Dual SDDP for risk-averse multistage stochastic programs

Bernardo Freitas Paulo da Costa,* Vincent Leclère,†

July 22, 2021

Abstract

Risk-averse multistage stochastic programs appear in multiple areas and are challenging to solve. Stochastic Dual Dynamic Programming (SDDP) is a well known tool to address such problems under time-independence assumptions. We show how to derive a dual formulation for these problems and apply an SDDP algorithm, leading to converging and deterministic upper bounds for risk averse problems.

Keywords. Stochastic programming, Dynamic programming, SDDP, Risk measures, Duality

AMS subject classification. 90C15, 90C39, 49N15

1 Introduction

Multistage stochastic programming is a powerful framework with numerous applications [WZ05, GZ13], especially in the finance, energy and supply chain sectors. If the uncertainty is finitely supported, those problems can be seen as large scale deterministic problems. Unfortunately, when the number of stages is bigger than 4 or 5, the size of the deterministic equivalent is too big to be tackled directly. Among the classical approaches, one of the most successful paradigms consists in leveraging time independence assumptions to derive Bellman equations [Bel57, Ber05]. The Stochastic Dual Dynamic Programming (SDDP) algorithm, and its numerous variants ([PP91, PdMF13, BDZ17, ZAS19, ACdC20]), consists in using those equations to derive approximations of the cost-to-go functions. It has been successfully used on a number of real-world problems, especially in the field of energy.

While the classical formulation of a multistage program is risk neutral, meaning that we aim at minimizing an expected cost, a large part of the recent litterature [Sha12] has been devoted to efficiently introduce risk aversion in this framework, in particular inside the SDDP algorithm. Coherent risk measures [ADEH99] have become a usual tool to represent risk aversion in stochastic optimization problem. In multistage stochastic programming, aiming at minimizing a risk measure of the sum of costs leads to time-inconsistency, therefore it appears more natural to minimize a time-consistent dynamic risk measure (see e.g. [BF06]). The easiest way to come up with a time-consistent risk averse problem is to use composed markovian risk measures [Rus10], which, roughly speaking, means replacing the expectation by a risk measure inside the dynamic programming equation.

*UFRJ, Rio de Janeiro, Brazil
†CERMICS, Ecole des Ponts, Marne-la-Vallée, France
Further, the classical SDDP focuses on deriving outer approximations of the cost-to-go function, leading to exact lower bounds on the problem. In a risk neutral framework, upper bounds can be obtained through statistical methods. Unfortunately it is unclear how to extend such statistical methods to the risk averse setting [STdCS13]. Instead of statistical upper bounds, one can use exact upper bounds: Through backward recursion as in [PdMF13]; by maintaining upper and lower bounds at all stages as in the problem-child approaches [BDZ17, DDB20]; or through Fenchel duality [LCC20, GSC19]. Backward recursion requires a given set of points at which to evaluate an upper bound on the cost-to-go function at time $t$, that is then used to obtain the upper bounds at time $t-1$ and so on. While this idea can be used for risk-averse problems, it is more a one-time evaluation than an improving process. On the other hand, maintaining upper and lower bounds in the problem-child method can be used in a risk averse setting. In this paper we adapt the third approach to the risk averse setting.

More precisely, let $(\Omega, F, P)$ be a probability space, and $\{\omega_t\}_{t\in[T]}$ be a sequence random variables. We consider the following risk-averse multistage linear program (RA-MSLP)

$$\begin{aligned}
\text{min} & \quad \rho_1 \left( c_1^\top y_1 + \rho_2 |\omega_1| \left( c_2^\top y_2 + \cdots + \rho_T |\omega_{T-1}| (c_T^\top y_T) \right) \right) \\
\text{s.t.} & \quad A_t x_t + B_t x_{t-1} + T_t y_t = d_t \\
& \quad x_t \geq 0, y_t \geq 0 \\
& \quad x_t, y_t \preceq \omega_t \quad \forall t \in [T]
\end{aligned}$$

(1a)

where $\rho_t |\omega_t|$ is a coherent risk measure conditional on the past noises $\omega_{[t]} := \{\omega_1, \ldots, \omega_t\}$, all equalities hold almost surely, and constraint (1d) is the so-called non-anticipativity constraint, ensuring that decisions $x_t, y_t$ are taken with only the information available at time $t$.

Assuming that $\{\omega_t\}_{t\in[T]}$ is a sequence of independent random variables, this problem can be tackled by Dynamic Programming through the use of risk-averse Linear Bellman Operators, given by

$$B_t(V) : x_0 \mapsto \inf_{x,y} \rho_t \left[ c_t^\top y + V(x) \right]$$

s.t. $A_t x + B_t x_0 + T_t y = d_t$

$$x, y \geq 0.$$ (2)

Indeed, with the following Bellman’s recursion

$$V_t = B_t(V_{t+1}), \quad V_{T+1} = 0$$ (3)

the value of Problem (1) is given by $V_0(x_0)$.

In the risk neutral case, it has been shown in [LCC+20] that we can also derive a Bellman-like recursion for the Fenchel transform of $V$. Equivalently, this recursion can be obtained by dualizing the extensive formulation of the MLSP problem, and recognizing a time-decomposition, as presented in [GSC19]. In both cases, this provides a dual problem on which SDDP can be applied, and for which lower bounds correspond to upper bounds of (1). Here we adopt the same idea for the risk averse case: we dualize the extended formulation and recognize a time-decomposition of the dual problem. This leads to a Bellman recursion on the perspective of the Fenchel transform of the value function, and to upper bounds for the risk-averse setting.

**Structure of the paper.** We start, in Section 2, with a simple three-stage example showing how we can obtain a decomposition of the dual problem. We then proceed to show, in Section 3, the time decomposition for a generic risk averse linear multistage problem. Section 4 shows the link between said time decomposition of the dual and the co-perspective of the value function. In
particular, we show that there exists a linear Bellman operator for this dual, paving the way to the use of SDDP. Finally, Section 5 gives a numerical illustration of our approach, by detailing the dual SDDP algorithm used and showing encouraging results.

2 An illustrative example

We start our example introducing the notation for a risk-neutral problem. For concreteness, we suppose a three-stage optimization problem:

\[
\begin{align*}
\min_{x_1,x_2_1,x_3_{ij}} & \quad c_1^\top x_1 + \sum_i p_i c_2^\top x_2_i + \sum_{ij} p_{ij} q_{ij} c_3^\top x_{3ij} \\
\text{s.t.} & \quad A_1 x_1 = b_1 \quad A_2 x_{2i} + B_2 x_1 = b_2_i \quad A_3 x_{3ij} + B_3 x_{2i} = b_{3j} \\
& \quad x_1 \geq 0 \quad x_{2i} \geq 0 \quad x_{3ij} \geq 0.
\end{align*}
\]  

(4)

The stagewise independent structure is reflected in:

- The conditional probabilities \( q_{ij} \) and costs \( c_{3j} \), that do not depend on the second stage scenario \( i \).
- The transition constraint \( A_{3j} x_{3ij} + B_{3j} x_{2i} = b_{3j} \), whose coefficients do not depend on the second stage scenario \( i \), but only on its initial condition \( x_{2i} \) and the corresponding decision \( x_{3ij} \).

