Stochastic Optimization
Decomposition Methods for Two-stage problems

V. Leclère

December 01st 2023

École des Ponts ParisTech

Presentation Outline

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)

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

Can be written in a splitted formulation

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

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

$$\begin{aligned}
\min_{\bar{u}_0,\{u_0^s, u_1^s\}_{s\in[\![1,S]\!]}} \quad & \sum_{s=1}^{S} \pi^s L(u_0^s, \xi^s, u_1^s) \\
s.t \quad & g(u_0^s, \xi^s, u_1^s) \leq 0, \qquad \forall s \in [\![1, S]\!] \\
& u_0^s = \bar{u}_0
\end{aligned}$$

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

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

Dualizing non-anticipativity constraint

$$\min_{\{u_0^s, u_1^s\}_{s \in [\![1,S]\!]}} \quad \sum_{s=1}^{S} \pi^s L(u_0^s, \xi^s, u_1^s)$$

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

$$u_0^s = \sum_{s'} \pi^{s'} u_0^{s'} \qquad \forall s$$

is equivalent to

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

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

Dualizing non-anticipativity constraint

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

is equivalent to

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

Dualizing non-anticipativity constraint

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

is equivalent to

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

Dualizing non-anticipativity constraint I

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

is equivalent to

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

Dualizing non-anticipativity constraint

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

is equivalent to

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

Dualizing non-anticipativity constraint II

Thus, the dual problem reads

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

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

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

$$\min_{\{u_0^s, u_1^s\}} \quad L(u_0^s, \xi^s, u_1^s) + \lambda^s u_0^s$$

$$s.t \qquad g(u_0^s, \xi^s, u_1^s) \leq 0$$

Dualizing non-anticipativity constraint II

Thus, the dual problem reads

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

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

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

$$\min_{\{u_0^s,u_1^s\}} \quad L(u_0^s,\xi^s,u_1^s) + \lambda^s u_0^s$$

$$s.t \qquad g(u_0^s,\xi^s,u_1^s) \leq 0$$

Price of information

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

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

$$s.t \quad g(u_0^s, \xi^s, u_1^s) \leq 0$$

- $\boldsymbol{\lambda} = 0$ 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 $\mathbb{E}\big[\boldsymbol{\lambda}\big] = 0$
2. For each scenario solve

$$\begin{aligned} \min_{u_0^s, u_1^s} \quad & L(u_0^s, \xi^s, u_1^s) + \lambda^s u_0^s \\ s.t \quad & g(u_0^s, \xi^s, u_1^s) \leq 0 \end{aligned}$$

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.

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 $\mathbb{E}\big[\boldsymbol{\lambda}\big] = 0$
2. For each scenario solve

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

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

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

where $u_1^s(\bar{u}_0)$ is the optimal recourse for the first-stage control $\bar{u}_0$

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

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

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

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

Which we rewrite

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

with

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

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

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

Linear 2-stage stochastic program : Extensive Formulation

The associated extensive formulation read

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

Which we rewrite

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

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

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

Decomposition of linear 2-stage stochastic program

We rewrite the extended formulation as

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

Note that $Q^s(u_0)$ is a polyhedral function of u_0, hence $\theta^s \geq Q^s(u_0)$ can be rewritten $\theta^s \geq \alpha_k^s \cdot u_0 + \beta_k^s, \forall k$.

Decomposition of linear 2-stage stochastic program

We rewrite the extended formulation as

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

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

Recall that

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

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

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

Obtaining (optimality) cuts II

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

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

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

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

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

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.

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}.

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

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

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

4. construct S new cuts with

$$\alpha_{K+1}^s := -(T^s)^\top \lambda_{K+1}^s, \qquad \beta_{K+1}^s := h^s \cdot \lambda_{K+1}^s$$

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}.

