Dual SDDP for risk-averse multistage stochastic programs

Bernardo Freitas Paulo da Costa[a], Vincent Leclère[b]

[a] *EMAp-FGV, Rio de Janeiro, Brazil*
[b] *CERMICS, Ecole des Ponts, Marne-la-Vallée, France*

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
2010 MSC: 90C15, 90C39, 49N15

1. Introduction

Multistage stochastic programming is a powerful framework with multiple applications [7], *e.g.* in the finance, energy and supply chain sectors. If the uncertainty is finitely supported, those problems can be seen as large-scale deterministic problems. When there is more than 4 or 5 stages, the deterministic equivalent is usually too large to be solved directly. One of the most successful paradigms in this setting consists in leveraging time-independance assumptions to derive Bellman equations [4]. The Stochastic Dual Dynamic Programming (SDDP) algorithm, and its numerous variants ([10, 3, 17, 1]), 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 minimize an expected cost, a large part of the recent litterature sparked by [14, 11, 16] has been devoted to efficiently introduce *risk aversion* in this framework, in particular inside the SDDP algorithm. Coherent risk measures [2] have become a usual tool to represent risk aversion in stochastic optimization problems. In multistage stochastic programming, minimizing a risk measure of the sum of costs leads to time-inconsistency. The easiest way to come up with a time-consistent risk-averse problem is to use composed Markovian risk measures [13], which, roughly speaking, means replacing the expectation by a risk measure inside the dynamic programming equation.

More precisely, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, and $\{\boldsymbol{\omega}_t\}_{t\in[T]}$ be a sequence of finitely supported, independent random variables (by convention, boldscript refers to random variables, normal script to an element of their support, and equalities between random variables hold almost surely.) We consider the following risk-averse multistage linear program (RA-MSLP)

$$\min_{\boldsymbol{x}_t,\boldsymbol{y}_t} \rho_1\Big(\boldsymbol{c}_1^\top \boldsymbol{y}_1 + \rho_{2|\boldsymbol{\omega}_1}\Big(\cdots + \rho_{T|\boldsymbol{\omega}_{[T-1]}}(\boldsymbol{c}_T^\top \boldsymbol{y}_T)\Big)\Big) \tag{1a}$$

$$\text{s.t. } \boldsymbol{A}_t\boldsymbol{x}_t + \boldsymbol{B}_t\boldsymbol{x}_{t-1} + \boldsymbol{T}_t\boldsymbol{y}_t = \boldsymbol{d}_t \quad \forall t\in[T] \tag{1b}$$

$$0 \le \boldsymbol{x}_t \le \bar{x}_t,\ 0 \le \boldsymbol{y}_t \le \bar{y}_t \quad \forall t\in[T] \tag{1c}$$

$$\boldsymbol{x}_t, \boldsymbol{y}_t \preceq \boldsymbol{\omega}_{[t]} \quad \forall t\in[T] \tag{1d}$$

where $\boldsymbol{\rho}_{t|\omega_{[t]}}$ is a coherent risk measure conditional on the past noises $\boldsymbol{\omega}_{[t]} := \{\boldsymbol{\omega}_1,\dots,\boldsymbol{\omega}_t\}$, all equalities hold almost surely, and constraint (1d) is the *non-anticipativity* constraint, stating that decisions $\boldsymbol{x}_t, \boldsymbol{y}_t$ are measurable with respect to $\boldsymbol{\omega}_{[t]} := \{\boldsymbol{\omega}_1,\dots,\boldsymbol{\omega}_t\}$. Convexity of ρ_t is crucial both for the SDDP algorithm and the duality theory developed here. Moreover, in this paper we restrict ourselves to polyhedral risk measures (defined in Section 2.2) to avoid dealing with technical constraint qualification considerations which would distract the reader.

Finally, note that, by construction, the nested multistage risk measure used in Problem (1) is time-consistent.

Since $\{\boldsymbol{\omega}_t\}_{t\in[T]}$ is a sequence of *independent* random variables, Dynamic Programming leads to the following recursion:

$$
\begin{aligned}
V_{T+1}(x_T) &= 0, && (2)\\
V_t(x) = \min_{\boldsymbol{x}_t,\boldsymbol{y}_t} \quad & \rho_t\left[\boldsymbol{c}_t^\top \boldsymbol{y}_t + V_{t+1}(\boldsymbol{x}_t)\right] && (3)\\
\text{s.t.} \quad & \boldsymbol{A}_t\boldsymbol{x}_t + \boldsymbol{B}_t x + \boldsymbol{T}_t\boldsymbol{y}_t = \boldsymbol{d}_t\\
& 0 \le \boldsymbol{x}_t \le \bar{x}_t,\ 0 \le \boldsymbol{y}_t \le \bar{y}_t
\end{aligned}
$$

where the value of Problem (1) is given by $V_1(x_0)$.

