Stochastic Programming

Stochastic programming with recourse

François Pacaud and Welington de Oliveira

frapac.github.io

Centre Automatique et Systèmes - CAS
École Nationale Supérieure des Mines de Paris - Mines Paris PSL

C1MINES-09 PSL week: Optimisation Stochastique, November 2024

Outline

- SP with recourse: Two-stage Linear Programming - 2SLP
- SP with recourse: The L-Shaped Method
- SP with recourse: The Progressive-Hedging Method
- SP with recourse: Multistage Linear Programming - MSLP
- SP with recourse: Sample Average Approximation - SAA
- SP with recourse: Estimating optimality gap

Two-stage Linear Programming - 2SLP

Stochastic programming with recourse

Consider the LP

$$\begin{cases} \min\limits_{(x,y)\geq 0} & c^\top x + q^\top y \\ s.t. & Ax = b \\ & Tx + Wy = h \end{cases}$$

Now suppose that some (or all) the data q, T, W, h depend on some random vector ω:

$$q(\omega),\ T(\omega),\ W(\omega),\ h(\omega)$$

Decisions are sequential in nature:

$$x \rightsquigarrow \omega \rightsquigarrow y$$

We shall to give a "meaning" to the random LP: $q(\omega)^\top y(\omega)$ is now random!

$$\begin{cases} \min\limits_{(x,y(\omega))\geq 0} & c^\top x + q(\omega)^\top y(\omega) \\ s.t. & Ax = b \\ & T(\omega)x + W(\omega)y(\omega) = h(\omega) \end{cases}$$

A manner is to minimize the expected cost

Two-stage stochastic linear programs with recourse - 2SLP

Minimizing the present cost + the expected value of the future costs

Optimization on average

$$\begin{cases} \min\limits_{(x,y(\omega))\geq 0} & c^\top x + \mathbb{E}[q(\omega)^\top y(\omega)] \\ s.t. & Ax = b \\ & T(\omega)x + W(\omega)y(\omega) = h(\omega) \ a.s. \end{cases}$$

(a.s. = almost surely)

- x represents the *here-and-now* variables
(the decisions we have to make in the present)
- $y(\omega)$ represents the *wait-and-see* decisions, a.k.a *recourse*
(the decisions we have to make in the future, depending on the scenario)
- $W(\omega)$ is the matrix of recourse, and the matrix of technologies $T(\omega)$ couples the variables x and y

Finitely many scenarios

In two-stage stochastic linear programming problems with finitely many scenarios (q^i, T^i, W^i, h^i), $i = 1, \ldots, N$, we wish to solve the high dimensional problem

Deterministic equivalent

$$\begin{cases} \min\limits_{x, y^i} & c^\top x + \sum_{i=1}^N p_i [q^{i\,\top} y^i] \\ \text{s.t.} & Ax = b, \; x \geq 0 \\ & T^i x + W^i y^i = h^i, \; y^i \geq 0, \; i = 1, \ldots, N \end{cases}$$

Finitely many scenarios

In two-stage stochastic linear programming problems with finitely many scenarios (q^i, T^i, W^i, h^i), $i = 1, \ldots, N$, we wish to solve the high dimensional problem

Deterministic equivalent

$$\begin{cases} \min\limits_{x, y^i} & c^\top x + \sum_{i=1}^N p_i [q^{i\,\top} y^i] \\ \text{s.t.} & Ax = b, \; x \geq 0 \\ & T^i x + W^i y^i = h^i, \; y^i \geq 0, \; i = 1, \ldots, N \end{cases}$$

This is a LP with a block-arrowhead structure!

$$\begin{cases} \begin{array}{lllllll} \min & c^\top x & +p_1 q^{1\,\top} y^1 & +p_2 q^{2\,\top} y^2 & + \cdots & +p_N q^{N\,\top} y^N & \\ \text{s.t} & Ax = b & & & & & \\ & & & & & & \\ & T^1 x & +W^1 y^1 & & & & = h^1 \\ & T^2 x & & +W^2 y^2 & & & = h^2 \\ & \vdots & & & \ddots & & \\ & T^N x & & & & +W^N y^N & = h^N \\ & & & & & & \\ & (x, y) \geq 0 & & & & & \end{array} \end{cases}$$

Some orders of magnitude

How fast can we solve a large-scale LP using a state-of-the-art solver?

n	m	Solve time (s)
85M	98M	6,000
223M	254M	66,100

Table: Time to solve a LP with n variables, m constraints

Source:

- Rehfeldt, D., Hobbie, H., Schönheit, D., Koch, T., Möst, D., & Gleixner, A. (2022). *A massively parallel interior-point solver for LPs with generalized arrowhead structure, and applications to energy system models.*

Finitely many scenarios

In two-stage stochastic linear programming problems with finitely many scenarios (q^i, T^i, W^i, h^i), $i = 1, \dots, N$, we wish to solve the high dimensional problem

Deterministic equivalent problem

$$
\left\{
\begin{array}{llllllll}
\min & c^\top x & +p_1 {q^1}^\top y^1 & +p_2 {q^2}^\top y^2 & +\cdots & +p_N {q^N}^\top y^N & \\
\text{s.t} & x \in X,\ y \in Y & & & & & \\
& T^1 x & +W^1 y^1 & & & & = h^1 \\
& T^2 x & & +W^2 y^2 & & & = h^2 \\
& \vdots & & & \ddots & & \\
& T^N x & & & & +W^N y^N & = h^N
\end{array}
\right.
$$

We can have "mixed-integer constraints" in X and Y

However, depending on N the deterministic equivalent problem cannot be solved directly...

- # variables: $n_x + N\, n_y$
- # constraints: $m_x + N\, n_y$

The deterministic equivalent problem is only useful when N is small enough...

The deterministic equivalent problem can be too large to be solved directly

$\boxed{\text{We need to decompose the problem}}$

For convenience of notation, let's write the 2SLP more compactly as

$$\begin{cases} \min\limits_{(x,y)\geq 0} & c^\top x + \mathbb{E}[q^\top y] \\ s.t. & Ax = b \\ & Tx + Wy = h \quad a.s. \end{cases}$$

We define also the random vector

$$\xi(\omega) := (q(\omega), h(\omega), T(\omega), W(\omega)) \quad \text{or simply} \quad \xi := (q, h, T, W)$$

and split the problem according to first and second-stage variables

Two-stage decomposition

Value function

For each pair (x, ξ), define the simple LP

$$Q(x,\xi) := \begin{cases} \min & q^\top y \\ \text{s.t.} & Wy = h - Tx \\ & y \geq 0 \end{cases}$$

The program 2SLP is equivalent to

$$\begin{cases} \min & c^\top x + \mathbb{E}\left[Q(x,\xi)\right] \\ \text{s.t.} & Ax = b\,, \quad x \geq 0\,, \end{cases}$$

- For fixed $\tilde{\xi}$, when is $Q(\cdot, \tilde{\xi})$ finite?
- What does $Q(x, \tilde{\xi}) = -\infty$ mean?
- What does $Q(x, \tilde{\xi}) = +\infty$ mean?

Value function

For each pair (x, ξ), define the simple LP

$$Q(x,\xi) := \begin{cases} \min & q^\top y \\ \text{s.t.} & Wy = h - Tx \\ & y \geq 0 \end{cases}$$

The program 2SLP is equivalent to

$$\begin{cases} \min & c^\top x + \mathbb{E}\left[Q(x,\xi)\right] \\ \text{s.t.} & Ax = b\,, \quad x \geq 0\,, \end{cases}$$

- For fixed $\tilde{\xi}$, when is $Q(\cdot, \tilde{\xi})$ finite?
- What does $Q(x, \tilde{\xi}) = -\infty$ mean? (no solution - depending on $\mathbb{P}(\tilde{\xi})$)
- What does $Q(x, \tilde{\xi}) = +\infty$ mean? (infeasibility!)

Well-posedness

Fixed recourse

The two-stage problem has *fixed* recourse if W does not depend on ω.

Complete recourse

The two-stage problem has *complete* recourse if the system $Wy = z$ has a solution for every z (ensure feasibility for all $z := h - Tx$)

Relatively complete recourse

The two-stage problem has *relatively complete* recourse if for every x in the set $\{x \ : \ Ax = b \ , \ x \geq 0\}$ and for every $\xi \in \Xi$ the feasible set is non-empty:

$$Y(x, \xi) \neq \emptyset$$

with

$$Y(x, \xi) = \{y \ : \ Tx + Wy = h \ , \ y \geq 0\}$$

Properties of the value function

The dual of the second-stage problem is

$$\min_{u} \; u^\top (h - Tx) \quad \text{s.t.} \quad W^\top u \le q$$

Structure

- The function $Q(\cdot, \xi)$ is convex.
- If $\{u \; : \; W^\top u \le q\}$ is non-empty and second-stage problem is feasible, then $Q(\cdot, \xi)$ is *polyhedral*.

Dual reformulation

Suppose $Y(x, \xi) \ne \emptyset$. Then

$$Q(x, \xi) = \max_{u} \; u^\top (h - Tx) \quad \text{s.t.} \quad W^\top u \le q$$

Subdifferentiability

Suppose for (x, ξ) $Q(x, \xi) < +\infty$. Then $Q(\cdot, \xi)$ is subdifferentiable at x, with

$$\partial Q(x, \xi) = -T^\top \mathcal{D}(x, \xi) \quad \text{where} \quad \mathcal{D}(x, \xi) := \arg\max_{u \in \Lambda(q)} u^\top (h - Tx)$$

Suppose we have a finite number of scenarios $\xi^1, \cdots, \xi^N$. Let

$$\phi(x) := \mathbb{E}\big[Q(x,\xi)\big] = \sum_{i=1}^{N} p_i Q(x,\xi_i)$$

The two-stage program is equivalent to

$$\begin{cases} \min & c^\top x + \sum_{i=1}^{N} p_i Q(x,\xi_i) \\ \text{s.t.} & Ax = b\,, \quad x \geq 0\,, \end{cases}$$

Proposition

Suppose there exists x_0 such that $\phi(x_0) < +\infty$.
Then $\phi(\cdot)$ is polyhedral, and for all $x \in \mathrm{dom}(\phi)$,

$$\partial\phi(x) = \sum_{i=1}^{N} p_k \partial Q(x,\xi_i)$$

The L-Shaped Method

Decomposition

We aim at solving the two-stage program:

$$\begin{cases} \min & c^\top x + \sum_{i=1}^N p_i Q(x, \xi_i) \\ \text{s.t.} & Ax = b\,, \quad x \geq 0\,, \end{cases}$$

Reformulation:

Decomposition

We aim at solving the two-stage program:

$$\begin{cases} \min & c^\top x + \sum_{i=1}^N p_i Q(x, \xi_i) \\ \text{s.t.} & Ax = b\,, \quad x \geq 0\,, \end{cases}$$

Reformulation:

- $Q(\cdot, \xi)$ is polyhedral

$$Q(x, \xi) = \max_{k=1,\cdots,V} \{\beta_k x + \alpha_k\}$$

Decomposition

We aim at solving the two-stage program:

$$\left\{\begin{array}{ll} \min & c^\top x + \sum_{i=1}^N p_i r_i \\ \text{s.t.} & Ax = b\,, \quad x \geq 0\,, \\ & r_i \geq \alpha_k^i + \beta_k^i x \quad \forall k = 1, \cdots, V \quad \forall i = 1, \cdots, k \end{array}\right.$$

Reformulation:

- $Q(\cdot, \xi)$ is polyhedral

$$Q(x,\xi) = \max_{k=1,\cdots,V} \{\beta_k x + \alpha_k\}$$

- $Q(\cdot, \xi)$ is convex

$$\min_x \; Q(x,\xi) \quad \equiv \quad \min_{x,r} \; r \quad \text{s.t.} \quad r \geq \beta_k x + \alpha_k \quad \forall k = 1, \cdots, V$$

Decomposition: multi-cut version

Given x^k, The Benders' decomposition computes a vertex by solving, for $i = 1, \cdots, N$,

$$\text{(Lower problem)} \quad u_i^k \in \arg\max_u \, (h_i - T_i x^k)^\top u \quad \text{s.t.} \quad W_i^\top u \leq q_i$$

and updates (x^{k+1}, r^{k+1}) by solving the LP

$$\text{(Upper problem)} \quad \left\{ \begin{array}{ll} \min\limits_{x, r_1, \cdots, r_N} & c^\top x + \sum_{i=1}^N p_i r_i \\ \text{s.t.} & Ax = b \, , \; x \geq 0 \\ & (h_i - T_i x)^\top u_i^\ell \leq r_i \quad i = 1, \ldots, N \quad \forall \ell = 1, \cdots, k \end{array} \right.$$

The Upper Problem

(Upper problem) $$\begin{cases} \min\limits_{x, r_1, \cdots, r_N} & c^\top x + \sum_{i=1}^N p_i r_i \\ \text{s.t.} & Ax = b \ , \ x \geq 0 \\ & (h_i - T_i x)^\top u_i^\ell \leq r_i \quad i = 1, \dots, N \quad \forall \ell = 1, \cdots, k \end{cases}$$

is equivalent to

$$\begin{cases} \min\limits_{x, r} & c^\top x + r \\ \text{s.t.} & Ax = b \ , \ x \geq 0 \\ & \sum_{i=1}^N p_i (h_i - T_i x)^\top u_i^\ell \leq r \quad \forall \ell = 1, \dots, k \end{cases}$$

with

$$u_i^\ell \in \arg\max_u \ (h_i - T_i x^\ell)^\top u \quad \text{s.t.} \quad W_i^\top u \leq q_i$$

The Benders' decomposition applied to the LP

$$
\left\{
\begin{array}{llllllr}
\min & c^\top x & +p_1 {q^1}^\top y^1 & +p_2 {q^2}^\top y^2 & + \cdots & +p_N {q^N}^\top y^N & \\
\text{s.t} & Ax = b & & & & & \\
& & & & & & \\
& T^1 x & +W^1 y^1 & & & & = h^1 \\
& T^2 x & & +W^2 y^2 & & & = h^2 \\
& \vdots & & & \ddots & & \\
& T^N x & & & & +W^N y^N & = h^N \\
& & & & & & \\
& (x, y) \geq 0 & & & & &
\end{array}
\right.
$$

is known as the **L-Shaped method**

The L-Shaped Method

Given x^1 feasible, set $k = 1$ and $UB^0 = +\infty$

1. Send x^k to the Lower Problems: for $i = 1, \dots, N$, compute a new vertex u_i^k by solving

$$\text{(Lower Problem)} \quad Q(x^k, \xi_i) = \begin{cases} \max\limits_{u} & (h_i - T_i x^k)^\top u \\ \text{s.t.} & W_i^\top u \le q_i \end{cases}$$

   Set $\mathbf{Q}(x^k) = \sum_{i=1}^N p_s Q(x^k, \xi_i)$ and $UB^k = \min\{UB^{k-1}, c^\top x^k + \mathbf{Q}(x^k)\}$