For the risk-averse case, we must replace the second (resp. third) stage costs by an evaluation of the risk measure for all scenarios \( i \) (resp. \( ij \)). Let us illustrate how this might be achieved for the nested AV@R risk measure. For each scenario \( i \), the optimization problem we consider at the second stage will be

\[
\begin{align*}
\min_{x_{2i},x_{3ij}} & \quad c_{2i}^\top x_{2i} + \rho q_{ij} c_{3j}^\top x_{3ij} \\
\text{s.t.} & \quad A_{2i} x_{2i} + B_{2i} x_1 = b_{2i} \quad A_{3j} x_{3ij} + B_{3j} x_{2i} = b_{3j} \\
& \quad x_{2i} \geq 0 \quad x_{3ij} \geq 0.
\end{align*}
\]  

(5)

The risk measure for the future of scenario \( i \), corresponding to scenarios \( ij \) for all \( j \), can be modeled with extra variables:

\[
\begin{align*}
\min_{x_{2i},z_{2i},x_{3ij},u_{3ij}} & \quad c_{2i}^\top x_{2i} + z_{2i} + \frac{1}{\alpha} \sum_j q_{ij} u_{3ij} \\
\text{s.t.} & \quad A_{2i} x_{2i} + B_{2i} x_1 = b_{2i} \quad A_{3j} x_{3ij} + B_{3j} x_{2i} = b_{3j} \\
& \quad x_{2i} \geq 0 \quad x_{3ij} \geq 0 \quad \alpha z_{2i} + u_{3ij} \geq c_{3j}^\top x_{3ij} \quad u_{3ij} \geq 0.
\end{align*}
\]  

(6)

Putting it all together with extra variables, the optimization problem corresponding to

\[
\begin{align*}
\min_{x_1,x_{21},x_{3ij}} & \quad c_1^\top x_1 + \rho p_i c_{2i}^\top x_{2i} + \rho q_{ij} c_{3j}^\top x_{3ij} \\
\text{s.t.} & \quad A_1 x_1 = b_1 \quad A_{2i} x_{2i} + B_{2i} x_1 = b_{2i} \quad A_{3j} x_{3ij} + B_{3j} x_{2i} = b_{3j} \\
& \quad x_1 \geq 0 \quad x_{2i} \geq 0 \quad x_{3ij} \geq 0.
\end{align*}
\]  

(7)
analogously for

\[
\begin{align*}
\min_{x_1, x_{2i}, x_{3ij}} & \quad c_i^T x_1 + z_1 + \frac{1}{\lambda} \sum_{i} p_i u_{2i} \\
\text{s.t.} & \quad A_1 x_1 = b_1 \\
& \quad x_{2i} \geq 0 \quad \text{(8)} \\
& \quad z_1 + u_{2i} \geq c_i^T x_{2i} + z_2i + \frac{1}{\lambda} \sum_{j} q_{ij} u_{3ij} \quad u_{3ij} \geq 0
\end{align*}
\]

Interchanging the sup and min, by linear programming strong duality, we obtain the dual problem

\[
\begin{align*}
\sup_{\gamma_1, \gamma_{3ij}, \delta_{2i}, \delta_{3ij}} & \quad (c_1 + A_2^T \lambda_1 - \mu_1 + \sum_{i} p_i B_2^T \lambda_{3ij})^T x_1 \\
& \quad + (1 - \sum_{i} p_i \gamma_{2i}) z_1 \\
& \quad + \sum_{i} p_i \left( \frac{1}{\alpha} - \gamma_{2i} - \delta_{2i} \right) u_{2i} \\
& \quad + \sum_{i} p_i \left( A_2^T \lambda_{2i} + \sum_{j} q_{ij} B_2^T \lambda_{3ij} - \mu_{2i} + \gamma_{2ij} c_{2i} \right)^T x_{2i} \\
& \quad + \sum_{i} p_i \left( \gamma_{2i} - \sum_{j} q_{ij} \gamma_{3ij} \right) z_{2i} \\
& \quad + \sum_{i} p_i q_{ij} \left( \frac{1}{\alpha} \gamma_{2ij} - \gamma_{3ij} - \delta_{3ij} \right) u_{3ij} \\
& \quad + \sum_{i} p_i q_{ij} \left( A_2^T \lambda_{3ij} - \mu_{3ij} + \gamma_{3ij} c_{3ij} \right)^T x_{3ij}
\end{align*}
\]
Transforming the minimum over the primal variables to constraints, the dual problem becomes

\[
\sup_{\lambda_1, \lambda_2, \lambda_{3ij}} \inf_{\mu_1, \mu_{2i}, \mu_{3ij}} -\lambda_1^T b_1 - \sum_i p_i \lambda_2^T b_{2i} - \sum_{ij} p_{ij} \lambda_{3ij}^T b_{3ij} \\
+ \sup_{\gamma_{2i}, \gamma_{3ij}} \sum_i p_i \gamma_{2i} = 1 \\
\frac{1}{\delta} = \gamma_{2i} + \delta_{2i} \\
A_{2i}^T \lambda_{2i} + \sum_j q_{ij} B_{3j}^T \lambda_{3ij} + \gamma_{3ij} c_{2i} = \mu_{2i} \\
\sum_j q_{ij} \gamma_{3ij} = \gamma_{2i} \\
\frac{1}{\delta} \gamma_{2i} = \gamma_{3ij} + \delta_{3ij} \\
A_{3ij}^T \lambda_{3ij} + \gamma_{3ij} c_{3ij} = \mu_{3ij} \\
\forall i, j.
\]