The classical SDDP algorithm builds outer approximations of the cost-to-go functions V_t, leading to exact lower bounds on the problem. In a risk-neutral framework, upper bounds can be estimated via Monte Carlo sampling. Unfortunately, it is unclear how to extend such statistical methods to the risk-averse setting [16]. Instead of statistical upper bounds, one can use exact upper bounds: Through backward recursion ([11]); by maintaining upper and lower bounds for all value functions ([3, 6]); or using Fenchel duality ([9, 8]). Up to now, the first approach has not been used to compute improving upper bounds along SDDP iterations, while the second approach relies on a *problem-child* node selection method. Finally, the last approach was developed only in a risk-neutral setting. The aim of this work is to adapt the latter approach to a risk-averse setting. By dualizing the extensive formulation of the risk-averse MLSP problem, and recognizing a time-decomposition, we obtain a Bellman recursion on which SDDP can be applied, yielding converging exact upper bounds.

Contributions. In this paper we i) derive a dual formulation of RA-MLSP with polyhedral risk measure; ii) show that it is time-decomposable and solvable through SDDP, yielding exact upper bounds of the original problem; iii) link the value function of the dual formulation with the co-perspective of the primal value function; and iv) illustrate the approach with numerical results.

2. Time decomposition of the dual of a risk averse MSLP

2.1. Risk-averse duals with AV@R

We start by showing how to build the dual problem in a very specific setting: for a single step of the recursion, with no upper bounds on $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, and when the risk measure ρ is a convex combination of the mean and the α-AV@R, given by, for $\alpha \in (0,1)$ and $\beta \in [0,1]$,

$$
\rho[\boldsymbol{\theta}] := \beta\mathbb{E}[\boldsymbol{\theta}] + (1-\beta)\,\mathrm{AV@R}_\alpha[\boldsymbol{\theta}]\,. \tag{4}
$$

This risk measure assumes an underlying probability for the scenarios, with respect to which one calculates the expectation and the AV@R. The risk measures we employ in the example in section 4.2 will be of this class.

We rewrite equation (3) using the Rockafellar-Uryasev representation of AV@R, with $\boldsymbol{\theta}$ as epigraphical variables for the scenario costs. For simplicity, we represent a random variable as a vector in $\mathbb{R}^J$, denoted with bold letters such as $\boldsymbol{x} = (x_1, \dots x_J)$, and the expectation $\mathbb{E}[\boldsymbol{x}]$ is the sum $\sum p_j x_j$. So, the value of $V_t(x_{t-1})$ is given by:

$$
\begin{aligned}
\inf_{\boldsymbol{x},\boldsymbol{y};q,\boldsymbol{\theta},\boldsymbol{u}} \quad & \beta\mathbb{E}[\boldsymbol{\theta}] + (1-\beta)\left[q + \tfrac{1}{\alpha}\mathbb{E}[\boldsymbol{u}]\right]\\
\text{s.t.} \quad & q + u_j \ge \theta_j && \forall j \in [J]\\
& \theta_j \ge c_j^\top y_j + V_t(x_j) && \forall j \in [J]\\
& A_j x_j + B_j x_{t-1} + T_j y_j = d_j && \forall j \in [J]\\
& x_j, y_j, u_j \ge 0 && \forall j \in [J]
\end{aligned} \tag{5}
$$

We define dual multipliers for every constraint: in order, $\boldsymbol{\delta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, $\boldsymbol{\nu}$, and $\boldsymbol{\eta}$. With the expectation inner product, this yields the Lagrangian:

$$
\begin{aligned}
&(1-\beta)q + \beta\mathbb{E}[\boldsymbol{\theta}] + (1-\beta)/\alpha\cdot\mathbb{E}[\boldsymbol{u}]\\
&\quad \mathbb{E}[\boldsymbol{\gamma}(\boldsymbol{c}^\top\boldsymbol{y} + V(\boldsymbol{x}) - \boldsymbol{\theta}) + \boldsymbol{\delta}(\boldsymbol{\theta} - q - \boldsymbol{u})]\\
&\qquad + \mathbb{E}[\boldsymbol{\lambda}^\top(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}x_{t-1} + \boldsymbol{T}\boldsymbol{y} - \boldsymbol{d})]\\
&\qquad\qquad - \mathbb{E}[\boldsymbol{\mu}^\top\boldsymbol{x} + \boldsymbol{\nu}^\top\boldsymbol{y} + \boldsymbol{\eta}\boldsymbol{u}]
\end{aligned}
$$