2. Find (x^{k+1}, r^{k+1}) by solving the LP

$$\text{(Upper problem)} \quad \begin{cases} \min\limits_{x,r} & c^\top x + r \\ \text{s.t.} & Ax = b\ ,\ x \ge 0 \\ & \sum_{s=1}^N p_s (h^s - T^s x)^\top u^{i,s} \le r \quad i = 1, \dots, k \end{cases}$$

3. If $UB^k - [c^\top x^{k+1} + r^{k+1}] \le \delta_{\texttt{Tol}}$, stop
4. Set $k = k + 1$ and go back Step 1

The algorithm stops after finitely many steps (even if $\delta_{\texttt{Tol}} = 0$) at the solution without enumerating all the vertices

Benders' decomposition for 2SLP

The L-Shaped Method

Given x^1 feasible, set $k = 1$ and $UB^0 = +\infty$

1. Send x^k to the Lower Problems: for $i = 1, \dots, N$, compute a new vertex u_i^k by solving

$$\text{(Lower Problem)} \quad Q(x^k, \xi_i) = \begin{cases} \max\limits_{u} & (h_i - T_i x^k)^\top u \\ \text{s.t.} & W_i^\top u \leq q_i \end{cases}$$

   Set $\mathbf{Q}(x^k) = \sum_{i=1}^N p_s Q(x^k, \xi_i)$ and $UB^k = \min\{UB^{k-1}, c^\top x^k + \mathbf{Q}(x^k)\}$ and

$$\beta^k = -\sum_{i=1}^N p_i [T_i^\top u_i^k] \text{ and } \alpha^k = \sum_{i=1}^N p_i [h_i^\top u_i^k]$$

2. Find (x^{k+1}, r^{k+1}) by solving the LP

$$\text{(Upper problem)} \quad \begin{cases} \min\limits_{x,r} & c^\top x + r \\ \text{s.t.} & Ax = b \ , \ x \geq 0 \\ & {\beta^i}^\top x + \alpha^i \leq r \quad i = 1, \dots, k \end{cases}$$

3. If $UB^k - [c^\top x^{k+1} + r^{k+1}] \leq \delta_{\texttt{Tol}}$, stop
4. Set $k = k + 1$ and go back Step 1

The algorithm stops after finitely many steps (even if $\delta_{\texttt{Tol}} = 0$) at the solution without enumerating all the vertices

Progressive Hedging

Dual Decomposition

We remind the two-stage program:

$$\begin{cases} \min_x & c^\top x + \sum_{i=1}^N p_i Q(x, \xi_i) \\ \text{s.t.} & Ax = b\,, \quad x \geq 0\,, \end{cases}$$

Reformulation:

$$\begin{cases} \min_x & \sum_{i=1}^N p_i \Big(c^\top x + Q(x, \xi_i) \Big) \\ \text{s.t.} & Ax = b\,, \quad x \geq 0\,, \end{cases}$$

or equivalently, using a splitted formulation:

$$\begin{cases} \min_{\{x_i\}_i} & \sum_{i=1}^N p_i \Big(c^\top x_i + Q(x_i, \xi_i) \Big) \\ \text{s.t.} & Ax_i = b\,, \quad x_i \geq 0\,, \\ & x_i = {\color{red} x_j} \quad \forall i = 1, \cdots, N \end{cases}$$

Dual Decomposition

We remind the two-stage program:

$$\begin{cases} \min_x & c^\top x + \sum_{i=1}^N p_i Q(x, \xi_i) \\ \text{s.t.} & Ax = b\,, \quad x \geq 0\,, \end{cases}$$

Reformulation:

$$\begin{cases} \min_x & \sum_{i=1}^N p_i \Big(c^\top x + Q(x, \xi_i) \Big) \\ \text{s.t.} & Ax = b\,, \quad x \geq 0\,, \end{cases}$$

or equivalently, using a splitted formulation:

$$\begin{cases} \min_{\{x_i\}_i} & \sum_{i=1}^N p_i \Big(c^\top x_i + Q(x_i, \xi_i) \Big) \\ \text{s.t.} & Ax_i = b\,, \quad x_i \geq 0\,, \\ & x_i = {\color{red}\sum_{j=1}^N p_j x_j} \quad \forall i = 1, \cdots, N \end{cases}$$

Dualization of the coupling constraints

By dualizing each coupling constraints with a multiplier λ_i:

$$\begin{cases} \min_{\{x_i\}_i} \max_{\lambda} & \sum_{i=1}^N p_i \Big(c^\top x_i + Q(x_i,\xi_i) + \lambda_i \big(x_i - \sum_{j=1}^N p_j x_j\big) \Big) \\ \text{s.t.} & Ax_i = b\,, \quad x_i \geq 0\,, \quad \forall i = 1,\cdots,N \end{cases}$$

Note that:

$$\begin{aligned} \sum_{i=1}^N p_i \lambda_i \big(x_i - \sum_{j=1}^N p_j x_j\big) &= \sum_{i=1}^N p_i \lambda_i x_i - \sum_{i=1}^N \sum_{j=1}^N p_i p_j \lambda_i x_j \\ &= \sum_{i=1}^N p_i \lambda_i x_i - \sum_{j=1}^N \sum_{i=1}^N \big(p_i \lambda_i\big) p_j x_j \\ &= \sum_{i=1}^N p_i \lambda_i x_i - \sum_{j=1}^N \mathbb{E}(\lambda) p_j x_j \end{aligned}$$

The problem is equivalent to

$$\begin{cases} \min_{\{x_i\}_i} \max_{\lambda} & \sum_{i=1}^N p_i \Big(c^\top x_i + Q(x_i,\xi_i) + \big(\lambda_i - \mathbb{E}(\lambda)\big) x_i \Big) \\ \text{s.t.} & Ax_i = b\,, \quad x_i \geq 0\,, \quad \forall i = 1,\cdots,N \end{cases}$$

Dual problem

The dual problem reads

$$\begin{cases} \max_{\lambda} \min_{\{x_i\}_i} & \sum_{i=1}^{N} p_i \Big(c^\top x_i + Q(x_i, \xi_i) + \big(\lambda_i - \mathbb{E}(\lambda)\big) x_i \Big) \\ \text{s.t.} & A x_i = b \,, \quad x_i \geq 0 \,, \quad \forall i = 1, \cdots, N \end{cases}$$

For λ_i given, the inner problem decomposes in N deterministic problems

$$\min_{x_i} c^\top x_i + Q(x_i, \xi_i) + \big(\lambda_i - \mathbb{E}(\lambda)\big) x_i$$
$$\text{s.t. } A x_i = b \,, \quad x_i \geq 0$$

Price of information

Any multiplier λ satisfying the KKT conditions of the two-stage problem satisfies

$$\mathbb{E}(\lambda) = 0$$

Subproblem is equivalent to

$$\min_{x_i} c^\top x_i + Q(x_i, \xi_i) + \lambda_i x_i$$
$$\text{s.t. } A x_i = b \,, \quad x_i \geq 0$$

Set an initial multiplier λ^0 such that $\mathbb{E}(\lambda^0) = 0$

1. Solve for each scenario
$$\min_{x_i} c^\top x_i + Q(x_i, \xi_i) + \lambda_i^k x_i$$
$$\text{s.t. } Ax_i = b\,, \quad x_i \geq 0$$
2. Update the first-stage variable
$$\overline{x}^{k+1} = \sum_{i=1}^{N} p_i x_i^{k+1}$$
3. Update the price of information as
$$\lambda_i^{k+1} = \lambda_i^k + \rho(x_i^{k+1} - \overline{x}^{k+1})$$

Progressive Hedging algorithm

Set an initial multiplier λ^0 such that $\mathbb{E}(\lambda^0) = 0$

1. Solve for each scenario

$$\min_{x_i} c^\top x_i + Q(x_i, \xi_i) + \lambda_i^k x_i + \rho \|x_i - \overline{x}^k\|^2$$
$$\text{s.t. } Ax_i = b\,, \quad x_i \geq 0$$

2. Update the first-stage variable

$$\overline{x}^{k+1} = \sum_{i=1}^{N} p_i x_i^{k+1}$$

3. Update the price of information as

$$\lambda_i^{k+1} = \lambda_i^k + \rho(x_i^{k+1} - \overline{x}^{k+1})$$

Convergence

Assume that for all $i = 1, \cdots, N$, there exists x_i such that $c^\top x_i + Q(x_i, \xi_i) < +\infty$ with $Ax_i = b$, $x_i \geq 0$.
Then the progressive hedging algorithm converges toward an optimal primal solution and the price of information converges toward an optimal price of information

Multistage Stochastic Linear Programming - MSLP

Multistage stochastic linear programs - T-SLP

Nested formulation

$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \mathbb{E}\left[\min_{\substack{B_2 x_1 + A_2 x_2 = b_2 \\ x_2 \geq 0}} c_2^\top x_2 + \mathbb{E}\left[\cdots + \mathbb{E}[\min_{\substack{B_T x_{T-1} + A_T x_T = b_T \\ x_T \geq 0}} c_T^\top x_T]\right]\right]$$

- Some elements of the data $\xi = (c_t, B_t, A_t, b_t)$ depend on uncertainties

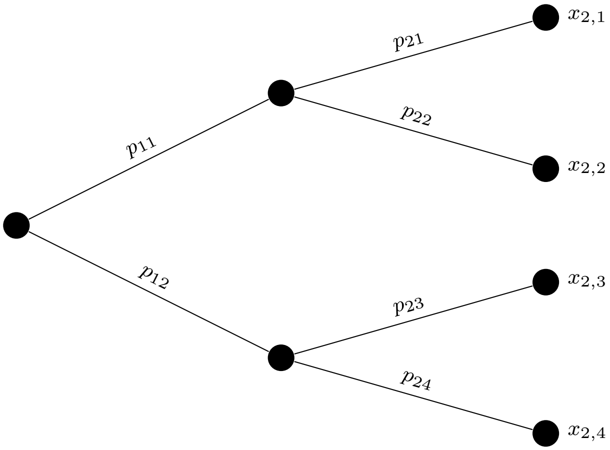
Multistage stochastic linear programs - T-SLP

Nested formulation

$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \mathbb{E}_{|\xi_1} \left[\min_{\substack{B_2 x_1 + A_2 x_2 = b_2 \\ x_2 \geq 0}} c_2^\top x_2 + \mathbb{E}_{|\xi_{[2]}} \left[\cdots + \mathbb{E}_{|\xi_{[T-1]}} \left[\min_{\substack{B_T x_{T-1} + A_T x_T = b_T \\ x_T \geq 0}} c_T^\top x_T \right] \right] \right].$$

- Some elements of the data $\xi = (c_t, B_t, A_t, b_t)$ depend on uncertainties

Scenario tree



Scenario trees

- Assume that the stochastic process $\xi = (\xi_1, \dots, \xi_T)$ has a finite number K of realizations
- Each realization (sequence) is called a scenario $\xi^i = (\xi^i_1, \dots, \xi^i_T)$
- Each scenario $\xi^i = (\xi^i_1, \dots, \xi^i_T)$ has a probability $p_i > 0$ associated
- The value of a given scenario ξ^i at stage t is denoted a node of the tree
- The set of all nodes at stage t is denoted by Ω_t
- The total number of scenarios is $K = |\Omega_T|$
- We sometimes use the short hand ι to denote a node: $\iota \in \Omega_t$
- The ancestor of a node $\iota \in \Omega_t$ is $a(\iota) \in \Omega_{t-1}$
- The set of descendants (children) of a node $\iota \in \Omega_t$ is denoted by C_ι
- $\Omega_{t+1} = \cup_{\iota \in \Omega_t} C_\iota$, and $C_\iota \cap C_{\iota'} = \emptyset$ if $\iota \neq \iota'$
- $\mathcal{S}^{(\iota)}$ is the set of all scenarios passing through node ι
- The probability of node ι is $p^{(\iota)} := \mathbb{P}[\mathcal{S}^{(\iota)}]$
- Conditional probability $\rho_{a\iota} = \frac{p^{(\iota)}}{p^{(a)}}$ if $a = a(\iota)$
- The probability of reaching a node $\iota \in \Omega_t$ is $p^{(\iota)} = \rho_{\iota_1\iota_2}\rho_{\iota_2\iota_1} \dots \rho_{\iota_{t-1}\iota_t} = \mathbb{P}[\mathcal{S}^{(\iota)}]$

Nested formulation

Linear case

$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \mathbb{E}\left[\min_{\substack{B_2 x_1 + A_2 x_2 = b_2 \\ x_2 \geq 0}} c_2^\top x_2 + \mathbb{E}\left[\cdots + \mathbb{E}[\min_{\substack{B_T x_{T-1} + A_T x_T = b_T \\ x_T \geq 0}} c_T^\top x_T]\right]\right]$$

Linear case + scenario tree

$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \sum_{\iota_2 \in \Omega_2} \rho_{1\iota_2} \left[\min_{\substack{B_2^{\iota_2} x_1 + A_2^{\iota_2} x_2 = b_2^{\iota_2} \\ x_2 \geq 0}} c_2^{\iota_2 \top} x_2 + \sum_{\iota_3 \in \Omega_3} \rho_{\iota_2 \iota_3} \left[\cdots + \right.\right.$$

$$\left.\left.\cdots + \sum_{\iota_T \in \Omega} \rho_{\iota_{T-1}\iota_T} \left[\min_{\substack{B_T^{\iota_T} x_{T-1} + A_T^{\iota_T} x_T = b_T^{\iota_T} \\ x_T \geq 0}} c_T^{\iota_T \top} x_T\right]\right]\right]$$

Equivalent deterministic

Denoting $\xi_t^{\iota i} = (c_t^{\iota i}, B_t^{\iota i}, A_t^{\iota i}, b_t^{\iota i})$ we can rewrite the above problem as

$$\begin{cases}
\begin{array}{llllllll}
\min & c_1^\top x_1 & + & \sum_{\iota_2 \in \Omega_2} p^{(\iota_2)} c_2^{\iota 2\,\top} x_2^{\iota 2} & + & \sum_{\iota_3 \in \Omega_3} p^{(\iota_3)} c_3^{\iota 3\,\top} x_3^{\iota 3} & + \cdots + & \sum_{\iota_T \in \Omega} p^{(\iota_T)} c_T^{\iota T\,\top} x_T^{\iota T} \\
\text{s.t.} & A_1 x_1 & & & & & & = b_1 \\
 & B_2^{\iota 2} x_1 & + & A_2^{\iota 2} x_2^{\iota 2} & & & & = b_2^{\iota 2} \ \ \forall \iota_2 \in \Omega_2 \\
 & & & B_3^{\iota 3} x_2^{a(\iota_3)} & + & A_3^{\iota 3} x_3^{\iota 3} & & = b_3^{\iota 3} \ \ \forall \iota_3 \in \Omega_3 \\
 & & & \ddots & & \ddots & & \\
 & & & B_T^{\iota T} x_{T-1}^{a(\iota_T)} & + & A_T^{\iota T} x_T^{\iota T} & & = b_T^{\iota T} \ \ \forall \iota_T \in \Omega_T
\end{array}
\end{cases}$$

This is a LP!

