boldsymbol{x}_t^{(k)}, \boldsymbol{x}_{t+1}^{(k)}, \boldsymbol{\xi}_t\big) + \underline{V}_{t+1}^{(k)}\big(\boldsymbol{x}_{t+1}^{(k)}\big)\Big] - V_t\big(\boldsymbol{x}_t^{(k)}\big) \to_k 0.$$

Conclusion

SDDP is an algorithm, more precisely a class of algorithms, that

- exploits convexity of the value functions (from convexity of costs...)
- does not require state discretization
- constructs outer approximations of V_t, those approximations being precise only "in the right places"
- gives bounds:
  - "true" lower bound $\underline{V}_0^{(k)}(x_0)$
  - estimated (by Monte-Carlo) upper bound
- constructs linear-convex approximations, thus enabling to use linear solver like CPLEX
- can be shown to display asymptotic convergence

Bibliography

- R. Van Slyke and R. Wets (1969).
  L-shaped linear programs with applications to optimal control and stochastic programming.
  SIAM Journal on Applied Mathematics

- M. Pereira, L.Pinto (1991).
  Multi-stage stochastic optimization applied to energy planning
  Mathematical Programming

- A. Shapiro (2011).
  Analysis of stochastic dual dynamic programming method.
  European Journal of Operational Research.

- P.Girardeau, V.Leclère, A. Philpott (2014).
  On the convergence of decomposition methods for multi-stage stochastic convex programs.
  Mathematics of Operations Research.