Eliminating the multipliers $\boldsymbol{\nu}$ and $\boldsymbol{\eta}$, we obtain the dual problem

$$
\begin{aligned}
\sup_{\boldsymbol{\lambda},\boldsymbol{\gamma},\boldsymbol{\delta},\boldsymbol{\mu}} \quad & \mathbb{E}\Big[\boldsymbol{\lambda}^\top(\boldsymbol{B}x_{t-1} - \boldsymbol{d}) + \\
& \quad \inf_{\boldsymbol{x}}\Big[(\boldsymbol{A}^\top\boldsymbol{\lambda} - \boldsymbol{\mu})^\top\boldsymbol{x} + \boldsymbol{\gamma}V_t(\boldsymbol{x})\Big]\Big]\\
\text{s.t.} \quad & \mathbb{E}[\boldsymbol{\delta}] = (1-\beta)\\
& 0 \le \delta_j \le \tfrac{1-\beta}{\alpha} && \forall j \in [J]\\
& \gamma_j = \beta + \delta_j && \forall j \in [J]\\
& \gamma_j c_j + T_j^\top\lambda_j \ge 0 && \forall j \in [J]\\
& \mu_j \ge 0 && \forall j \in [J]
\end{aligned} \tag{6}
$$

Observe that the variable $\boldsymbol{\gamma}$ represents the "change-of-measure" implied by the mean-AV@R combination [15]. Indeed, $\boldsymbol{\gamma}$ is at least $\beta \le 1$, and some events will have an increased contribution, up to $\frac{1-\beta}{\alpha}$, so that $\mathbb{E}[\boldsymbol{\gamma}] = 1$.

2.2. Polyhedral risk measures and duality

To extend the previous approach to more general risk measures, we adopt a distributionally robust point of view. We consider a *polyhedral risk measure* ρ, that is, a coherent risk measure of the form

$$\rho : \boldsymbol{t} \mapsto \sup_{\mathbb{Q}\in\mathcal{Q}} \mathbb{E}_{\mathbb{Q}}[\boldsymbol{t}] = \max_{k\in[K]} \{\mathbb{E}_{\mathbb{Q}^k}[\boldsymbol{t}]\}, \tag{7}$$

where $\mathcal{Q} = \mathrm{conv}(\{\mathbb{Q}^k\}_{k\in[K]})$. Polyhedral risk measures can be either chosen as interpretable risk-measures (e.g. AV@R in a finite setting) or as the worst case among a set of probabilities estimated by various experts. Since we don't assume a reference probability, we resort to describing the extremal risk measures, which may be very numerous. This also changes the interpretation of the dual variables $\boldsymbol{\gamma}$: now they correspond to supporting probabilities, instead of a change-of-measure.

We denote the elements of the support of $\boldsymbol{\omega}$ by $\omega_1, \dots, \omega_J$, and let $q_j^k := \mathbb{Q}^k[\boldsymbol{\omega} = \omega_j]$. Now, $V_t(x_{t-1})$ is given by:

$$\begin{aligned}
\inf_{\boldsymbol{x},\boldsymbol{y};z,\boldsymbol{\theta}} \quad & z & & \quad (8)\\
\text{s.t.} \quad & z \geq \sum_{j\in[J]} q_j^k \theta_j & \forall k \quad & [\phi_k]\\
& \theta_j \geq c_j^\top y_j + V_{t+1}(x_j) & \forall j \quad & [\gamma_j]\\
& A_j x_j + B_j x_{t-1} + T_j y_j = d_j & \forall j \quad & [\lambda_j]\\
& 0 \leq x_j \leq \bar{x}_t & \forall j \quad & [\mu_j, \zeta_j]\\
& 0 \leq y_j \leq \bar{y}_t & \forall j \quad & [\nu_j, \xi_j]
\end{aligned}$$