Equivalent deterministic

Denoting $\xi_t^{\iota i} = (c_t^{\iota i}, B_t^{\iota i}, A_t^{\iota i}, b_t^{\iota i})$ we can rewrite the above problem as

$$
\left\{
\begin{array}{lllllllll}
\min & c_1^\top x_1 & + & \sum_{\iota_2 \in \Omega_2} p^{(\iota_2)} c_2^{\iota_2\,\top} x_2^{\iota_2} & + & \sum_{\iota_3 \in \Omega_3} p^{(\iota_3)} c_3^{\iota_3\,\top} x_3^{\iota_3} & + \cdots + & \sum_{\iota_T \in \Omega} p^{(\iota_T)} c_T^{\iota_T\,\top} x_T^{\iota_T} & \\
\text{s.t.} & A_1 x_1 & & & & & & = b_1 & \\
 & B_2^{\iota_2} x_1 & + & A_2^{\iota_2} x_2^{\iota_2} & & & & = b_2^{\iota_2} & \forall \iota_2 \in \Omega_2 \\
 & & & B_3^{\iota_3} x_2^{a(\iota_3)} & + & A_3^{\iota_3} x_3^{\iota_3} & & = b_3^{\iota_3} & \forall \iota_3 \in \Omega_3 \\
 & & & \ddots & & \ddots & & & \\
 & & & B_T^{\iota_T} x_{T-1}^{a(\iota_T)} & + & A_T^{\iota_T} x_T^{\iota_T} & & = b_T^{\iota_T} & \forall \iota_T \in \Omega_T
\end{array}
\right.
$$

This is a LP!

Example:

- Suppose $T = 12$, each node $\iota_t \in \Omega_t$ has 4 children

This gives $4^{11} = 4,194,304$ scenarios.

- Suppose each $x_t \in \Re^{100}$, $t = 1, \ldots, 12$

The number of variables of the above problem is approximately 4.59×10^8!

Dynamic programming formulation

- Stage $t = T$

$$Q_T(x_{T-1}, \xi^{\iota}_{[T]}) := \min_{\substack{B^{\iota}_T x^{a(\iota)}_{T-1} + A^{\iota}_T x_T = b^{\iota}_T \\ x_T \geq 0}} {c^{\iota}_T}^{\top} x_T$$

Dynamic programming formulation

- Stage $t = T$

$$Q_T(x_{T-1}, \xi^\iota_{[T]}) := \min_{\substack{B^\iota_T x^{a(\iota)}_{T-1} + A^\iota_T x_T = b^\iota_T \\ x_T \geq 0}} {c^\iota_T}^\top x_T$$

- At stages $t = 2, \ldots, T-1$

$$Q_t(x_{t-1}, \xi^\iota_{[t]}) := \min_{\substack{B^\iota_t x^{a(\iota)}_{t-1} + A^\iota_t x_t = b^\iota_t \\ x_t \geq 0}} {c^\iota_t}^\top x_t + \sum_{j \in C_\iota} p^{(j)} \left[Q_{t+1}(x_t, \xi^j_{[t+1]}) \right]$$

Dynamic programming formulation

- Stage $t = T$

$$Q_T(x_{T-1}, \xi^\iota_{[T]}) := \min_{\substack{B^\iota_T x^{a(\iota)}_{T-1} + A^\iota_T x_T = b^\iota_T \\ x_T \geq 0}} {c^\iota_T}^\top x_T$$

- At stages $t = 2, \ldots, T-1$

$$Q_t(x_{t-1}, \xi^\iota_{[t]}) := \min_{\substack{B^\iota_t x^{a(\iota)}_{t-1} + A^\iota_t x_t = b^\iota_t \\ x_t \geq 0}} {c^\iota_t}^\top x_t + \sum_{j \in C_\iota} p^{(j)} \left[Q_{t+1}(x_t, \xi^j_{[t+1]}) \right]$$

- Stage $t = 1$

$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \sum_{\iota \in C_1} p^{(\iota)} \left[Q_2(x_1, \xi^\iota_2) \right]$$

Dynamic programming formulation

- Stage $t = T$

$$Q_T(x_{T-1}, \xi^{\iota}_{[T]}) := \min_{\substack{B^{\iota}_T x^{a(\iota)}_{T-1} + A^{\iota}_T x_T = b^{\iota}_T \\ x_T \geq 0}} c^{\iota\top}_T x_T$$

- At stages $t = 2, \dots, T-1$

$$\underline{Q}_t(x_{t-1}, \xi^{\iota}_{[t]}) := \min_{\substack{B^{\iota}_t x^{a(\iota)}_{t-1} + A^{\iota}_t x_t = b^{\iota}_t \\ x_t \geq 0}} c^{\iota\top}_t x_t + \sum_{j \in C_\iota} p^{(j)} \left[\underline{Q_{t+1}}(x_t, \xi^{j}_{[t+1]}) \right]$$

- Stage $t = 1$

$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \sum_{\iota \in C_1} p^{(\iota)} \left[\underline{Q_2}(x_1, \xi^{\iota}_2) \right]$$

Dynamic programming formulation

- Stage $t = T$

$$Q_T(x_{T-1}, \xi^\iota_{[T]}) := \min_{\substack{B^\iota_T x^{a(\iota)}_{T-1} + A^\iota_T x_T = b^\iota_T \\ x_T \geq 0}} {c^\iota_T}^\top x_T$$

- At stages $t = 2, \ldots, T-1$

$$\underline{Q_t}(x_{t-1}, \xi^\iota_{[t]}) := \min_{\substack{B^\iota_t x^{a(\iota)}_{t-1} + A^\iota_t x_t = b^\iota_t \\ x_t \geq 0}} {c^\iota_t}^\top x_t + \check{\mathcal{Q}}_{t+1}(x_t, \xi^\iota_{[t]})$$

$$\check{\mathcal{Q}}_{t+1}(x_t, \xi^\iota_{[t]}) := \sum_{j \in C_\iota} p^{(j)} \left[\underline{Q_{t+1}}(x_t, \xi^j_{[t+1]}) \right]$$

- Stage $t = 1$

$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \check{\mathcal{Q}}_2(x_2, \xi_{[1]})$$

Cutting-plane approximation

Assumptions

- The set of nodes Ω_t has finitely many elements
- the problem has relatively complete recourse

The last hypotheses is made only for sake of simplicity!

- Stage $t = T$

$$Q_T(x_{T-1}^k, \xi_{[T]}) := \min_{\substack{B_T x_{T-1}^k + A_T x_T = b_T \\ x_T \geq 0}} c_T{}^\top x_T$$

- At stages $t = 2, \ldots, T-1$

$$\underline{Q}_t(x_{t-1}^k, \xi_{[t]}) := \begin{cases} \min\limits_{x_t \geq 0, r_{t+1}} & c_t{}^\top x_t + r_{t+1} \\ \text{s.t.} & B_t x_{t-1}^k + A_t x_t = b_t \\ & r_{t+1} \geq \alpha_{t+1}^j + \beta_{t+1}^j x_t \quad j = 1, \ldots, k \end{cases}$$

- Stage $t = 1$

$$\underline{z}^k := \begin{cases} \min\limits_{x_1 \geq 0, r_2} & c_1^\top x_1 + r_2 \\ \text{s.t.} & A_1 x_1 = b_1 \\ & r_2 \geq \alpha_2^j + \beta_2^j x_1 \quad j = 1, \ldots, k \end{cases}$$

Cutting-plane approximation

- At stages $t = 2, \dots, T-1$

$$
Q_t(x_{t-1}^k, \xi_{[t]}) := \begin{cases} \min\limits_{x_t \geq 0, r_{t+1}} & c_t^\top x_t + r_{t+1} & & \\ \text{s.t.} & B_t x_{t-1}^k + A_t x_t = b_t & & (\pi_t) \\ & r_{t+1} \geq \alpha_{t+1}^j + \beta_{t+1}^j x_t & j = 1, \dots, k & (\rho_j) \end{cases}
$$

- Cuts $(t = T)$

$$
\alpha_T^k := \mathbb{E}_{|\xi_{T-1}}[b_T^\top \pi_T^k] \quad \text{and} \quad \beta_T^k := -\mathbb{E}_{|\xi_{T-1}}[B_T^\top \pi_T^k]
$$

- Cuts $(t = T-1, \dots, 2)$

$$
\alpha_t^k := \mathbb{E}_{|\xi_{t-1}}[b_t^\top \pi_t^k + \sum_{j=1}^k \alpha_{t+1}^j \rho_k^j] \quad \text{and} \quad \beta_t^k := -\mathbb{E}_{|\xi_{t-1}}[B_t^\top \pi_t^k]
$$

Nested decomposition - (nested L-shaped method)

- J.R. Birge (1985)

It has two main steps:

- Forward that goes from $t = 1$ up to $t = T$ solving subproblems to define policy $x_t^k(\xi_t)$
  - In this step an upper bound $\overline{z}^k$ for the optimal value is determined
- Backward that comes from $t = T$ up to $t = 1$ solving subproblems to compute linearizations that improve the cutting-plane approximation.
  - In this step a lower bound $\underline{z}^k$ is obtained

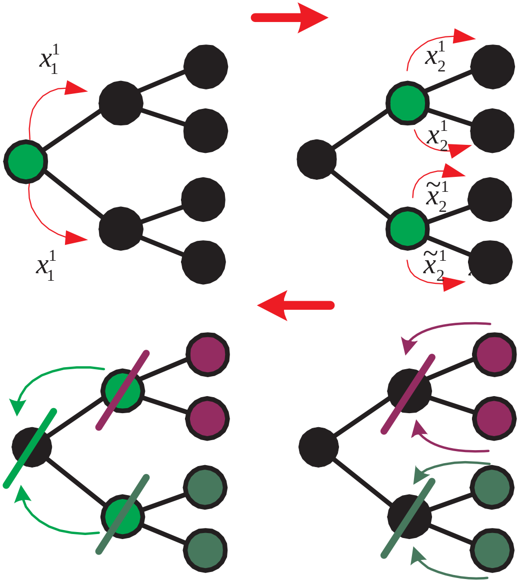
Stopping test

- The Nested decomposition stops when

$$\overline{z}^k - \underline{z}^k \leq \texttt{Tol}.$$

- In this case x_1^k is a Tol-solution to the T-SLP

Forward and Backward steps



Algorithm - Nested decomposition

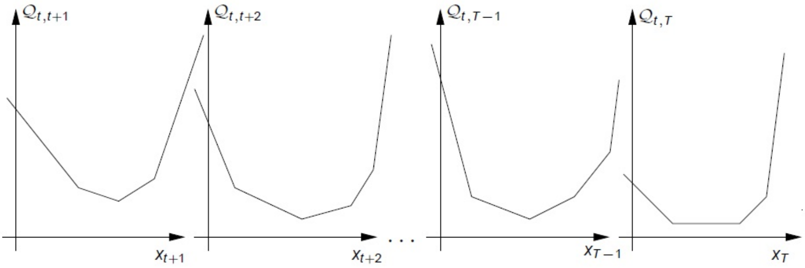
stages $t = 2, \ldots, T-1$

$$\underline{Q_t}(x_{t-1}^k, \xi_{[t]}) := \begin{cases} \min\limits_{x_t \geq 0, r_{t+1}} & c_t{}^\top x_t + r_{t+1} & & \\ \text{s.t.} & B_t x_{t-1}^k + A_t x_t = b_t & & (\pi_t) \\ & r_{t+1} \geq \alpha_{t+1}^j + \beta_{t+1}^j x_t & j = 1, \ldots, k & (\rho_j) \end{cases}$$

- **Step 0: inicialization**. Define $k = 1$ and add the constraint $r_t = 0$ in all LPs $\underline{Q_t}$, $t = 2, \ldots, T-1$ Compute $\underline{z}^1$ and let its solution be x_1^1
- **Step 1: forward**. For t=2,...,T, solve the LP $\underline{Q_t}$ to obtain $x_t^k := x_t^k(\xi_{[t]})$. Define $\bar{z}^k := \mathbb{E}[\sum_{t=1}^T c_t^\top x_t^k]$
- **Step 2: backward**. Compute α_T^k and β_T^k. Set $t = T$. Loop:
  - While $t > 2$
  - $t \leftarrow t - 1$
  - solve the LP $\underline{Q_t}(x_{t-1}^k, \xi_{[t]})$
  - Compute α_t^k and β_t^k

  Compute $\underline{z}^k$ and let its solution be x_1^{k+1}
- **Step 3: Stopping test**. If $\bar{z}^k - \underline{z}^k \leq \epsilon$, stop. Otherwise set $k \leftarrow k+1$ and go back to Step 1

Nested decomposition - iterative process

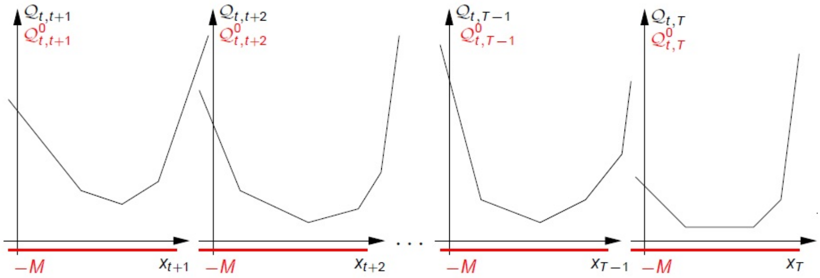


- $\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \mathcal{Q}_2(x_2, \xi_{[1]})$

- $\mathcal{Q}_{t+1}(x_t, \xi_{[t]}) = \mathbb{E}_{|\xi_{[t]}}[Q_{t+1}(x_t, \xi_{[t+1]})]$ for $t = 1, \ldots, T-1$, and $\mathcal{Q}_{T+1}(x_T, \xi_{[T]}) = 0$

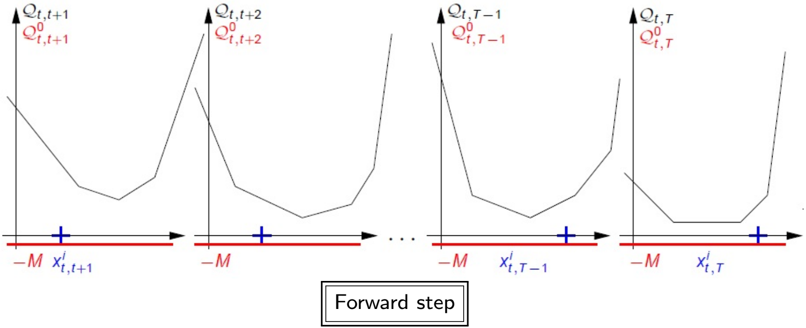
- $Q_t(x_{t-1}, \xi_{[t]}) = \min\limits_{x_t \geq 0} \; c_t^\top x_t + \mathcal{Q}_{t+1}(x_t, \xi_{[t]})$ s.t. $B_t x_{t-1} + A_t x_t = b_t$