Eliminating \(\mu\) and \(\delta\) multipliers, which are positive, this simplifies to

\[
\sup_{\lambda_1, \lambda_2, \lambda_{3ij}} \inf_{\gamma_{2i}, \gamma_{3ij} \geq 0} -\lambda_1^T b_1 - \sum_i p_i \lambda_2^T b_{2i} - \sum_{ij} p_{ij} \lambda_{3ij}^T b_{3ij} \\
+ \sup_{\gamma_{2i}, \gamma_{3ij} \geq 0} 0 \\
\text{s.t.} \quad c_1 + A_{1i}^T \lambda_1 + \sum_i p_i B_{2i}^T \lambda_{2i} \geq 0 \\
\sum_i p_i \gamma_{2i} = 1 \\
\frac{1}{\delta} \geq \gamma_{2i} \\
A_{2i}^T \lambda_{2i} + \sum_j q_{ij} B_{3j}^T \lambda_{3ij} + \gamma_{3ij} c_{2i} \geq 0 \\
\sum_j q_{ij} \gamma_{3ij} = \gamma_{2i} \\
\frac{1}{\delta} \gamma_{2i} \geq \gamma_{3ij} \\
A_{3ij}^T \lambda_{3ij} + \gamma_{3ij} c_{3ij} \geq 0 \\
\forall i, j.
\]

Now, we rearrange this optimization problem to emphasize its time-decomposition structure; it will be interesting to add the auxiliary variables \(\pi_1, \pi_{2i}, \pi_{3ij}\) as in \([LCC+20]\). We also change signs to make it a minimization problem:

\[
\inf_{\lambda_1, \lambda_2, \lambda_{3ij}} \pi_1, \pi_{2i}, \pi_{3ij} \geq 0 \sum_i p_i \lambda_2^T b_{2i} + \sum_{ij} p_{ij} \lambda_{3ij}^T b_{3ij} \\
\text{s.t.} \quad \pi_1 + \sum_i p_i B_{2i}^T \lambda_{2i} \geq 0 \\
\sum_i p_i \gamma_{2i} = 1 \\
\frac{1}{\delta} \geq \gamma_{2i} \\
A_{2i}^T \lambda_{2i} + \sum_j q_{ij} B_{3j}^T \lambda_{3ij} \geq 0 \\
\sum_j q_{ij} \gamma_{3ij} = \gamma_{2i} \\
\frac{1}{\delta} \gamma_{2i} \geq \gamma_{3ij} \\
A_{3ij}^T \lambda_{3ij} \geq \gamma_{3ij} c_{3ij} + A_{3ij}^T \lambda_{3ij} = \pi_{3ij} \forall i, j, j.
\]

To finally obtain a time-decomposition recursion that is amenable to SDDP, we introduce the \textit{perspective value functions}

\[
R_3(\pi_2, \gamma_2) = \inf_{\lambda_{3ij}, \pi_{3ij}, \gamma_{3ij} \geq 0} \sum_j q_{ij} \lambda_{3ij}^T b_{3ij} \\
\text{s.t.} \quad \pi_2 + \sum_j q_{ij} B_{3j}^T \lambda_{3ij} \geq 0 \\
\sum_j q_{ij} \gamma_{3ij} = \gamma_{2i} \\
\frac{1}{\delta} \gamma_{2i} \geq \gamma_{3ij} \\
\gamma_{3ij} c_{3ij} + A_{3ij}^T \lambda_{3ij} \geq \pi_{3ij} \forall j.
\]
The adjective “perspective” is justified because $R_3$ is positively homogeneous of degree one: $R_3(t \cdot \pi_2, t \cdot \gamma_2) = t \cdot R_3(\pi_2, \gamma_2)$ for $t \geq 0$. It can now be used for defining

$$R_2(\pi_1, \gamma_1) = \inf_{\lambda_{2i}, \pi_{2i}, \gamma_{2i} \geq 0} \sum_i p_i \left( \lambda_{2i}^T b_{2i} + R_3(\pi_{2i}, \gamma_{2i}) \right)$$

subject to

$$\pi_1 + \sum_i p_i B_{2i}^T \lambda_{2i} \geq 0$$

$$\frac{1}{2} \gamma_i \geq \gamma_{2i} \quad \forall i$$

$$\gamma_{2i} \pi_{2i} + A_{2i}^T \lambda_{2i} = \pi_{2i} \quad \forall i,$$

which is again homogeneous. Finally, the (negative of the) optimal value of the dual problem can be calculated from

$$\inf_{\lambda_1, \pi_1, \gamma_1 \geq 0} \lambda_1^T b_1 + R_2(\pi_1, \gamma_1)$$

subject to

$$\gamma_1 = 1$$

$$\gamma c_1 + A_1^T \lambda_1 = \pi_1,$$

where we added the extra variable $\gamma_1 = 1$ to be used in $R_2$, and included it in the constraint linking $\pi$, $\lambda$ and $c$, analogously to how it appeared in $R_2$ and $R_3$.

The recursive structure of dependence of the value functions $R_1$, $R_2$ and $R_3$ is similar to that of the primal value functions of SDDP. In this spirit, we can consider running SDDP on the dual problem, which will provide lower bounds for the optimal value of the dual problem, and therefore upper bounds for the optimal value of the primal.

**Remark 1.** The objective function in $R_3$ can have a “terminal cost” $R_4(\pi_{3j}, \gamma_{3j})$, given by the indicator function $I_{\{\pi_{3j} \geq 0\}}$, capturing this constraint in the objective. This further emphasizes the similarity between $R_3$ and $R_2$.

## 3 Time decomposition of the dual of a risk averse SDDP

Let $(\Omega, \mathcal{F})$ be a measurable space with finite $\Omega$. For the stagewise independent case we’re interested in, we take $\Omega = \prod_i \Omega_i$.

### 3.1 Multistage risk averse problem duality

Due to the independence assumption, the multistage risk averse problem can be solved through Dynamic Programming. In particular, we focus on the problem of computing $V_t(x_{n_0})$, where $V_t$ are given by the following Bellman Recursion

$$V_t(x_{t-1}) := \min_{x_t, y_t} \rho_t \left[ c_t^T y_t + V_{t+1}(x_t) \right]$$

subject to

$$A_t x_t + B_t x_{t-1} + T_t y_t = d_t \quad (16a)$$

$$x_t \geq 0, y_t \geq 0 \quad (16b)$$

where $\rho_t[\cdot] = \sup_{\pi_t \in \mathcal{Q}_t} \mathbb{E}[\cdot]$, for a given subset $\mathcal{Q}_t$ of probability measures on $\Omega_t$. We optimize over $x_t$ and $y_t$, seen as measurable functions of $\omega_{[t]}$. For simplicity, we assume that for all $t$, $\rho_t = \rho$ and that $\rho$ is a polyhedral risk measure. Then, we can assume that $\mathcal{Q}_t = \mathcal{Q}$ is given by a finite set $\{q^1, \ldots, q^K\}$ of measures.