Proceeding analogously to the AV@R case above, we introduce dual multipliers as indicated in the brackets, and obtain the following dual problem

$$\begin{aligned}
\sup_{\substack{\phi_k,\gamma_j,\lambda_j,\\ \mu_j,\zeta_j,\xi_j}} \quad & \sum_{j\in[J]} \Big[\lambda_j^\top (B_j x_{t-1} - d_j) - \bar{x}_t \zeta_j - \bar{y}_t \xi_j \quad (9)\\
& \qquad + \inf_{x_j} (A_j^\top \lambda_j - \mu_j + \zeta_j)^\top x_j + \gamma_j V_{t+1}(x_j)\Big]\\
\text{s.t.} \quad & \sum_k \phi_k = 1, \qquad \phi_k \geq 0,\\
& \sum_k \phi_k q_j^k = \gamma_j \geq 0 & \forall j\\
& \gamma_j c_j + T_j^\top \lambda_j + \xi_j \geq 0 & \forall j\\
& \mu_j, \zeta_j, \xi_j \geq 0 & \forall j.
\end{aligned}$$

The constraints on ϕ_k are equivalent to describing the vector of γ_j's as a convex combination of the extreme probabilities $\mathbb{Q}^k$. Therefore, one can rewrite problem (9) to include the constraint $\{\gamma_j\}_{j\in[J]} \in \mathcal{Q}$ instead of the first two lines. This shows that the variables γ_j correspond to one supporting probability of the risk measure ρ. In particular, if a given scenario is *effective*, in the sense of [12], then there exists an optimal $\boldsymbol{\gamma}$ which charges this scenario.

Moreover, the last two constraints here correspond exactly to the last two in problem (6), which emphasizes the similarity between (9) and (6).

2.3. Multistage risk averse problem duality

We now extend the duality to the full multistage problem. In the stagewise independent setting, we let Ω_t be the set of all possible realizations of ω_t, and the risk measure ρ_t is defined by $\rho_t = \sup_{\mathbb{Q}\in\mathcal{Q}_t} \mathbb{E}_{\mathbb{Q}}[\cdot]$, for a polyhedral subset $\mathcal{Q}_t$ of probability measures on Ω_t. The tree $\mathcal{T}$ describing the stochastic process is such that each node n of depth t is associated with a possible value of $\boldsymbol{\omega}_{[t]} = (\boldsymbol{\omega}_1, \dots \boldsymbol{\omega}_t)$. For any node n, the set of its children is denoted by C_n, and $\mathcal{L}$ is the set of leaves of $\mathcal{T}$.

In the spirit of the previous section, we introduce variables z_n to stand for the risk-adjusted value of our problem starting from node n, and θ_m represents the cost-to-go following the branch of node $m \in C_n$. To reduce notational burden, we assume that, for all t, $\rho_t = \rho$. Then, the risk averse problem (1), with value $V_{n_0}(\tilde{x}_{n_0})$, can be written as the following linear program:

$$\begin{aligned}
\min \quad & z_0 & & \quad (10)\\
\text{s.t.} \quad & \sum_{m\in C_n} q_m^k \theta_m \leq z_n & \forall n, \forall k \in [K] \quad & [\Phi_n^k]\\
& c_m^\top y_m + z_m \leq \theta_m & \forall m \in \mathcal{T}\backslash\{n_0\} \quad & [\gamma_m]\\
& A_m x_m + B_m \tilde{x}_n & &\\
& \qquad + T_m y_m = d_m & \forall n, \forall m \in C_n \quad & [\lambda_m]\\
& z_\ell = 0 & \forall \ell \in \mathcal{L} \quad & [\eta_\ell]\\
& x_m = \tilde{x}_m & \forall m \in \mathcal{T}\backslash\{n_0\} \quad & [\pi_m]\\
& 0 \leq \tilde{x}_m \leq \bar{x}_m & \forall m \in \mathcal{T}\backslash\{n_0\} \quad & [\mu_m, \zeta_m]\\
& 0 \leq y_m \leq \bar{y}_m & \forall m \in \mathcal{T}\backslash\{n_0\} \quad & [\mu_m, \xi_m]
\end{aligned}$$

where, when unspecified, $\forall n$ stands for $\forall n \in \mathcal{T}\backslash\mathcal{L}$, $\tilde{x}_{n_0}$ is a parameter and not a variable, and we add the equalities $x_m = \tilde{x}_m$ to highlight the time dynamics.