Nested decomposition - iterative process

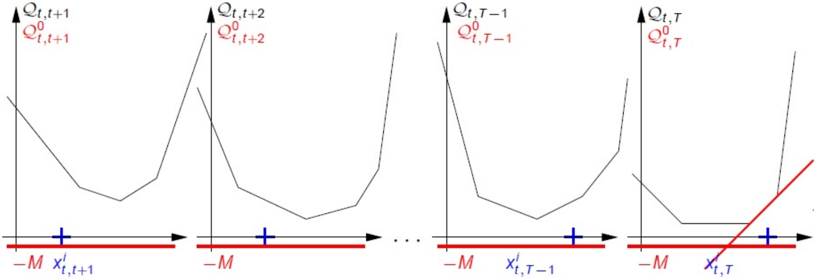


- $\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \check{\mathcal{Q}}_2(x_2, \xi_{[1]})$
- $\check{\mathcal{Q}}_{t+1}(x_t, \xi_{[t]}) = \mathbb{E}_{|\xi_{[t]}}[\underline{Q_{t+1}}(x_t, \xi_{[t+1]})]$ for $t = 1, \ldots, T-1$, e $\check{\mathcal{Q}}_{T+1}(x_T, \xi_{[T]}) = 0$
- $\underline{Q_t}(x_{t-1}, \xi_{[t]}) = \min\ c_t^\top x_t + \mathcal{Q}_{t+1}(x_t, \xi_{[t]})$ s.t. $B_t x_{t-1} + A_t x_t = b_t$, $x_t \geq 0$
- Figures by Vincent Guigues

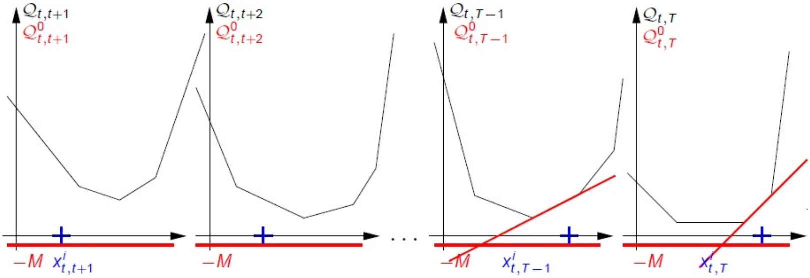
Nested decomposition - iterative process



Nested decomposition - iterative process

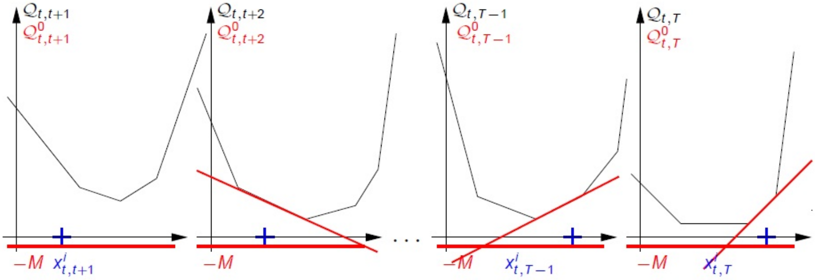


Backward step



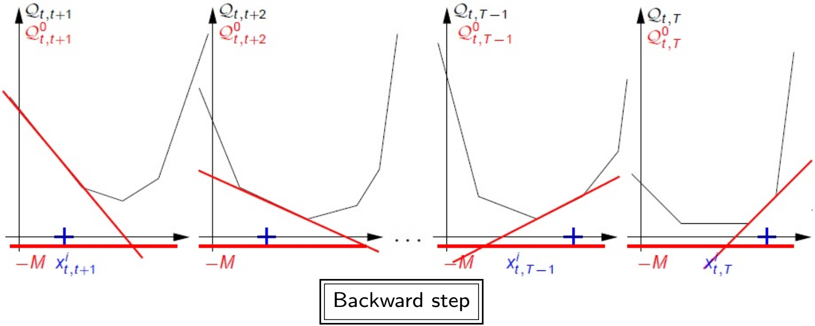
Backward step

Nested decomposition - iterative process

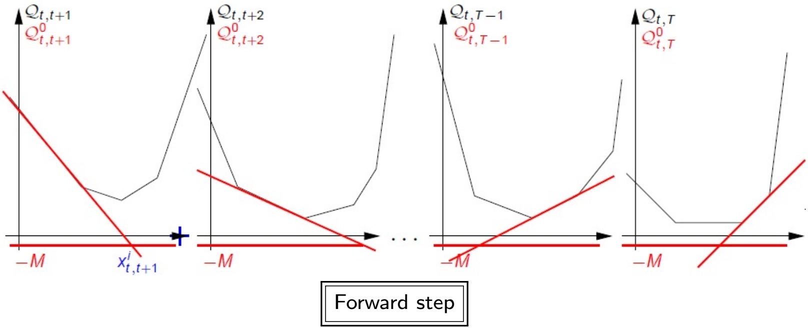


Backward step

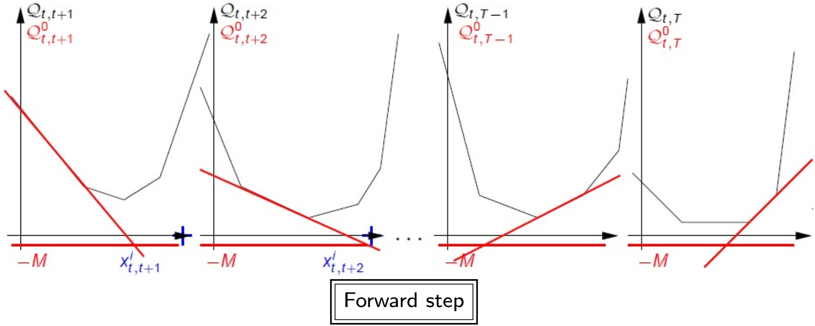
Nested decomposition - iterative process



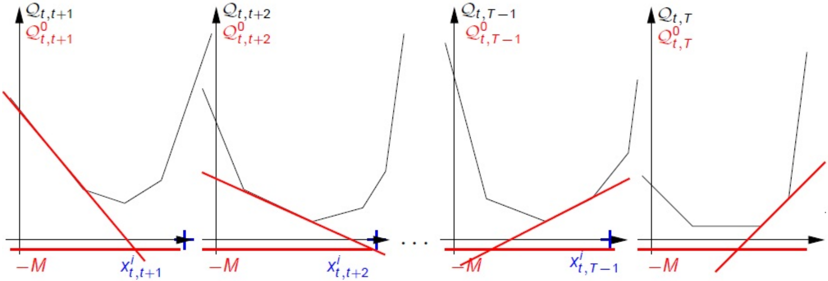
Nested decomposition - iterative process



Nested decomposition - iterative process

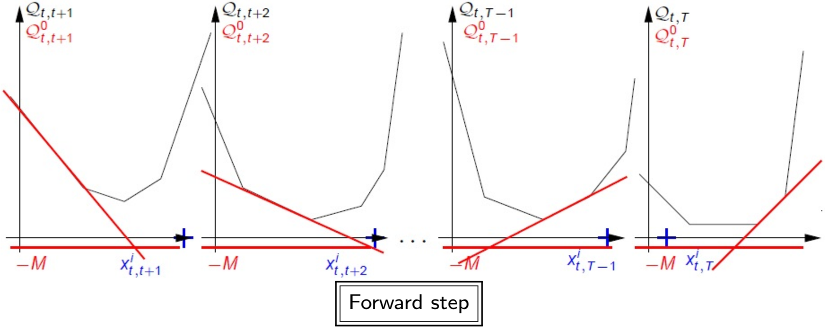


Nested decomposition - iterative process

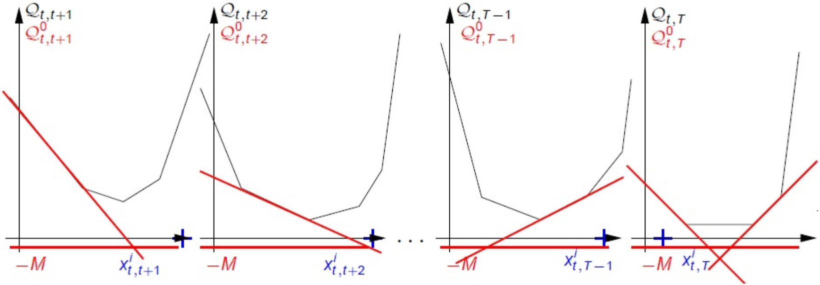


Forward step

Nested decomposition - iterative process

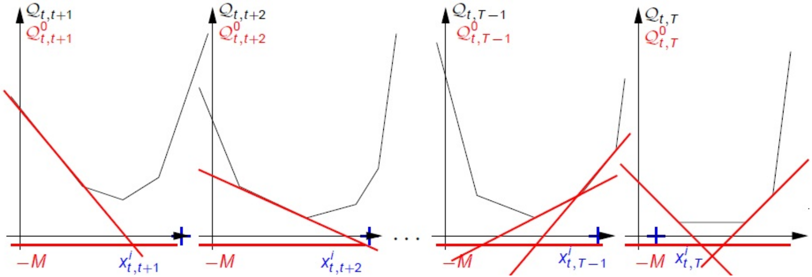


Nested decomposition - iterative process



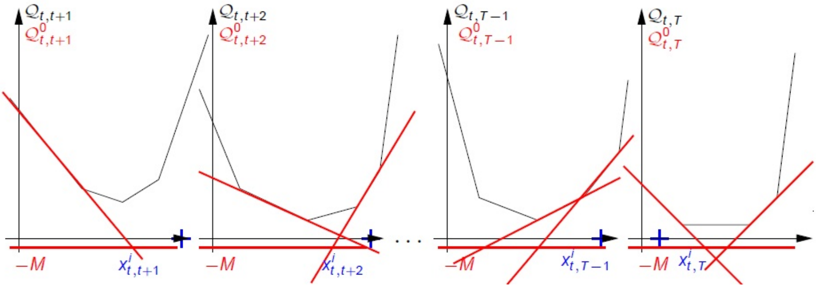
Backward step

Nested decomposition - iterative process



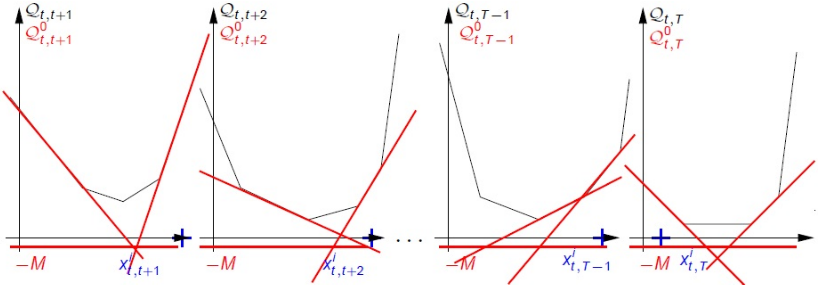
Backward step

Nested decomposition - iterative process



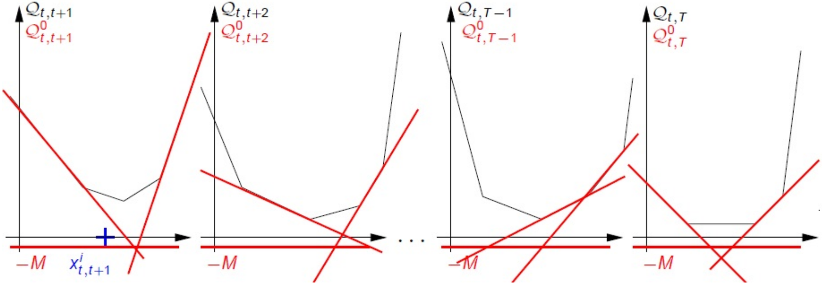
Backward step

Nested decomposition - iterative process

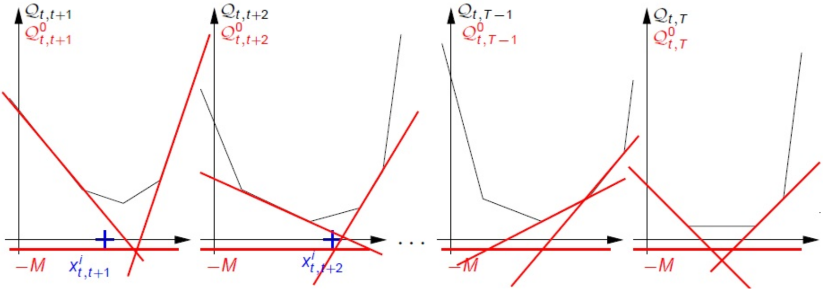


Backward step

Nested decomposition - iterative process

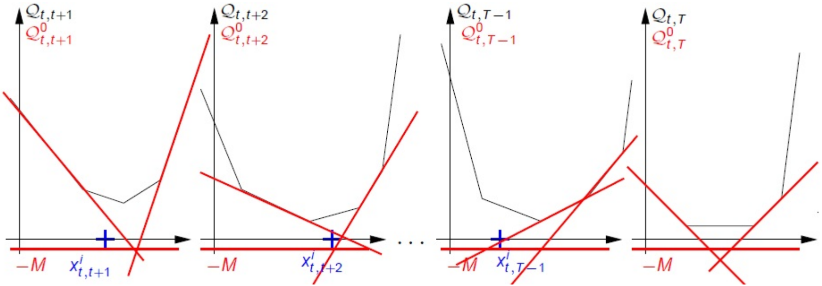


Forward step



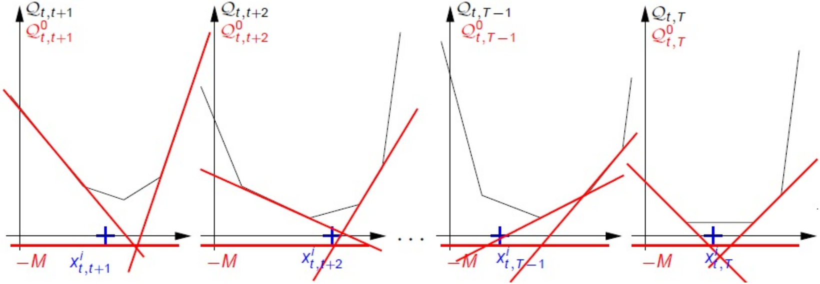
Forward step

Nested decomposition - iterative process



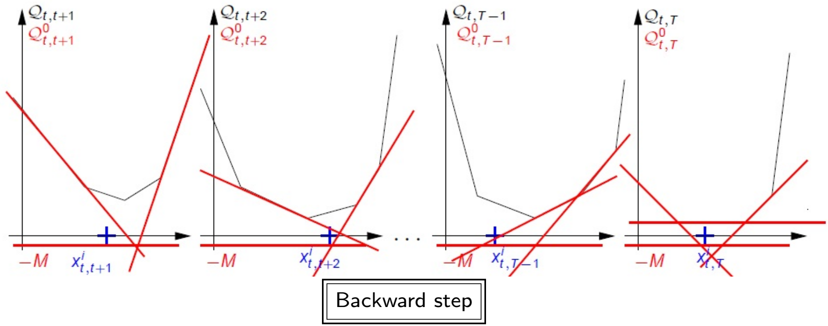
$\boxed{\text{Forward step}}$

Nested decomposition - iterative process

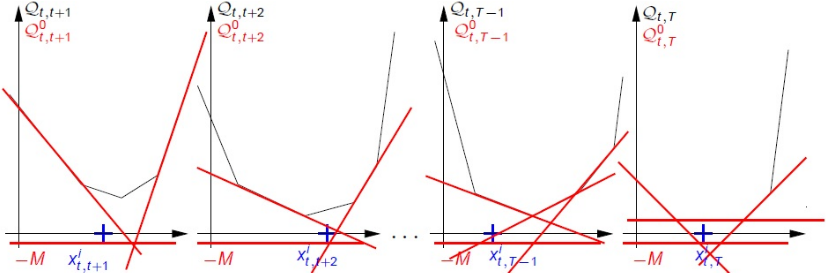


$\boxed{\text{Forward step}}$

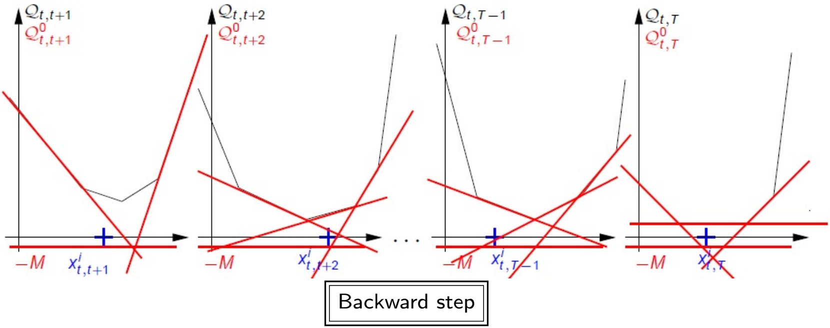
Nested decomposition - iterative process