Let $\mathcal{T}$ be the tree describing $(\Omega, \mathcal{F})$, that is, such that each node $n$ of depth $t$ is associated with a possible value of $\omega_{[t]}$. We denote $\mathcal{L}$ its set of leaves. For any node $n$ we denote the set of its children by $C_n$. Variable $z_n$ stands for the risk adjusted value of our problem starting...
from node \( n \), and \( \theta_m \) represents the cost-to-go starting from \( n \) and knowing that node \( m \in C_n \) is selected. Then our risk averse problem can be written as the following linear program:

\[
V_{n_0}(\tilde{x}_{n_0}) = \min_{x,\tilde{x},y,z,\theta} \quad z_0
\]

\[
\sum_{m \in C_n} q^k_m \theta_m \leq z_n \quad \forall n \in \mathcal{T} \setminus \mathcal{L}, \forall k \in [K] \quad [\Phi^k_n] \quad (17a)
\]

\[
c^\top_m y_m + z_m \leq \theta_m \quad \forall m \in \mathcal{T} \setminus \{n_0\} \quad [\gamma_m] \quad (17b)
\]

\[
A_m x_m + B_m \tilde{x}_n + T_m y_m = d_m \quad \forall n \in \mathcal{T} \setminus \mathcal{L}, \forall m \in C_n \quad [\lambda_m] \quad (17c)
\]

\[
x_n \geq 0, y_n \geq 0 \quad \forall n \in T \setminus \{n_0\} \quad [\mu_n, \nu_n] \quad (17d)
\]

\[
z_t = 0 \quad \forall \ell \in L \quad [\eta_\ell] \quad (17e)
\]

\[
x_n = \tilde{x}_n \quad \forall n \in T \setminus \mathcal{L} \quad [\pi_n] \quad (17f)
\]

where \( \tilde{x}_{n_0} \) is a parameter and not a variable, and we add the equality \( x_n = \tilde{x}_n \) to highlight the time dynamics. The linear programming dual of the above problem is given by:

\[
D_{n_0}(\tilde{x}_{n_0}) = \sup_{\Phi,\gamma,\lambda} \quad \inf_{x,y,z,\theta} \quad z_0 + \sum_{n \in \mathcal{T} \setminus \{n_0\}} \sum_{k \in [K]} \Phi^k_n \left( \sum_{m \in C_n} q^k_m \theta_m - z_n \right)
\]

\[
+ \sum_{m \in \mathcal{T} \setminus \{n_0\}} \gamma_m (c^\top_m y_m + z_m - \theta_m)
\]

\[
+ \sum_{n \in \mathcal{T} \setminus \mathcal{L}} \sum_{m \in C_n} \lambda_m (A_m x_m + B_m \tilde{x}_n + T_m y_m - d_m)
\]

\[
- \sum_{n \in \mathcal{T} \setminus \{n_0\}} \mu_n x_n - \sum_{n \in \mathcal{T} \setminus \{n_0\}} \nu_n y_n + \sum_{\ell \in L} \eta_\ell z_\ell
\]

\[
+ \sum_{n \in \mathcal{T} \setminus \mathcal{L}} \pi_n (x_n - \tilde{x}_n),
\]

which reads in turn (defining \( \gamma_{n_0} = 1 \)):

\[
D_{n_0}(\tilde{x}_{n_0}) = \sup_{\Phi,\gamma,\lambda} \quad \pi^\top_{n_0} \tilde{x}_{n_0} - \sum_{m \in \mathcal{T} \setminus \{n_0\}} \lambda^\top_m d_m
\]

\[
\sum_{k \in [K]} \Phi^k_n = \gamma_n \quad \forall n \in \mathcal{T} \setminus \mathcal{L} \quad [z_n]
\]

\[
\sum_{k \in [K]} \Phi^k_n q^k_m = \gamma_m \quad \forall n \in \mathcal{T} \setminus \mathcal{L}, \forall m \in C_n \quad [\theta_m]
\]

\[
\gamma_m c_m + T^\top_m \lambda_m \geq 0 \quad \forall m \in \mathcal{T} \setminus \{n_0\} \quad [y_m]
\]

\[
\sum_{m \in C_n} B^\top_m \lambda_m = \pi_n \quad \forall n \in \mathcal{T} \setminus \mathcal{L} \quad [\tilde{x}_n]
\]

\[
A^\top_m \lambda_m + \pi_m \geq 0 \quad \forall m \in \mathcal{T} \setminus \{n_0\} \quad [x_m]
\]

\[
\Phi^k_n \geq 0, \gamma_n \geq 0 \quad \forall n \in \mathcal{T}.
\]

Note that \( \Phi_n \) can be seen as barycentric coordinates of the extreme points of \( \mathcal{Q} \). Thus, the first two constraints can be more compactly written as \( (\gamma_m)_{m \in C_n} \in \gamma_n \mathcal{Q} \).
By backward recursion, this problem can be solved through the following recursive equations, where, for all leaves \( m \in \mathcal{L} \), \( D_m(\pi_m, \gamma_m) = \mathbb{I}_{\{\pi_m = 0\}} \) and for all nodes \( n \in \mathcal{T} \setminus \mathcal{L} \):

\[
D_n(\pi_n, \gamma_n) = \sup_{(\gamma_m, \pi_m) \in C_n} \mathbb{I}_{\{n = n_0\}} \pi_n^T x_{n_0} + \sum_{m \in C_n} -\lambda_m^T d_m + D_m(\pi_m, \gamma_m) \tag{18a}
\]

\[
(\gamma_m)_{m \in C_n} \in \gamma_n \mathcal{Q} \tag{18b}
\]

\[
\gamma_m c_m + T_m^T \lambda_m \geq 0 \quad \forall m \in C_n \tag{18c}
\]

\[
\sum_{m \in C_n} B_m^T \lambda_m = \pi_n \tag{18d}
\]

\[
\pi_m + A_m^T \lambda_m \geq 0 \quad \forall m \in C_n \tag{18e}
\]

By the independence assumption, a backward induction shows that \( D_n = D_n' \) for all nodes \( n \) and \( n' \) of the same depth. More precisely, for \( t \in [T - 1] \), we denote \( \{\omega_j\}_{j \in [J_t]} \) the support of \( \{\omega_t\} \), and we have

\[
D_T(\pi_T, \gamma_T) = \mathbb{I}_{\{\pi_T = 0\}} \tag{19a}
\]

\[
D_t(\pi_t, \gamma_t) = \sup_{(\gamma_j, \pi_j) \in [J]} \mathbb{I}_{\{t = t_0\}} \pi_j^T x_{n_0} + \sum_{j \in [J]} -\lambda_j^T d_j + D_{t+1}(\pi_j, \gamma_j) \tag{19b}
\]

\[
(\gamma_j)_{j \in [J]} \in \gamma_t \mathcal{Q} \tag{19c}
\]

\[
\gamma_j c_j + T_j^T \lambda_j \geq 0 \quad \forall j \in [J] \tag{19d}
\]

\[
\sum_{j \in [J]} B_j^T \lambda_j = \pi_t \tag{19e}
\]

\[
\pi_j + A_j^T \lambda_j \geq 0 \quad \forall j \in [J] \tag{19f}
\]

### 3.2 Relatively complete recourse

Bellman’s recursion [19] cannot directly be used to apply SDDP. Indeed, convergence of SDDP requires additional assumptions, in particular relatively complete recourse (RCR) and boundedness of the state.