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

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

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

4. construct S new cuts with

$$
\alpha_{K+1}^s := -(T^s)^\top \lambda_{K+1}^s, \qquad \beta_{K+1}^s := h^s \cdot \lambda_{K+1}^s
$$

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 $Q(u_0) := \sum_{s \in S} \pi^s Q^s(u_0) \geq \alpha_k \cdot u_0 + \beta_k$.
2. Solve the master problem, with optimal primal solution u_0^{K+1}.

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

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

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

4. construct new cut with

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

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

$$\max_{\lambda^s \in \mathbb{R}^m} \quad \lambda^s \cdot \big(h^s - T^s u_0^k\big)$$
$$s.t. \quad W^s \cdot \lambda^s \leq q^s$$

      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

$$\bar{\lambda}^k \cdot \big(h^s - T^s u_0\big) \leq 0$$

      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.

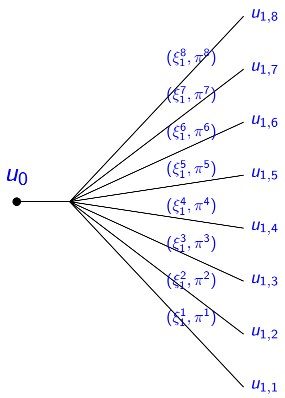
Comparison of Progressive Hedging and L-shaped

	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

Presentation Outline

Where do we come from: two-stage programming



- We take decisions in two stages

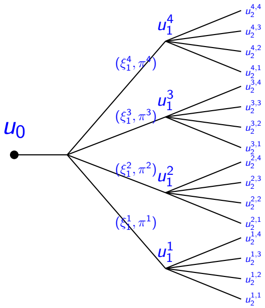
$$u_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow \boldsymbol{u}_1 \ ,$$

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

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

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

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 realization of $(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_t)$, and to which is attached a probability p_n.
- Then, the extensive formulation reads

$$\min_{\{u_n\}_{n\in\mathcal{T}}} \qquad \sum_{n\in\mathcal{T}} \pi^n c^n(u_n)$$

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

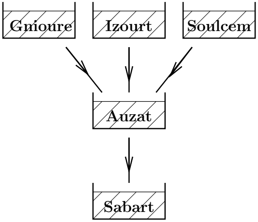
$$\min_{\{u_n\}_{n\in\mathcal{T}}} \quad \sum_{n\in\mathcal{T}} \pi^n c_n(u_n) = \sum_{s\in S} \pi^s \sum_{n\in S} c_n(u_n)$$

- The splitted extended formulation reads

$$\begin{aligned} \min_{\{u_{s,t}\}_{s\in S, t\in[\![0,T]\!]}} \quad & \sum_{s\in S} \pi^s \sum_{t=0}^{T} c_{s,t}(u_{s,t}) \\ s.t. \quad & u_{s,t} = u_{s',t} \qquad \forall t, \forall n \in \mathcal{N}_t, \forall s, s' \ni n \end{aligned}$$

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

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.

Presentation Outline

Optimization Problem

We want to solve the following optimization problem

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

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 chosen knowing the realization of the noises $\boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_t$ but without knowing the realization of the noises $\boldsymbol{\xi}_{t+1}, \dots, \boldsymbol{\xi}_{T-1}$.

Information structure

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 chosen without knowledge of the realization 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

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 chosen without knowledge of the realization 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

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 chosen without knowledge of the realization 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

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 chosen without knowledge of the realization 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

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 chosen without knowledge of the realization 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

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

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

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

Presentation Outline

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 :

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

- 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 ξ_{t+1} by an estimation (often its expectation $\mathbb{E}[\xi_{t+1}]$), 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

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

    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 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

$$v^{2SA} \leq v^\sharp$$

$$v_m^{2SA} \to v^{2SA}$$

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

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 $\boldsymbol{u}_t = \Phi_t(\xi_0, \dots, \xi_{t+1})$ 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.

Presentation Outline

Nested Bender's decomposition