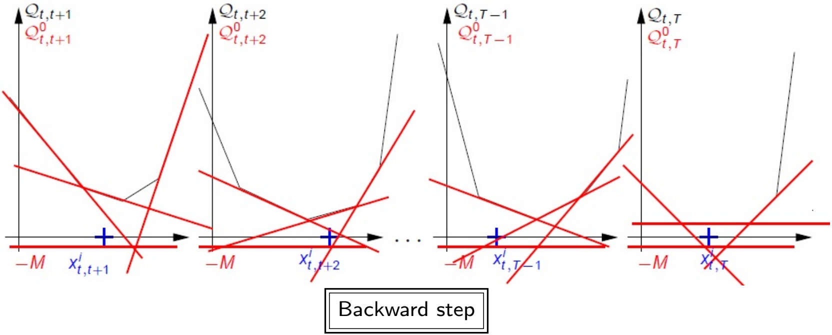
Nested decomposition - iterative process



Backward step


$\mathcal{Q}_{t,t+1}$
$\mathcal{Q}^0_{t,t+1}$
$-M$
$x^i_{t,t+1}$
$\mathcal{Q}_{t,t+2}$
$\mathcal{Q}^0_{t,t+2}$
$-M$
$x^i_{t,t+2}$
$\mathcal{Q}_{t,T-1}$
$\mathcal{Q}^0_{t,T-1}$
$-M$
$x^i_{t,T-1}$
$\mathcal{Q}_{t,T}$
$\mathcal{Q}^0_{t,T}$
$-M$
$x^i_{t,T}$
Backward step

Nested decomposition - iterative process



- Figures by Vincent Guigues.

Convergence analysis

Assumptions

- The set of nodes Ω_t has finitely many elements, $t = 1, \dots, T$
- the problem has recourse relatively complete (for simplicity, only)
- the feasible set, in each stage $t = 1, \dots, T$, is compact

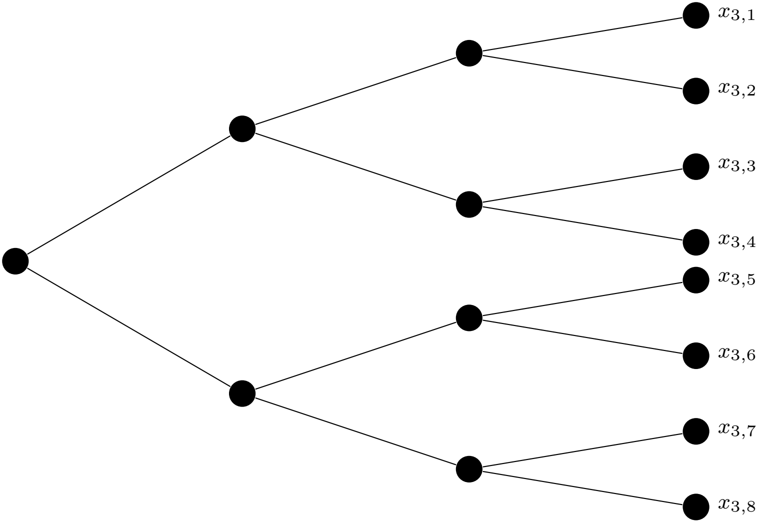
Lemma

$\check{\mathcal{Q}}_t^k(x_{t-1}, \xi_{[t-1]}) \leq \mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}) \quad \forall\, x_{t-1}$ *and* $\forall t = 2, \dots, T$

Theorem

The Nested Decomposition converges finitely to an optimal solution of the considered T-SLP

Scenario tree



Question

Can we compress the information?

Information structure

In multistage problems, decisions are sequential in nature

$$x_0 \rightsquigarrow \xi_1 \rightsquigarrow x_1 \rightsquigarrow \xi_2 \rightsquigarrow \cdots \rightsquigarrow x_T$$

The sequence $\{\xi_t\}$ is a *stochastic process*.

Non-anticipativity

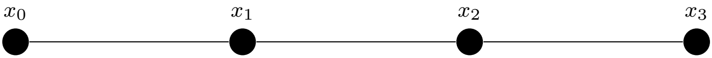
We note $\xi_{[t]} := \{\xi_1, \cdots, \xi_{t-1}\}$ the information available up to time t.
The process $\{x_t\}_t$ is *non-anticipative* if for all t, the values of x_t depends only on the past information:

$$x_t = x_t(\xi_{[t]}) \; .$$

Information structure #1: Stage-wise independence

Stage-wise independence

Assume uncertainties $\xi_1, \cdots, \xi_T$ are time-step independent:
for all t, the random variable ξ_t is stochastically independent from $\xi_{[t-1]}$.



The tree reduces to a line:
we will prove that x_4 depends only on the values in x_3, not on x_2, x_1

Proposition

Under stage-wise independence, the value functions $\{Q_t(\cdot,\xi)\}$ satisfy the Dynamic Programming backward recursions: setting $V_{T+1}(x_T) := 0$, we have for $t = T, \cdots, 1$,

$$\begin{cases} Q_t(x_{t-1},\xi) = \min_{x_t \geq 0} \left[c_t(\xi)^\top x_t + V_{t+1}(x_t)\right] \quad \text{s.t.} \quad B_t(\xi)x_{t-1} + A_t(\xi)x_t = b_t(\xi) \\ V_t(x_{t-1}) = \mathbb{E}_\xi\left[Q_t(x_{t-1},\xi)\right] \end{cases}$$

Theorem

A feasible policy $\{\overline{x}_t\}_t$ is *optimal* if and only if, for all $t = 1, \cdots, T$,

$$\overline{x}_t(\xi_{[t]}) \in \arg\min_{x_t \geq 0} \; c_t(\xi_t)^\top x_t + V_{t+1}(x_t) \quad \text{s.t.} \quad B_t(\xi_t)x_{t-1} + A_t(\xi_t)x_t = b_t(\xi_t)$$

A direct corollary is that $\overline{x}_t(\xi_{[t]})$ depends only on the values of x_{t-1} and of ξ_t.

Optimality conditions: sensitivity propagate backward

- Let the extended real-valued function $f_t(x_t, \xi) = \begin{cases} c_t(\xi)^\top x_t & \text{if} \quad x_t \geq 0 \\ +\infty & \text{otherwise} \end{cases}$
- Let the Lagrangian

$$L(x_t, \pi_t) := f_t(x_t, \xi) + V_{t+1}(x_t) + \pi_t(b_t - B_t x_{t-1} - A_t x_t)$$

Theorem

Under some assumptions (e.g. finitely many scenarios and polyhedral f_t). A feasible policy $\bar{x}_t(\xi_{[t]})$ is optimal iff there exists measurable $\bar{\pi}_t(\xi_{[t]})$, $t = 1, \dots, T$, such that

$$0 \in \partial f_t(\bar{x}_t(\xi_{[t]}), \xi_t) + \partial V_{t+1}(\bar{x}_t(\xi_{[t]})) - A_t^\top \bar{\pi}_t(\xi_{[t]})$$

and

$$\partial V_t(x_{t-1}) = \mathbb{E}_\xi \big[- B_t^\top \bar{\pi}_t(\xi_{[t]}) \big]$$

for a.e. $\xi_{[t]}$ and $t = 1, \dots, T$

Solution algorithm by discretization

Algorithm 1: Stochastic Dynamic Programming (SDP)

Data: Set $V_{T+1} = 0$
Find a discretization $\mathbb{X}_d$ of feasible space $\mathcal{X}$;
for $t = T, \cdots, 1$ **do**
  for $x_{t-1} \in \mathbb{X}_d$ **do**
    for $\xi_t \in \{\xi_t^1, \cdots, \xi_t^N\}$ **do**
      Solve $Q_t(x_{t-1}, \xi_t)$ using linear programming;
    end
    Set $V_t(x_{t-1}) = \sum_{i=1}^N p_i Q_t(x_{t-1}, \xi_t^i)$;
  end
end

Complexity

Algorithm SDP solves a total number of (small) LP:

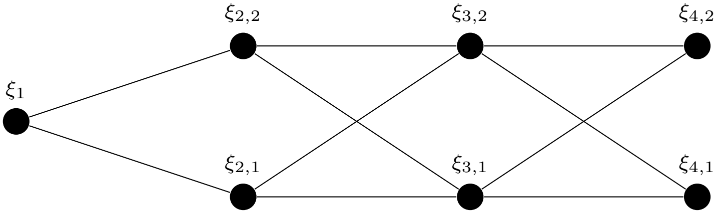
$$\mathcal{O}(|\mathbb{X}_d| \times N \times T)$$

Algorithm is useful only if dimension of the state is small (*curse of dimensionality*)

Information structure #2: Markov chain

Markov-chain lattice

The process $\{\xi_t\}_t$ is Markovian if for all t, the conditional distribution of ξ_t given $\xi_{[t-1]}$ depends only on ξ_{t-1}.



SP with recourse

Sample Average Approximation - SAA

Stochastic programs with recourse

Let's consider SP of the form

$$\min_{x \in X} \mathbb{E}[f(x, \omega)]$$

with

- $f : \Re^n \times \Omega \to \Re$ is convex on x (decision variable)
- $X \subset \Re^n$ is a deterministic set (e.g. a fixed polyhedron)
- ω is a random vector and $(\Omega, \mathcal{F}, P)$ is its probability space
- $\mathbb{E}[\cdot]$ is the expected value w.r.t. the probability measure P

For instance, in a two-stage stochastic linear framework, we have

$$f(x, \omega) = c^\top x + Q(x, \omega)$$

with

$$Q(x, \omega) := \begin{cases} \min & q(\omega)^\top y \\ \text{s.t.} & W(\omega) y = h(\omega) - T(\omega) x \\ & y \geq 0 \end{cases}$$

Representation of the uncertainties

Continuous probability distribution

- Sample space Ω contains infinitely many elements

$$\min_{x \in X} \mathbb{E}[f(x,\omega)]$$

- For computational reasons, it is necessary to consider finitely many scenarios $\omega^i \in \Omega$, with associated probability $p_i > 0$

- Resulting problem

$$\min_{x \in X} f^N(x) \quad \text{with} \quad f^N(x) := \sum_{i=1}^{N} p_i f(x,\omega^i)$$

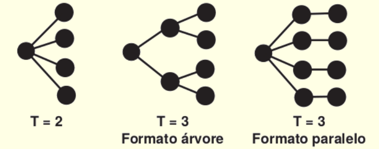
Sample Average Approximation - SAA

$$\min_{x \in X} \frac{1}{N} \sum_{i=1}^{N} f(x,\omega^i)$$

How to proceed when we do not know P?

- In many applications the probability distribution is not precisely known
- In these cases, P is estimated by using the historical of the stochastic vector
- Scenario generation can be done via Monte Carlo simulation (it is advisable not to use the historical as scenarios)

Representation of the uncertainties



- A newsvendor buys newspaper by the morning at price c and sells them along the day at price r
- Unsold newspaper are sent to be recycled. The value earned by every recycled newspaper is s
- The newsvendor wishes to maximize its expected income:

$$\min_{x \geq 0} \mathbb{E}[f(x, \omega)],$$

where

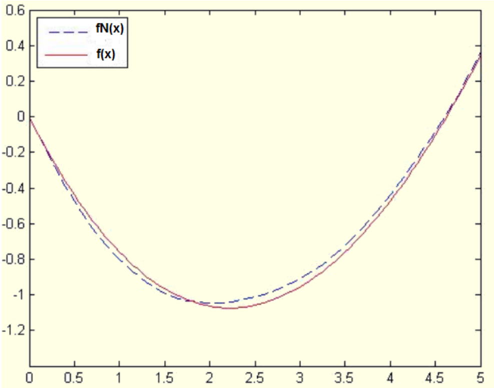
$$\begin{aligned} f(x, \omega) &= -[-cx + r \min\{x, \omega\} + s(x - \min\{x, \omega\})] \\ &= -[(s - c)x + (r - s) \min\{x, \omega\}] \end{aligned}$$

Sample Average Approximation - SAA

- The main idea of the Sample Average Approximation - SAA - approach is to use the same sample for all $x \in X$
- I.e., we draw a sample $\{\omega^1, \dots, \omega^N\}$ and approximate $f(x)$ by

$$f^N(x) = \frac{1}{N} \sum_{i=1}^{N} f(x, \omega^i)$$

regardless the given point x



Sample Average Approximation - SAA

- The approximation of f^N of f is quite close to f
- This suggests replacing the original problem $\min_{x \in X} f(x)$ by

$$\min_{x \in X} f^N(x)$$

  which can be solved by deterministic methods (L-Shaped, Nested Decomposition, Bundle Method, etc.)

Questions

- Does the SAA approach always work regardless the function $f(x, \omega)$?
- What is a good size N of the sample to be considered?
- What can we say about the quality of the SAA solution?