One of the difficulties of the dual formulation is that even if relatively complete recourse is ensured in the primal, it is not guaranteed in the dual (see for example [GSC19]). To deal with this problem we can either incorporate feasibility cuts in the algorithm, or add a penalization scheme to ensure RCR. We follow here a slightly different path, equivalent to the penalization approach.

We assume that we can bound the state and control variables a priori by constants \( \bar{x}_n, \bar{y}_n \), and add to Problem (17), for all \( n \in \mathcal{T} \setminus \{n_0\} \), the constraints \( x_n \leq \bar{x}_n \) with associated multiplier \( \zeta_n \), as well as the constraint \( y_n \leq \bar{y}_n \), with associated multiplier \( \xi_n \). Then, following the same steps as above, we keep (19a) for \( t = T \), and we obtain the following decomposition for all other \( t > t_0 \) (\( t = t_0 \) requiring a straightforward modification of the objective function and a new variable):
\[ D_t(\pi_t, \gamma_t) = \sup_{(\gamma_j, A_j, \pi_j, \xi_j) \in [J_t]} -\tilde{x}_t^\top \zeta + \sum_{j \in [J_t]} -d_j^\top \lambda_j - g_{t+1}^\top \xi_j + D_{t+1}(\pi_j, \gamma_j) \]  
(20a)

\[ (\gamma_j)_{j \in [J_t]} \in \gamma_t \mathcal{Q}, \]  
(20b)

\[ \gamma_j c_j + T_j^\top \lambda_j + \xi_j \geq 0 \]  
\forall j \in [J_t]  
(20c)

\[ \left( \sum_{j \in [J_t]} B_j^\top \lambda_j \right) + \zeta \geq \pi_t \]  
(20d)

\[ \pi_j + A_j^\top \lambda_j \geq 0 \]  
\forall j \in [J_t]  
(20e)

\[ \xi_j \geq 0, \quad \zeta \geq 0 \]  
(20f)

This decomposition satisfies the RCR conditions. Indeed, for every \( \pi_t \), and every \( \gamma_t \geq 0 \), any \( \gamma \in \gamma_t \mathcal{Q} \), and \( \lambda = 0 \) is admissible, using slack variable \( \zeta \) as needed. Then, \( \pi_j \) is determined by (20e), and the inequalities (20c) can be adjusted using \( \xi_j \).

**Remark 2.** Assuming boundedness of \( x \) is quite natural in most settings. Boundedness of \( y \) is more demanding, and can actually be replaced by one of the following assumptions:

- \( \exists q \in \mathcal{Q} \) such that \( E_q[c] \geq 0 \) (choose \( \lambda = 0 \), e.g. \( c \geq 0 \).  
- \( \text{range}(T^\top) \cap \mathbb{R}^n_+ \neq \emptyset \)  
- \( \forall j \in [J] \), \( c_j + \text{range}(T_j)^\top \cap \mathbb{R}^n_+ \neq \emptyset \)

### 4 Dual risk averse Bellman operator

We introduce convex analysis tools that shed new light on the link between the primal and dual value functions given in Section 3. More precisely we show that the dual value function \( D_t \) introduced in (19) is the negative of coperspective of the primal value function \( V_t \).

#### 4.1 Homogeneous Fenchel duality

Let \( f : \mathbb{R}^n \rightarrow \mathbb{R}^n \rightarrow ]-\infty, \infty[ \) be a proper lower semicontinuous convex function. Recall (see [Com18] for more details) that the perspective function of \( f \), denoted \( \tilde{f} \) is defined as

\[ \tilde{f} : \mathbb{R} \times \mathbb{R}^n \rightarrow ]-\infty, +\infty[ : (\gamma, x) \mapsto \begin{cases} \gamma f(x/\gamma), & \text{if } \gamma > 0 \\ \text{rec}(f)(x), & \text{if } \gamma = 0 \\ +\infty & \text{otherwise} \end{cases} \]  
(21)

where \( \text{rec}(f) \) is the recession function of \( f \) defined as

\[ \text{rec}(f) : \mathbb{R}^n \rightarrow ]-\infty, +\infty[ : x \mapsto \lim_{t \rightarrow +\infty} \frac{f(z + tx)}{t}, \]  
(22)

where \( z \) can be chosen as any point in the domain of \( f \). Both the recession and the perspective function of proper lower semicontinuous convex function are proper lower semicontinuous convex function as well. Recall that the Fenchel conjugate of \( f \) is defined as

\[ f^* : \mathbb{R}^n \rightarrow \mathbb{R} : \pi \mapsto \sup_{x \in \mathbb{R}^n} \pi^\top x - f(x). \]  
(23)
Inspired by the example from , we introduce the coperspective function:

**Definition 3.** Let $f : \mathbb{R}^n \to \mathbb{R}$. The coperspective of $f$ is the perspective of the Fenchel conjugate, that is $(f^*)^\sim$, that we denote $f^\oplus$. In particular, for $\pi \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}^+$, we have

$$f^\oplus(\pi, \gamma) := \sup_{x \in \mathbb{R}^n} \pi^\top x - \gamma f(x). \quad (24)$$

**Remark 4.** The coperspective is jointly convex in $\pi, \gamma$, and lower semicontinuous. Moreover, it is a positively homogeneous function of degree 1:

$$f^\oplus(t \cdot \pi, t \cdot \gamma) = t \cdot f^\oplus(\pi, \gamma)$$

for $t > 0$.

**Remark 5.** Cuts for a convex function and its perspective are essentially equivalent. If $f(x) \geq f(x_0) + g^\top (x - x_0) = \theta + g^\top x$, then

$$\tilde{f}(x, t) = t \cdot f(x/t) \geq tf(x_0) + tg^\top (x/t - x_0) \geq tf(x_0) + g^\top (x - t \cdot x_0) \geq \theta \cdot t + g^\top x \quad (25)$$

Similarly, if $\tilde{f}(x, t) \geq \theta \cdot t + g^\top x + \beta$, then $f(x) \geq g^\top x + \theta + \beta$. Note that if the cut for $\tilde{f}$ is exact, we can assume $\beta = 0$.