Asymptotic properties

Let

- $\hat{x}^N$ be a solution of the SAA problem
- $\hat{S}^N$ be the solution set of the SAA problem
- $\hat{f}^N$ be the optimal value of the SAA problem

and

- x^* be a solution of the true problem
- S^* be the solution set of the true problem
- f^* be the optimal value of the true problem

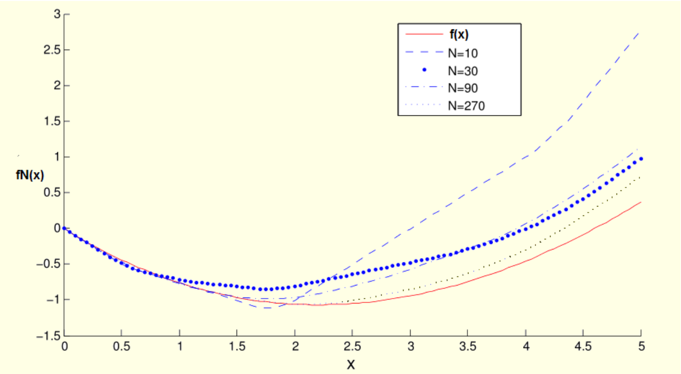
Questions

- $\lim_{N\to\infty} \hat{x}^N = x^*$?
- $\lim_{N\to\infty} dist(S^N, S^*) = 0$?
- $\lim_{N\to\infty} \hat{f}^N = f^*$?

Asymptotic properties

- Firstly, let's try to answer these questions by considering the newsvendor problem
- Suppose the demand for newspaper follows an Exponential probability distribution

$$\omega \sim \texttt{Exponencial}(10), \quad \mathbb{P}[\omega \leq x] = 1 - e^{-10x} \quad \text{(if } x \geq 0)$$



- The table presents the values of $\hat{x}^N$ and $\hat{f}^N$ for some different sample size N

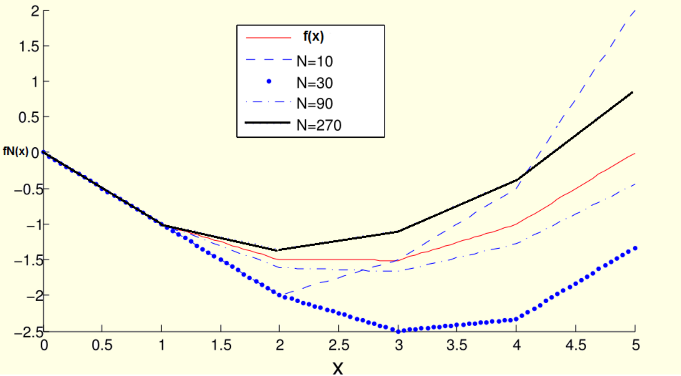
N	10	30	90	270	∞
$\hat{x}^N$	1.46	1.44	1.54	2.02	2.23
$\hat{f}^N$	-1.11	-0.84	-0.98	-1.06	-1.07

- It seems that f^N approximates well f^* when N increases

- Notice that $\hat{x}^N \to x^*$ and $\hat{f}^N \to f^*$

- Suppose now that demand for newspaper follows a discrete uniform probability distribution on the set

$$\{1, 3, \ldots, 10\}$$



- The table presents the values of $\hat{x}^N$ and $\hat{f}^N$ for some different sample size N

N	10	30	90	270	∞
$\hat{x}^N$	2	3	3	2	[2,3]
$\hat{f}^N$	-2.00	-2.50	-1.67	-1.35	-1.50

- Again, it seems that f^N approximates well f^* when N increases
- Notice that $\hat{f}^N \to f^*$, but $\hat{x}^N$ does not seem to converge
- $\hat{x}^N$ is oscillating between two optimal solutions of the problem
- What can we conclude?

Convergence results

- In both cases (continuous and discrete) the function f^N converges uniformly to f
- Uniform convergence occurs, for instance, when f is continuous
- When continuous convergence is observed, we have the following results

Theorem

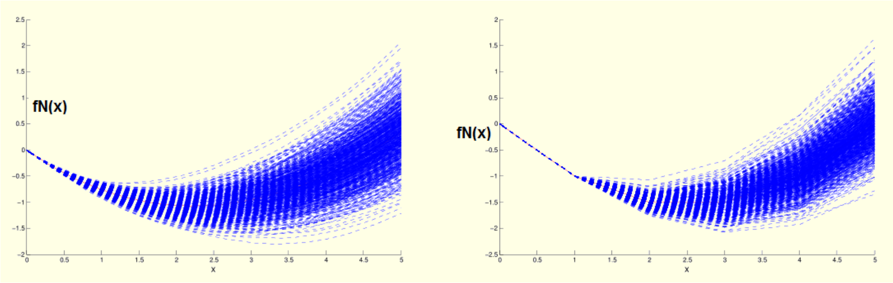
- $\lim_{N\to\infty} \hat{f}^N = f^*$ *with probability 1 (w.p.1)*
- *Suppose there exists a compact set* C *such that*
  - $\emptyset \neq S^* \subset C$ *and* $\emptyset \neq S^N \subset C$ *(w.p.1) for* N *large enough*
  - *the objective function is continuous and finite-valued on* C

Then $\lim_{N\to\infty} dist(S^N, S^*) = 0$ *w.p.1*

What does "convergence w.p.1" mean?

- Each function f^N is constructed with a single sample
- Regardless the sample, convergence results hold provided that $N \to \infty$

Let's repeat the same experiment for $N = 270$ several times:



(a) Exponential distribution (b) Discrete uniform distribution

What does "convergence w.p.1" mean?

- For some samples with $N = 270$ the approximation is quite good. However, for other samples the approximation is poor
- Why don't we have convergence for every sample?

What does "convergence w.p.1" mean?

- For some samples with $N = 270$ the approximation is quite good. However, for other samples the approximation is poor
- Why don't we have convergence for every sample?
- The theorem only ensures convergence when $N \to \infty$...

Given a sample of size of N, we solve the SAA problem $\min_{x \in X} f^N(x)$

How do we know if we have a "good" or "bad " sample?

What does "convergence w.p.1" mean?

- For some samples with $N = 270$ the approximation is quite good. However, for other samples the approximation is poor
- Why don't we have convergence for every sample?
- The theorem only ensures convergence when $N \to \infty$...

Given a sample of size of N, we solve the SAA problem $\min_{x \in X} f^N(x)$

How do we know if we have a "good" or "bad " sample?

- The answer is: we do not know for sure. However, we may employ simulation and statistical tools to access quality

A result on the sample size

$$\min_{x\in X} f(x) = \mathbb{E}[f(x,\omega)] \qquad (SAA) \quad \min_{x\in X} f^N(x) = \frac{\sum_{i=1}^N f(x,\omega^i)}{N}$$

Theorem

Suppose the true stochastic program has a unique solution x^*, X is a compact set and $f(\cdot,\omega)$ is strongly convex. Then, for all $\epsilon > 0$ there exist constants $\beta(\epsilon) > 0$ and $C > 0$ such that

$$\mathbb{P}[\|\hat{x}^N - x^*\| > \epsilon] \leq Ce^{-N\,\beta(\epsilon)}$$

A result on the sample size

$$\min_{x\in X} f(x) = \mathbb{E}[f(x,\omega)] \qquad (SAA) \quad \min_{x\in X} f^N(x) = \frac{\sum_{i=1}^N f(x,\omega^i)}{N}$$

Theorem

Suppose the true stochastic program has a unique solution x^*, X is a compact set and $f(\cdot,\omega)$ is strongly convex. Then, for all $\epsilon > 0$ there exist constants $\beta(\epsilon) > 0$ and $C > 0$ such that

$$\mathbb{P}[\|\hat{x}^N - x^*\| > \epsilon] \leq Ce^{-N\,\beta(\epsilon)}$$

The theorem ensures the existence of such constants, but not their values

We need to perform simulation...

Simulation

- Given a sample $\{\omega^1, \ldots, \omega^N\}$, define

$$\hat{x}^N \in \arg\min_{x \in X} f^N(x), \quad \text{and} \quad f^N(x) = \frac{\sum_{i=1}^N f(x, \omega^i)}{N} \quad (SAA)$$

- In order to assess the quality of $\hat{x}^N$ it is mandatory to generate a larger sample $\{\tilde{\omega}^1, \ldots, \tilde{\omega}^{N'}\}$ independent of $\{\omega^1, \ldots, \omega^N\}$ and evaluate the costs

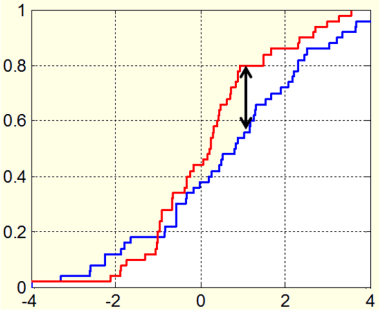
$$f(\hat{x}^N, \tilde{\omega}^j), \quad j = 1, \ldots N' \;\; (>> N)$$

(Evaluating the function is easier than solving the SAA problem)

Simulation

$$\hat{x}^N \in \arg\min_{x \in X} f^N(x), \qquad f^N(x) = \frac{\sum_{i=1}^N f(x, \omega^i)}{N} \quad (SAA)$$

- In order to infer if the sample size N is satisfactory we may compare
  - $f^N(\hat{x}^N)$ with $f^{N'}(\hat{x}^N)$ (average of the individual costs $f(\hat{x}^N, \tilde{\omega}^j)$)
  - Empirical distribution of the individual costs $f(\hat{x}^N, \tilde{\omega}^j), \quad j = 1, \ldots, N'$ and $f(\hat{x}^N, \omega^i), \quad i = 1, \ldots, N$ (KS-test)



Another idea widely used in practice is:

- Given N and M, generate M samples of size N and solve M problems SAA
- Compare the most important variables of the M SAA solutions $\hat{x}_i^N$, $i = 1, \ldots, M$
- Evaluate the SAA solutions using a larger sample $\{\tilde{\omega}^1, \ldots, \tilde{\omega}^{N'}\}$
- Compare the empirical cost distributions

If there is a certain "adherence" among the results, the size N can be considered satisfactory. Otherwise, it is suggested to increase N

Importance of simulation

- It allows us to analyze the quality of solution obtained with the stochastic model
- it allows us to estimate an appropriated sample size

Computing confidence intervals

Let x be fixed (for instance, $x = x^N$ the solution of the SAA model)

The strong law of large numbers ensures, for N large enough

$$f^N(x) := \frac{1}{N}\sum_{i=1}^{N} f(x, \omega^i) \approx \int_\Omega f(x,\omega)dP(\omega) = \mathbb{E}[f(x,\omega)] := f(x)$$

w.p.1, provided that $\{\omega^1, \dots, \omega^N\}$ is a iid sample of ω and $\mathbb{E}[\omega]$ is finite (no assumption on the distribution of ω is required!)

Thus, we can use $f^N(x)$ as an approximation of

$$f(x) = \mathbb{E}[f(x,\omega)]$$

Difficulties

- $f^N(x)$ is a random variable itself: it depends on the sample $\{\omega^1, \dots, \omega^N\}$
- Sometimes f^N can be an accurate approximation of f, sometimes not

The Central Limit Theorem

As f^N is a random variable, it makes sense to compute its mean and variance. Let x be a given point:

$$\mathbb{E}[f^N(x)] = \mathbb{E}[\frac{1}{N}\sum_{i=1}^{N} f(x,\omega^i)] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[f(x,\omega^i)] = f(x)$$

Therefore, $f^N(x)$ is an unbiased estimator of $f(x)$

$$\mathbb{V}\mathtt{ar}[f^N(x)] \quad = \mathbb{V}\mathtt{ar}[\tfrac{1}{N}\sum_{i=1}^{N} f(x,\omega^i)] = \tfrac{1}{N^2}\sum_{i=1}^{N}\mathbb{V}\mathtt{ar}[f(x,\omega^i)] = \tfrac{1}{N}\mathbb{V}\mathtt{ar}[f(x,\omega)]$$

Thus, the variance of $f^N(x)$ vanishes when N goes to infinity (if $\sigma^2 = \mathbb{V}\mathtt{ar}[f(x,\omega)]$ is finite, of course)

The Central Limit Theorem - CLT

$$\frac{\sqrt{N}[f^N(x) - f(x)]}{\sigma} \approx \mathcal{N}(0,1)$$

The Central Limit Theorem

$$\frac{\sqrt{N}[f^N(x) - f(x)]}{\sigma} \approx \mathcal{N}(0,1)$$

The CLT ensures that, for an arbitrary $x \in \texttt{dom}\,(f)$, $\sqrt{N}[f^N(x) - f(x)]$ converges in distribution to normal distribution with zero mean and variance equal to $\sigma^2 = \mathbb{V}\mathtt{ar}[f(x,\omega)]$ (provided that this variance is finite)

Consequently $f^N(x)$ converges to $f(x)$ at a (stochastic) rate of $\mathcal{O}_1(N^{-1/2})$

In other words, in order to estimate $f(x)$ by its sample average with an accuracy $\epsilon > 0$ one needs a sample of size $N = \mathcal{O}_2(\epsilon^{-2})$

Although various techniques, e.g., variance reduction techniques and quasi-Monte Carlo methods, were developed in simulation literature in order to enhance the accuracy of such estimates, it is basically impossible to evaluate multivariate integrals with a high precision

As a conclusion, when solving a SAA problem, keep in mind that you're solving only an (possibly rough) approximation of the "true" stochastic program

However, there are manners to estimate how good or how bad is the SAA solution...

The CLT ensures that

$$P\left[f^N(x) - 1.96\frac{\sigma}{\sqrt{N}} \leq f(x) \leq f^N(x) + 1.96\frac{\sigma}{\sqrt{N}}\right] = 0.95$$

In practical terms, the above expression means that for each 100 samples $\{\omega^1, \ldots, \omega^N\}$, the true value $f(x)$ is contained in 95 intervals

$$\left[f^N(x) - 1.96\frac{\sigma}{\sqrt{N}},\ f^N(x) + 1.96\frac{\sigma}{\sqrt{N}}\right]$$

The variance σ^2 is generally unknown, but it can be estimated by

$$S^2 = \frac{\sum_{i=1}^{N}[f(x,\omega^i) - f^N(x)]^2}{N-1}$$

The true stochastic optimization problem can be solved by the SAA approach in a reasonable time with a reasonable accuracy provided that the following conditions are satisfied:

- (i) the required sample size is manageable
- (ii) it is possible to solve the constructed SAA problem with a reasonable efficiency

From this point of view the number of scenarios of the true problem (i.e., cardinality of the support Ω of the distribution of ω) is irrelevant and can be infinite

Condition (ii) holds in the case of two stage linear stochastic programming with recourse

Some conclusions

- Stochastic optimization decisions are in general balanced: protect against "bad" scenarios
- The complexity of the optimization problem grows with the number of scenarios
- A SAA problem is frequently used to approximate a stochastic program
- Once a SAA solution is determined, it is crucial to evaluate the quality of the output through simulations

Estimating optimality gap

Let's consider SP of the form

$$\min_{x \in X} \mathbb{E}[f(x, \omega)]$$

where

- $f : \Re^n \times \Omega \to \Re$ is convex on x (decision variable)
- $X \subset \Re^n$ is a deterministic set (e.g. a fixed polyhedron)
- ω is a random vector and $(\Omega, \mathcal{F}, P)$ is its probability space
- $\mathbb{E}[\cdot]$ is the expected value w.r.t. the probability measure P

Continuous probability distribution

- Sample space Ω contains infinitely many elements

$$\min_{x \in X} \mathbb{E}[f(x,\omega)]$$

- For computational reasons, it is necessary to consider finitely many scenarios $\omega^i \in \Omega$, with associated probability $p_i > 0$

- Resulting problem

$$\min_{x \in X} f^N(x) \quad \text{with} \quad f^N(x) := \sum_{i=1}^{N} p_i f(x,\omega^i)$$

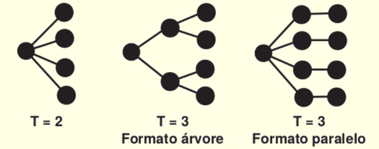
Sample Average Approximation - SAA

$$\min_{x \in X} \frac{1}{N} \sum_{i=1}^{N} f(x,\omega^i)$$

How to proceed when we do not know P?

- In many applications the probability distribution is not precisely known
- In these cases, P is estimated by using the historical of the stochastic vector
- Scenario generation can be done via Monte Carlo simulation (it is advisable not to use the historical as scenarios)

Representation of the uncertainties



Evaluating a candidate solution

The point $\hat{x}^N$ solution of

$$(SAA) \quad \hat{f}^N = \min_{x \in X} f^N(x) \quad \text{with} \quad f^N(x) = \frac{\sum_{i=1}^N f(x, \omega^i)}{N}$$

is a candidate solution to the "true" problem

$$f^* = \min_{x \in X} f(x) \quad \text{with} \quad f(x) = \mathbb{E}[f(x, \omega)]$$

As the feasible set of both problems are the same, we get $f(\hat{x}^N) \geq f^*$

We thus have the following unknown optimality gap

$$\mathsf{gap}(\hat{x}) = f(\hat{x}^N) - f^* \geq 0$$

In what follows we are going to find an estimator $\widehat{\mathsf{gap}}(\hat{x}^N)$ for $\mathsf{gap}(\hat{x}^N)$

An upper bound (confidential interval) for $\text{gap}(\hat{x}^N)$

- Generate randomly, from the probability distribution of ω, a (iid) sample with N scenarios
- Obtain $\hat{x}^N$ a solution of the resulting SAA problem
- Generate randomly, from the probability distribution of ω, another (iid) sample with $N' >> N$ scenarios
- Compute $f^{N'}(\hat{x}^N) = \frac{1}{N'} \sum_{i=1}^{N'} f(\hat{x}^N, \omega^i)$
- Compute the variance of $f^{N'}(\hat{x})$

$$\hat{\sigma}^2_{N'}(\hat{x}^N) := \frac{1}{N'(N'-1)} \sum_{i=1}^{N'} [f(\hat{x}^N, \omega^i) - f^{N'}(\hat{x}^N)]^2$$

- Compute the upper bound for the $100(1-\alpha)$-confidential interval of $f(\hat{x}^N)$:

$$U'_N(\hat{x}^N) := f^{N'}(\hat{x}^N) + z_\alpha \hat{\sigma}_{N'}(\hat{x}^N)\,,$$

where $z_\alpha = \Phi^{-1}(1-\alpha)$ and $\Phi(z)$ is standard normal distribution. Ex: if $\alpha = 5\%$, then $z_\alpha \approx 1.64$.

- Before defining a lower bound for the confidential interval of gap$(\hat{x}^N)$, notice that, for every given $x \in X$

$$f(x) = \mathbb{E}[f^N(x)] \geq \mathbb{E}[\min_{y \in X} f^N(y)] = \mathbb{E}[\hat{f}^N]$$

(because $f^N(x)$ is an unbiased estimator of $f(x)$)

- Hence, we get the following useful inequality

$$f^* \geq \mathbb{E}[\hat{f}^N]$$

- We can estimate $\mathbb{E}[\hat{f}^N]$ by solving several SAA problems

A lower bound (confidential interval) for $\mathsf{gap}(\hat{x}^N)$

- Choose $M > 0$, and randomly generate M samples of size N
- Solve M problems SAA to obtain $\hat{f}_i^N$, $i = 1, \ldots, M$
- Compute the unbiased estimator of $\mathbb{E}[\hat{f}^N]$:

$$\bar{f}^{N,M} := \frac{1}{M} \sum_{i=1}^{M} \hat{f}_i^N$$

- Compute the variance of $\bar{f}^{N,M}$:

$$\hat{\sigma}^2_{N,M} := \frac{1}{M(M-1)} \sum_{i=1}^{M} [\hat{f}_i^N - \bar{f}^{N,M}]^2$$

- Compute the lower bound for the $100(1-\alpha)$-confidential interval of $\mathbb{E}[\hat{f}^N]$:

$$L'_N := \bar{f}^{N,M} - t_{\alpha,\nu} \hat{\sigma}_{N,M} \,,$$

  where $\nu = M - 1$ and $t_{\alpha,s}$ is the critical value of the Student's t-distribution with ν degrees of freedom

Evaluating a candidate solution

Given $\hat{x}^N \in X$, solution of a SAA problem, we wish to estimate the optimality gap

$$\texttt{gap}(\hat{x}^N) = f(\hat{x}^N) - f^* \geq 0$$

Upper bound

In order to calculate a statistical upper bound $U_{N'}(\hat{x}^N)$ for gap$(\hat{x}^N)$ we "only" need to evaluate $f^{N'}(x^N)$ and calculate its variance

$$U'_N(\hat{x}^N) := f^{N'}(\hat{x}^N) + z_\alpha \hat{\sigma}_{N'}(\hat{x}^N)$$

Evaluating a candidate solution

Given $\hat{x}^N \in X$, solution of a SAA problem, we wish to estimate the optimality gap

$$\mathtt{gap}(\hat{x}^N) = f(\hat{x}^N) - f^* \geq 0$$

Upper bound

In order to calculate a statistical upper bound $U_{N'}(\hat{x}^N)$ for gap$(\hat{x}^N)$ we "only" need to evaluate $f^{N'}(x^N)$ and calculate its variance

$$U'_N(\hat{x}^N) := f^{N'}(\hat{x}^N) + z_\alpha \hat{\sigma}_{N'}(\hat{x}^N)$$

Lower bound

In order to calculate a statistical lower bound $L_{N,M}$ to $\mathbb{E}[\hat{f}^N]$ we need to solve M SAA problems and calculate its average and variance

$$L'_N := \bar{f}^{N,M} - t_{\alpha,\nu}\hat{\sigma}_{N,M}$$

$$\mathsf{gap}(\hat{x}^N) = f(\hat{x}^N) - f^* \geq 0$$

- We have that $\hat{\mathsf{gap}}(\hat{x}^N) := U_{N'}(\hat{x}) - L_{N,M} \geq 0$. Then,

$$[0,\ \hat{\mathsf{gap}}(\hat{x}^N)]$$

  is a $(1-2\alpha)$-confidence interval for the optimality gap $\mathsf{gap}(\hat{x}^N)$

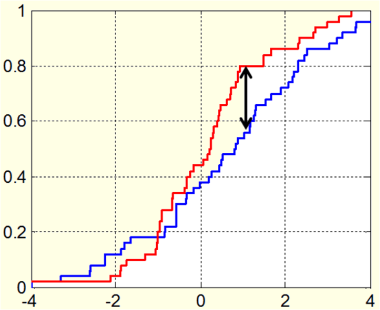
- If the estimator $\hat{\mathsf{gap}}(\hat{x}^N) := U_{N'}(\hat{x}) - L_{N,M}$ is small enough, so is the optimality gap $\mathsf{gap}(\hat{x}^N)$
- Hence, we can say that $\hat{x}^N$ is a good candidate for solving the true problem

$$f^* = \min_{x \in X} f(x) \quad \text{with} \quad f(x) = \int_{\omega \in \Omega} f(x,\omega) dP(\omega)$$

The KS-test
Kolmogorov-Smirnov test

$$\hat{x}^N \in \arg\min_{x \in X} f^N(x), \qquad f^N(x) = \frac{\sum_{i=1}^N f(x, \omega^i)}{N} \quad (SAA)$$

- In order to infer if the sample size N is satisfactory we may compare
  - $f^N(\hat{x}^N)$ with $f^{N'}(\hat{x}^N)$ (average of the individual costs $f(\hat{x}^N, \tilde{\omega}^j)$)
  - Empirical distribution of the individual costs $f(\hat{x}^N, \tilde{\omega}^j)$, $\; j = 1, \ldots, N'$ and $f(\hat{x}^N, \omega^i)$, $\; i = 1, \ldots, N$ (KS-test)



The KS-test

Another idea widely used in practice is:

- Given N and M, generate M samples of size N and solve M problems SAA
- Compare the most important variables of the M SAA solutions $\hat{x}_i^N$, $i = 1, \ldots, M$
- Evaluate the SAA solutions using a larger sample $\{\tilde{\omega}^1, \ldots, \tilde{\omega}^{N'}\}$
- Compare the empirical cost distributions

If there is a certain "adherence" among the results, the size N can be considered satisfactory. Otherwise, it is suggested to increase N

Importance of simulation

- It allows us to analyze the quality of solution obtained with the stochastic model
- it allows us to estimate an appropriated sample size

Consider the following 2-SLP

$$\min_{x \in X} f(x) \quad \text{com} \quad f(x) := c^\top x + \mathbb{E}[Q(x,\xi)] \ \text{ with}$$

$$Q(x,\xi) := \min_y q^\top y \quad \text{s.a} \quad Tx + Wy = \xi, \ \ y \geq 0$$

In this example, $x, y \in \Re^{60}$, $T, W \in \Re^{40\times 60}$ and $X = \{x \geq 0 : Ax = b\}$ with $b \in \Re^{30}$

The random vector $\xi = h(\omega)$ follows a multivariate probability distribution

The problem's data and scenarios are available at the link
`www.oliveira.mat.br/teaching`

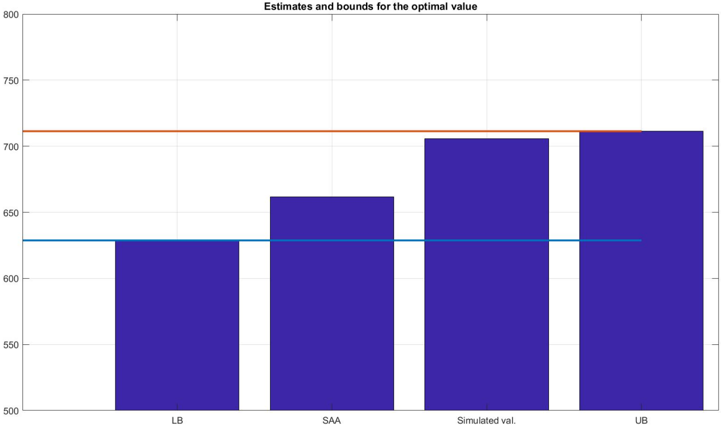
A line of the file `Sample1.csv` = a scenario ξ^i (first line = first scenario)

Solve the Equivalent deterministic for $N = 5, 10, 1\,000$ and $10\,000$

A numerical example

N = 5 scenarios. Simulation N'=1000, M = 10. Sample 1

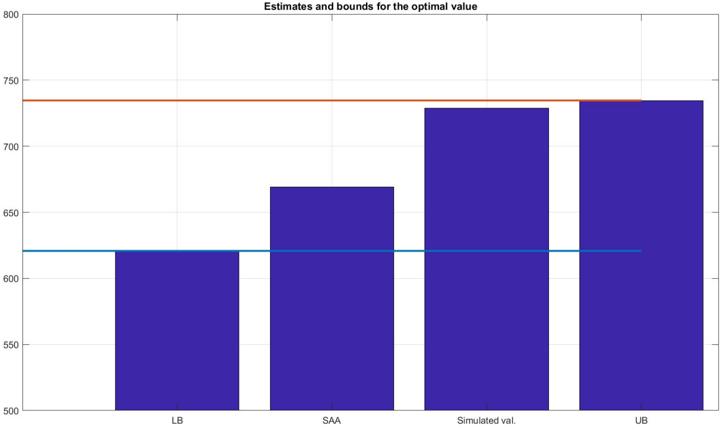
- Lower bound: 628.744
- SAA value: 661.549
- Simulated value: 705.676
- Upper bound: 711.364



A numerical example

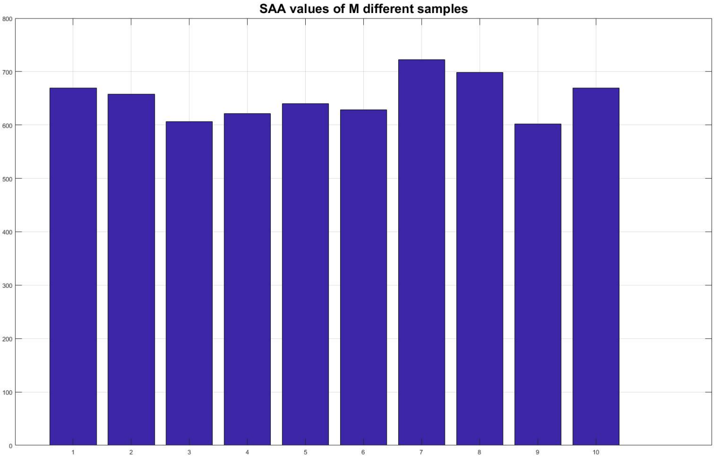
N = 5 scenarios. Simulation N'=1000, M = 10. Sample 2

- Lower bound: 620.804
- SAA value: 669.281
- Simulated value: 728.717
- Upper bound: 734.633