### 4.2 Polyhedral risk measures

We now assume that the coherent risk measure $\rho$ is a polyhedral risk measure. More precisely, there is a set $Q = \text{conv}([q_1, \ldots, q_K])$ of probabilities such that, for any random variable $t$,

$$\rho[\mu] := \sup_{q \in Q} \mathbb{E}_q[\mu] = \max_{k \in [K]} \{ \mathbb{E}_{q_k}[\mu] \}, \quad (29)$$

which is homogeneous because $\rho$ is coherent. Assume that $\omega$ can take values $\omega_1, \ldots, \omega_J$, and that $Q_k[\omega = \omega_j] = q_{k,j}$. With this assumption, the Bellman operator $B(V)$ from (2) can be expressed by:

$$B(V)(x_0) = \inf_{x,y,z_0,t} z_0 \quad (30)$$

s.t.

- $z_0 \geq \sum_{j \in [J]} q_{k,j} \theta_j \forall k [\phi_k]$;
- $\theta_j \geq c_j^\top y_j + V(x_j) \forall j [\gamma_j]$;
- $A_jx_j + B_jx_0 + T_jy_j = d_j \forall j [\lambda_j]$;
- $x_j, y_j \geq 0 \forall j [\mu_j, \nu_j]$;

Now, we introduce dual multipliers for each one of the constraints: in order, $\phi_k, \gamma, \lambda, \mu, \nu$. This allows us to write:

$$B(V)(x_0) = \inf_{x,y,z_0,t} z_0 + \sup_{\phi_k,\gamma,\lambda,\mu,\nu} \sum_k \phi_k(\sum_{j \in [J]} q_{k,j} \theta_j - z_0) + \sum_{j \in [J]} \gamma_j(c_j^\top y_j + V(x_j) - \theta_j) + \sum_{j \in [J]} \lambda_j^\top (A_jx_j + B_jx_0 + T_jy_j - d_j) - \mu_j^\top x_j - \nu_j^\top y_j \quad (31)$$

s.t. $\phi \geq 0, \gamma \geq 0, \mu \geq 0, \nu \geq 0$. 

Interchanging \( \inf \) and \( \sup \) by strong linear programming duality, we get

\[
B(V)(x_0) = \sup_{\phi, \gamma, \lambda, \mu, \nu} \inf_x \sum_{j \in [J]} \gamma_j V(x_j) + \lambda_j^\top (Ax_j + B_j x_0 - d_j) - \mu_j^\top x_j \\
\text{s.t.} \quad \sum_k \phi_k = 1, \quad \phi_k \geq 0, \quad \forall j \quad [z_0] \\
\sum_k \phi_k q_{k,j} = \gamma_j, \quad \forall j \quad [\theta_j] \\
\gamma_j c_j + T_j^\top \lambda_j = \nu_j, \quad \forall j \quad [\eta_j] \\
\gamma_j \geq 0, \mu_j \geq 0, \nu_j \geq 0.
\]

(32)

Splitting everything out of the \( \inf \) that does not depend on \( x \), and transposing the linear operator \( A \):

\[
B(V)(x_0) = \sup_{\phi, \gamma, \lambda, \mu, \nu} \sum_{j \in [J]} \lambda_j^\top B_j x_0 - d_j + \inf_x \sum_{j \in [J]} \gamma_j V(x_j) + (A_j^\top \lambda_j - \mu_j)^\top x_j \\
\text{s.t.} \quad \sum_k \phi_k = 1, \quad \phi_k \geq 0, \quad \forall j \\
\sum_k \phi_k q_{k,j} = \gamma_j, \quad \forall j \\
\gamma_j c_j + T_j^\top \lambda_j = \nu_j, \quad \forall j \\
\gamma_j \geq 0, \mu_j \geq 0, \nu_j \geq 0.
\]

(33)

Note that the constraints on \( \phi_k \) imply that \( \gamma \) is a convex combination of the extreme probabilities \( q_k \). For conciseness, we denote by \( Q \) the polytope of probabilities corresponding to the dual representation of the risk measure.

Moreover,

\[
\inf_{x_j} \gamma_j V(x_j) + (A_j^\top \lambda_j - \mu_j)x_j = -\sup_{x_j} -\gamma_j V(x_j) - (A_j^\top \lambda_j - \mu_j)^\top x_j \\
= -V^\otimes(\mu_j - A_j^\top \lambda_j, \gamma)
\]

(34)

(35)

Therefore, the risk-averse linear Bellman operator \( B \) is also given by

\[
B(V)(x_0) = \sup_{\gamma, \lambda, \mu} \sum_{j \in [J]} \lambda_j^\top B_j x_0 - d_j - V^\otimes(\mu_j - A_j^\top \lambda_j, \gamma_j) \\
\text{s.t.} \quad \gamma \in Q, \quad \forall j \\
\gamma_j c_j + T_j^\top \lambda_j \geq 0, \quad \mu_j \geq 0.
\]

(36)

Now, we can evaluate the projective Fenchel dual of \( B(V) \) for \( \pi_0 \in \mathbb{R}^n \) and \( \gamma_0 > 0 \).

\[
B(V)^\otimes(\pi_0, \gamma_0) = \sup_{x_0} \pi_0^\top x_0 - \gamma_0 B(V)(x_0) \\
= \sup_{x_0} \pi_0^\top x_0 - \gamma_0 \cdot \sup_{\gamma, \lambda, \mu} \sum_{j \in [J]} \lambda_j^\top (B_j x_0 - d_j) - V^\otimes(\mu_j - A_j^\top \lambda_j, \gamma_j) \\
\text{s.t.} \quad \gamma \in Q, \quad \forall j \\
\gamma_j c_j + T_j^\top \lambda_j \geq 0, \quad \mu_j \geq 0.
\]

(37)

(38)

The maximization subproblem has an \textit{homogeneous} objective on its decision variables, since \( V^\otimes \) is homogeneous of degree one. Consequently, by scaling each variable by \( \gamma_0 > 0 \), and noting that
the constraints are positively homogeneouxs except $\gamma \in \mathbb{Q}$, we get

$$B(V)^{\mathbb{Q}}(\pi_0, \gamma_0) = \sup_{x_0} \pi_0^T x_0 + \inf_{\gamma, \lambda, \mu} \sum_{j \in [J]} \lambda_j^T (d_j - B_j x_0) + V^{\mathbb{Q}}(\mu_j - A_j \lambda_j, \gamma_j) \quad (39)$$

Interchanging sup and inf by strongly convex duality we get

$$B(V)^{\mathbb{Q}}(\pi_0, \gamma_0) = \inf_{\gamma, \lambda, \mu} \sum_{j \in [J]} \lambda_j^T d_j + V^{\mathbb{Q}}(\mu_j - A_j \lambda_j, \gamma_j) \quad (40)$$

which is a risk-neutral LBO that takes a homogeneous recourse function to another homogeneous convex function of the same dimension. This motivates the following

**Definition 6.** The projective dual Bellman operator associated to $B$ is the LBO

$$B^{\mathbb{Q}}(V^{\mathbb{Q}})(\pi_0, \gamma_0) = \inf_{\gamma, \lambda, \pi} \sum_{j \in [J]} \lambda_j^T d_j + V^{\mathbb{Q}}(\pi_j, \gamma_j) \quad (42)$$

Remark 7. If $V$ is polyhedral, so are its Fenchel dual and its perspective, so that one does not need convex duality for the purely polyhedral case of linear programming primal LBO’s and polyhedral risk measures.

The following proposition links the projective dual Bellman operator with the value of the dual problem.

**Proposition 8.** For $t \in [T]$, if the dual value function $D_t$ is defined by (19), and $V_t^t$ is the primal value function defined by (16), then we have

$$D_t = -V_t^t. \quad (43)$$

This proposition paves the way to a dual SDDP algorithm. Indeed it was shown in [LCC+20] that SDDP can be applied to any sequence of function linked through a linear Bellman operator (LBO) like $B^{\mathbb{Q}}$.  

12
4.2.1 Bounding the dual state

We have already seen in 3.2 how to obtain relatively complete recourse. To ensure convergence we still need to ensure that the state remains bounded through a compactification process.

Assuming relatively complete recourse, as well as explicit bounds, in the primal, we know that there exists an optimal primal solution. Further, by linear programming duality, we know that there exists an optimal dual solution. The marginal interpretation of the Lagrange multiplier $\pi$ (see Problem (17)) state that, for each node, the optimal dual $\pi_n$ is a subgradient of the primal value function $V_n$. Hence, assuming that $V_n$ is $L_t$-Lipschitz continuous on its domain, we can add the constraint $|\pi_m| \leq \gamma_m L_t + 1$ to (19) for each node $m$ of depth $t+1$, without changing its value. This method is similar to the compactification process through Lipschitz-regularization used in [LCC+20].

Finally we use the following compactified version of $B_\star$:

$$B_{t,L_t+1}^\square(Q)(\pi_0, \gamma_0) = \inf_{\gamma, \lambda, \pi, \xi, \zeta} \frac{\bar{\gamma}}{c} \sum_{j \in [J_t]} B_j^\top \lambda_j + \bar{y}_{t+1} \xi_j + Q(\pi_j, \gamma_j)$$

s.t.

$$\gamma \in \gamma_0 Q, \quad \gamma_j c_j + T_j^\top \lambda_j + \xi_j \geq 0 \quad \forall j$$

$$\sum_j B_j^\top \lambda_j + \zeta \geq \bar{\pi}_0$$

$$\pi_j + A_j^\top \lambda_j \geq 0 \quad \forall j \in [J_t]$$

$$|\pi_j| \leq \gamma_j L_{t+1} \quad \forall j \in [J_t]$$

$$\zeta \geq 0, \xi_j \geq 0$$

and we still have the Bellman’s recursion $V_t^\square = B_{t,L_t+1}^\square (V_{t+1})$. The sequence of LBOs $(B_{t,L_t+1}^\square)_{t \in [T]}$ being compact and compatible, the SDDP algorithm on this recursion converges. This is illustrated in the next section.

5 Examples

In this section, we provide an algorithm, in the lineage of SDDP, for the risk-averse dual problem given by the recursion (20). Then, we show the construction of the dual problem in the case of (nested) mean-AVAR risk measures, and close with two numerical examples.

5.1 A dual risk-averse algorithm

The recursion of (perspective) value functions $D_t$ given by (20) can be solved by recursively constructing piecewise-linear (upper) approximations, which we call $D_t$. As usual, one needs to ensure that the domain of the state variables $\pi_t$ and $\gamma_t$ remains bounded. Since all $\gamma_t$ remain in $[0, 1]$, we only need bounds for $\pi_t$, which we assume are given by the user. If one can estimate Lipschitz constants $L_t$ for the primal value functions $Q_t$, these can be used as $-\gamma_t L_t \leq \pi_t \leq \gamma_t L_T$. Moreover, one needs a starting upper bound for $D_t$. These can be obtained, for example, choosing $\pi_0 = 0$ and $\gamma_0 = 1$, and constructing cuts from $t = T - 1$ back to $t = t_0$.

As we observed in section 3.2, the first stage problem, corresponding to $t = t_0$, is slightly different. It is obtained as the fusion of the “zeroth stage”, equation (43), containing $\pi_{n_0}$ as a decision variable, and the first stage in (20). Furthermore, since $x_{n_0}$ is fixed, there’s no corresponding slack variable $\zeta_{n_0}$, so it must enforce the equality

$$\sum_{j \in [J_0]} B_j^\top \lambda_j = \pi_0$$

13
Instead of \((20d)\).

With this, we can now present how one can perform Bellman iterations on the recursion defined by \((20)\) to obtain convergence. We highlight the following differences with the standard ("primal") SDDP:

- The time decomposition does not allow for scenario decomposition, because of the linking constraint \(\zeta + \sum_{j \in [J_t]} B_j^T \lambda_j \geq \pi_t\), so we can use cuts calculated directly from the forward step, and not use a backward step;

- Again because we don’t have time decomposition, cuts have to be added for each scenario in the previous stage;

- In the forward step, we choose the scenario \(j\) according to a (smoothed) "importance sampling" procedure, giving it weight equal to \(\gamma_j + \varepsilon\);

- By homogeneity, we normalize the state variables \((\pi_j, \gamma_j)\) that will be used in the next step of the forward pass to have \(\gamma_{t+1} = 1\), unless we are in a branch where \(\gamma_t = 0\). This has had a positive impact in the numerical stability of the algorithm;

- Finally, by remark \([3]\) we ensure that, for every cut, its parameter \(\beta\) is always at least zero (there’s a sign change since we’re approximating concave functions from above).