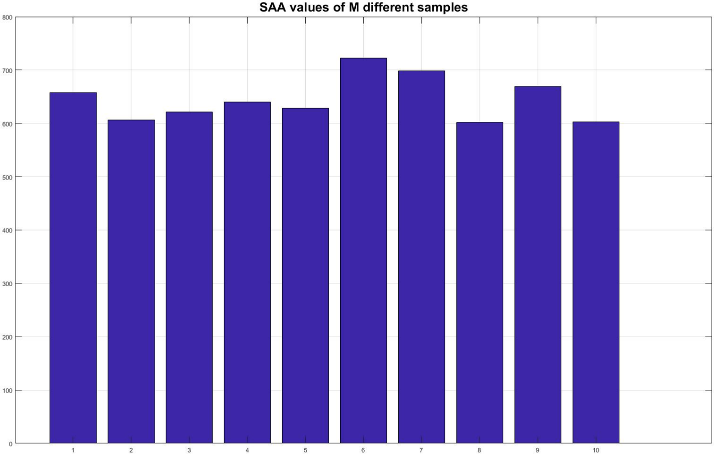
A NUMERICAL EXAMPLE

N = 5 SCENARIOS. SIMULATION N'=1000, M = 10. SAMPLE 1



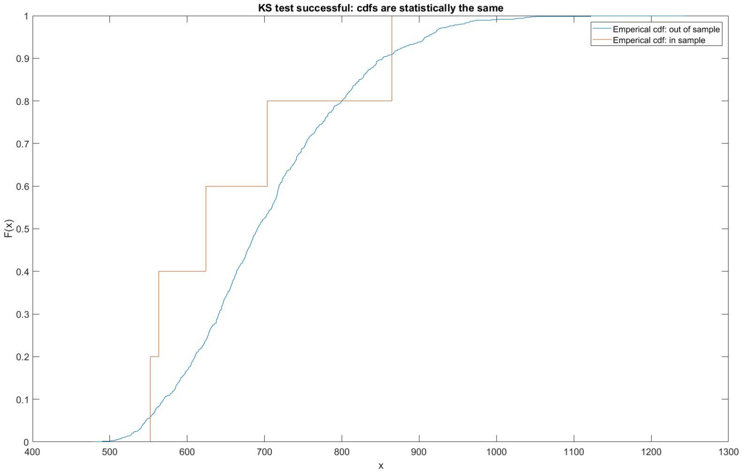
A numerical example

N = 5 scenarios. Simulation N'=1000, M = 10. Sample 2



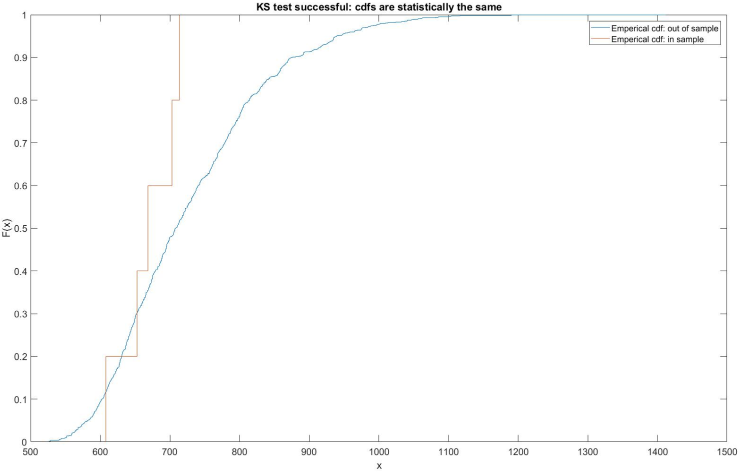
A numerical example

N = 5 scenarios. Simulation N'=1000, M = 10. Sample 1



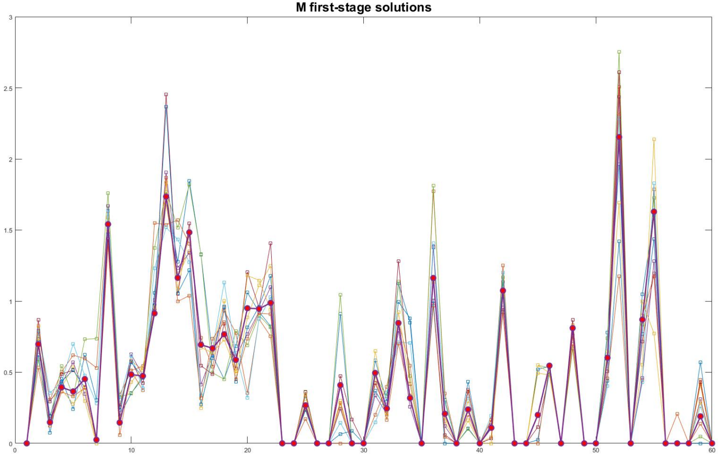
A numerical example

N = 5 scenarios. Simulation N'=1000, M = 10. Sample 2



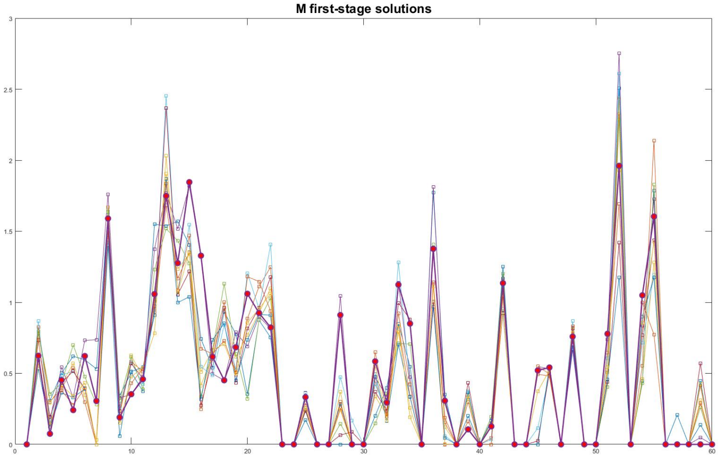
A numerical example

N = 5 scenarios. Simulation N'=1000, M = 10. Sample 1



A numerical example

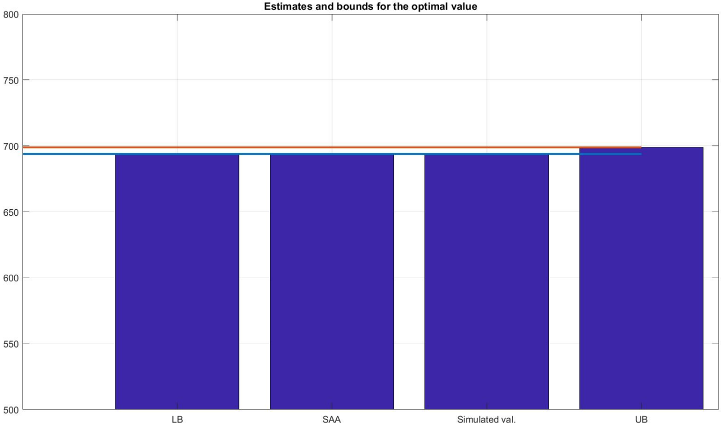
N = 5 scenarios. Simulation N'=1000, M = 10. Sample 2



A numerical example

N = 100 scenarios. Simulation N'=1000, M = 10. Sample 1

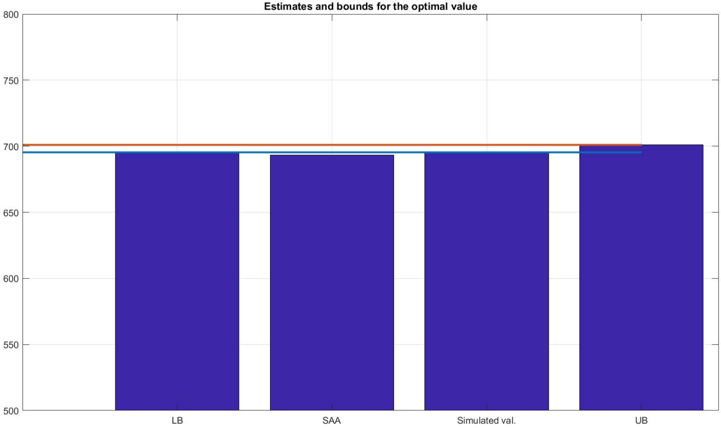
- Lower bound: 693.909
- SAA value: 693.871
- Simulated value: 693.746
- Upper bound: 698.871



A numerical example

N = 100 scenarios. Simulation N'=1000, M = 10. Sample 2

- Lower bound: 695.389
- SAA value: 693.353
- Simulated value: 695.700
- Upper bound: 701.052



This means that the optimal value of

$$\min_{x \in X} f(x)\,, \quad \text{with} \quad f(x) := c^\top x + \mathbb{E}[Q(x, \xi)],$$

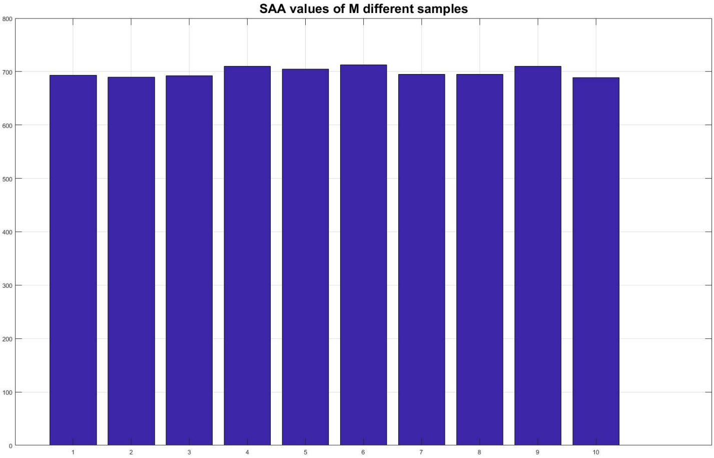
is withing the interval

$$[695.389,\ 701.052]$$

with 90% of confidence

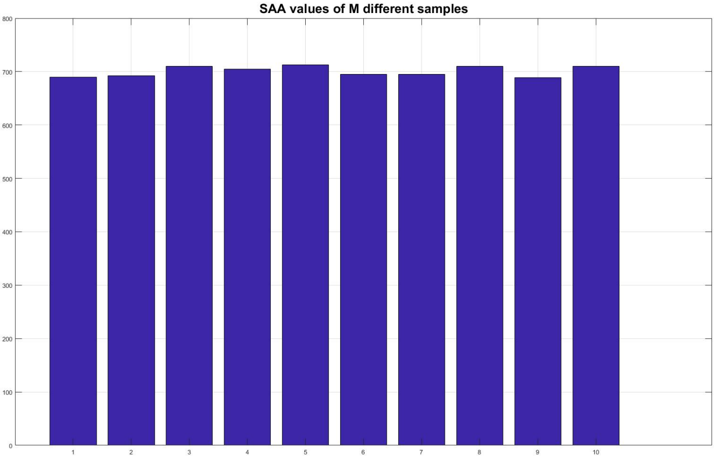
A numerical example

N = 100 scenarios. Simulation N'=1000, M = 10. Sample 1



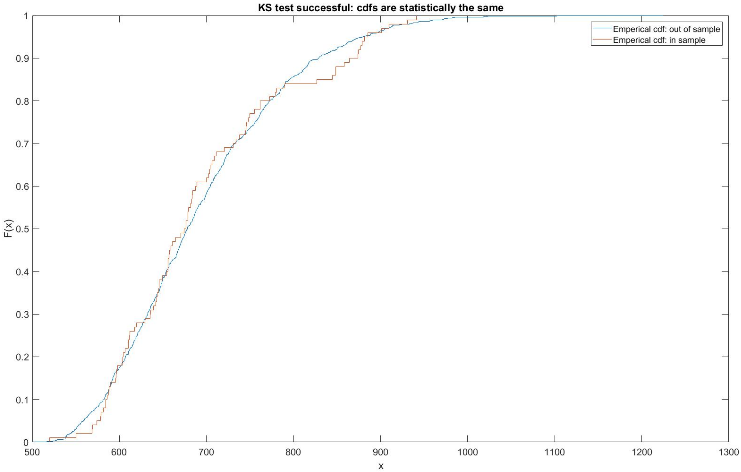
A numerical example

N = 100 scenarios. Simulation N'=1000, M = 10. Sample 2



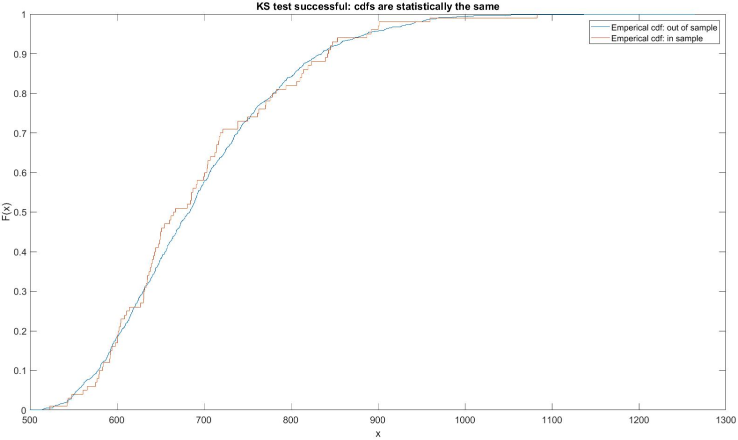
A numerical example

N = 100 scenarios. Simulation N'=1000, M = 10. Sample 1



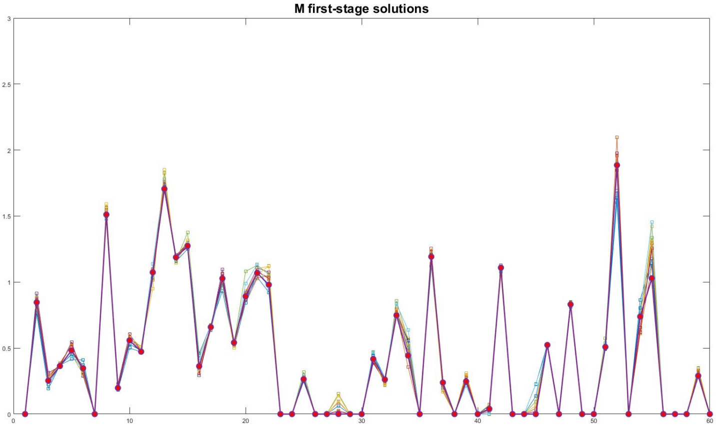
A numerical example

N = 100 scenarios. Simulation N'=1000, M = 10. Sample 2



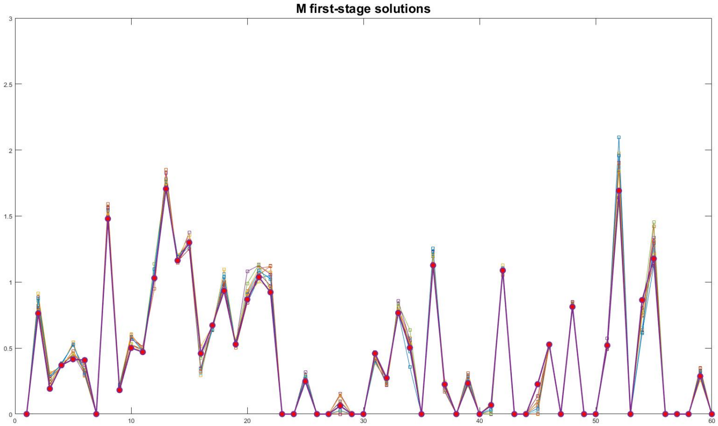
A NUMERICAL EXAMPLE

N = 100 SCENARIOS. SIMULATION N'=1000, M = 10. SAMPLE 1



A numerical example

N = 100 scenarios. Simulation N'=1000, M = 10. Sample 2



Further references

- Shapiro, A., Dentcheva, D., & Ruszczynski, A. (2021).
  Lectures on stochastic programming: modeling and theory.

Contact

✉ welington.oliveira@mines-paristech.fr

www.oliveira.mat.br