**Algorithm 1:** Dual Risk-Averse SDDP

**Data:** upper bounds \(D_t^0 \geq D_t\) and bounds \(L_t\) for \(\pi_t\)

**Data:** maximum number of iterations \(N\) and smoothing parameter \(\varepsilon > 0\)

**Result:** upper bound on the value of problem \((20)\)

Set initial iteration \(k = 1\)

while \(k < N\) do

    Set \(t = t_0\)

    Solve the first stage problem to obtain \(\pi_t\), and \(\gamma_t = 1\), and update the current upper bound of the problem

    for \(t = t_0\) to \(T - 1\) do

        // forward pass

        Solve problem \((20)\) with \(D_{t+1}^{k-1}\) instead of \(D_{t+1}\)

        Evaluate a cut for \(D_t\) using the optimal multipliers for \(\pi_t\) and \(\gamma_t\), truncating the affine coefficient \(\beta\) if needed

        Choose a branch \(\hat{j}\) according to probabilities \(\gamma_j + \varepsilon\), where \(\gamma_j\) were obtained in the previous step

        if \(\gamma_{\hat{j}} > 0\) then

            Set \(\pi_{t+1} \leftarrow \pi_{\hat{j}} / \gamma_{\hat{j}}\), and \(\gamma_{t+1} \leftarrow 1\)

        else

            Set \(\pi_{t+1} \leftarrow \pi_{\hat{j}}\), and \(\gamma_{t+1} \leftarrow 0\)

        Increment \(k\)

    Return the upper bound

Naturally, one can couple this algorithm with (say) SDDP running on the primal. This keeps track of both upper and lower bounds, therefore allowing to stop based on a prescribed tolerance, instead of just a maximum number of iterations as described above.

Let us close this section with two remarks. First, even if this algorithm uses only forward passes, one could use backward passes for calculating cuts, as in the classical SDDP algorithm.
This would require solving approximately twice the number of optimization problems, but would include in the backward pass the updated value function, which could potentially speed up the convergence of the algorithm. Furthermore, this algorithm is easily amenable to standard cut-selection techniques, which can be useful to reduce the computational burden of each iteration.

5.2 Risk-averse duals from AVAR

Here, we show how to build the Bellman operator in the case where the risk measure \( \rho \) is a convex combination of the mean and the \( \alpha \)-AV@R, given by

\[
\rho [t] := \beta \mathbb{E}[t] + (1 - \beta) \operatorname{AV@R}_\alpha[t].
\] (46)

We observe that if \( \alpha = 1 \), then this risk measure coincides with the expectation, and if \( \alpha = \beta = 0 \) then it becomes the worst-case risk measure. The risk measures we employ in both examples in section 5.3 will be of this class.

With a finite number of scenarios, this is a polyhedral risk measure, since both the expectation and the AV@R are polyhedral in \( t \), yielding the following primal formulation for (2):

\[
\mathcal{B}(V)(x_0) = \inf_{x, y, z_0, t, u} \beta \mathbb{E}[c^\top y + V(x)] + (1 - \beta) \operatorname{AV@R}_\alpha[c^\top y + V(x)]
\]
\[
\quad \text{s.t. } \begin{cases} 
Ax + Bx_0 + Ty = d \\ x, y \geq 0 \end{cases} \quad \text{a.s.}
\] (47)

where we introduced the epigraphical auxiliary variables \( t \) and employed the polyhedral representation of AV@R. Notice that this formulation is different from the formulation in (29), since we do not describe the extreme points of the AV@R polyhedron (which may be very numerous!). Nevertheless, the reasoning used here will similarly lead to a formulation where we will recognize dual variables \( \gamma \) corresponding to probabilities.

We define dual multipliers for each one of the constraints: in order, \( \gamma, \delta, \lambda, \mu, \nu, \text{ and } \eta \).

With the expectation inner product, this yields the following Lagrangian:

\[
\mathcal{L} := (1 - \beta)z_0 + \beta \mathbb{E}[t] + \frac{1 - \beta}{\alpha} \mathbb{E}[u] + \mathbb{E}[\gamma(c^\top y + V(x) - t)] + \mathbb{E}[\delta(t - z_0 - u)] + \mathbb{E}[\lambda^\top (Ax + Bx_0 + Ty - d)] - \mathbb{E}[\mu^\top x + \nu^\top y + \eta u].
\]

Passing to the dual formulation, and eliminating the multipliers \( \mu, \nu \) and \( \eta \), this Lagrangian
corresponds to constraints on the dual formulation:

\[
B(V)(x_0) = \sup_{\lambda, \gamma, \delta, \mu} \mathbb{E} \left[ \lambda^\top (B x_0 - d) - \inf_x [(A^\top \lambda - \mu)^\top x + \gamma V(x)] \right] \tag{49}
\]

\[
\begin{align*}
\mathbb{E}[\delta] &= (1 - \beta) \\
0 &\leq \delta \leq \frac{1 - \beta}{\alpha} & \text{a.s.} \\
\gamma &= \beta + \delta & \text{a.s.} \\
\gamma c^\top + T^\top \lambda &\geq 0 & \text{a.s.} \\
\mu &\geq 0 & \text{a.s.}
\end{align*}
\]

Observe that the perspective variable \( \gamma \) represents the “change-of-measure” implied by the mean-AV@R combination. Indeed, \( \gamma \) is at least \( \beta \leq 1 \), and some events will have an increased contribution, up to \( \frac{1 - \beta}{\alpha} \), until this sums back to one, since \( \mathbb{E}[\delta] = 1 - \beta \).

5.3 Numerical experiments

We provide two numerical experiments for our algorithm. The first one is a problem with only one state variable, 5 control variables, 4 stages and 10 branches per stage. The dynamics is given by 2 equality constraints, that model inventory balance in time, and that production equals demand.

This is a natural setting where both state and control can be bounded a priori, and moreover production costs are always positive. We choose a risk measure of AV@R_{0.4}, that is, at each stage we average the 4 costlier scenario branches.

In Figure 1, we show the lower and upper bound convergence, as well as the evolution of the relative gap, given by \( \frac{\text{upper} - \text{lower}}{\text{lower}} \). We observe that, after 10 iterations, the gap has converged essentially to machine precision.

The second example comes from the Brazilian Hydrothermal Energy planning problem, where the reservoirs and hydro dams are aggregated into 4 subsystems, and there is a 5th node in the network, as an interconnection. Therefore, it contains 4 state variables (the stored energy in each reservoir), 9 equality constraints for the dynamics (4 for the states, and 5 for demand in each node), and a total of 164 control variables, accounting for hydro and thermal energy produced, and energy exchange among the nodes in the system. The uncertainty at each time step is the inflow for each aggregated reservoir, and is different for each time step, corresponding to different months of the year.
For this example, we take 8 stages and 40 inflows per stage. Again, we have natural bounds for every state variable, given by the reservoirs’ limits, and control variables (power output, line capacities, . . . ). The risk measure considered was a combination of expectation and AV@R, given by $0.7E + 0.3AV@R_{0.2}$.

In Figure 2, analogous to [1] we show the bound convergence and the evolution of the relative gap. We observe that, after 2000 iterations, the gap is still decreasing, having reached 0.6%.

Moreover, we remark that the primal algorithm performed these 2000 iterations in 24 minutes, while the dual algorithm needed 142 minutes for the same number of iterations. This is expected, since each problem in the dual formulation includes all 40 scenarios and a linking constraint among all of them, whereas the primal problem also allows decomposing each time step in separate problems for each scenario.

**References**