Defining $\gamma_{n_0} = 1$, the linear programming dual of problem (10) is

$$\begin{aligned}
\sup_{\Phi,\gamma,\pi,\lambda} \quad & \pi_{n_0}^\top \tilde{x}_{n_0} - \sum_m \lambda_m^\top d_m + \bar{x}_m^\top \zeta_m + \bar{y}_m^\top \xi_m \\
\text{s.t.} \quad & \sum_{k\in[K]} \Phi_n^k = \gamma_n && \forall n && [z_n] \\
& \sum_{k\in[K]} \Phi_n^k q_m^k = \gamma_m \geq 0 && \forall n, \forall m \in C_n && [\theta_m] \\
& \pi_{n_0} = \sum_{m\in C_{n_0}} B_m^\top \lambda_m \\
& \pi_n \leq \zeta_n + \sum_{m\in C_n} B_m^\top \lambda_m && \forall n \in \mathcal{T}\backslash\{n_0\} && [\tilde{x}_m] \\
& \pi_m + A_m^\top \lambda_m = 0 && \forall m && [x_m] \\
& \gamma_m c_m + T_m^\top \lambda_m + \xi_m \geq 0 && \forall m && [y_m] \\
& \Phi_n^k \geq 0 && \forall n, \forall k \in [K] \\
& \zeta_m \geq 0, \xi_m \geq 0 && \forall m
\end{aligned}$$

where we keep $\forall n$ to imply $n \in \mathcal{T}\backslash\mathcal{L}$ as above, and unspecified $\forall m$, $\sum_m$ range over $m \in \mathcal{T}\backslash\{n_0\}$.

Note that Φ_n^k 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 $\ell \in \mathcal{L}$, $D_\ell(\pi_\ell, \gamma_\ell) = -\bar{x}_\ell^\top \max\{\pi_\ell, 0\}$, and for all nodes $n \in \mathcal{T}\backslash\mathcal{L}$, $D_n(\pi_n, \gamma_n)$ is given as the value of

$$\begin{aligned}
\sup_{\substack{\pi_m,\gamma_m,\lambda_m \\ \zeta_n,\xi_m\geq 0}} \quad & \mathbb{1}_{\{n=n_0\}} \pi_{n_0}^\top \tilde{x}_{n_0} - \bar{x}_n^\top \zeta_n + \\
& \sum_{m\in C_n} -\lambda_m^\top d_m - \bar{y}_m^\top \xi_m + D_m(\pi_m, \gamma_m) \\
\text{s.t.} \quad & (\gamma_m)_{m\in C_n} \in \gamma_n \mathcal{Q} \\
& \zeta_n + \sum_{m\in C_n} B_m^\top \lambda_m \geq \pi_n \\
& \pi_m + A_m^\top \lambda_m = 0, \quad \forall m \in C_n \\
& \gamma_m c_m + T_m^\top \lambda_m + \xi_m \geq 0, \quad \forall m \in C_n
\end{aligned} \tag{11}$$

By the independence assumption, a backward induction shows that $D_n = D_{n'}$ for all nodes n and n' of the same depth. Thus, defining $D_T(\pi_T, \gamma_T) = -\bar{x}_T^\top \max\{\pi_T, 0\}$, we obtain the following recursion for the dual value functions:

$$\begin{aligned}
& D_t(\pi_t, \gamma_t) = \\
& \sup_{\substack{\zeta,\gamma_j, \\ \lambda_j,\pi_j,\xi_j}} \quad -\bar{x}_t^\top \zeta + \sum_{j\in[J_t]} \Big[-d_j^\top \lambda_j - \bar{y}_{t+1}^\top \xi_j \\
& \qquad\qquad + D_{t+1}(\pi_j, \gamma_j) \Big] \\
& \text{s.t.} \quad (\gamma_j)_{j\in[J_t]} \in \gamma_t \mathcal{Q} \\
& \qquad \zeta + \sum_{j\in[J_t]} B_j^\top \lambda_j \geq \pi_t \\
& \qquad \pi_j + A_j^\top \lambda_j = 0 \qquad \forall j \in [J_t] \\
& \qquad \gamma_j c_j + T_j^\top \lambda_j + \xi_j \geq 0 \qquad \forall j \in [J_t] \\
& \qquad \xi_j \geq 0, \quad \zeta \geq 0
\end{aligned} \tag{12}$$

This decomposition satisfies the RCR conditions. Indeed, for every π_t and every $\gamma_t \geq 0$, any $\gamma \in \gamma_t \mathcal{Q}$ and $\lambda = 0$ are admissible, using slack variable ζ as needed. Then, π_j are given by the $\pi_j + A_j^\top \lambda_j = 0$, and the remaining constraints can be adjusted using ξ_j.

Remark 1. Relatively complete recourse in a dual formulation is not guaranteed (see for example [8]). In our setting, the explicit upper bounds of (1c) ensure RCR. The existence of such upper bounds is equivalent to the existence of exact penalization coefficients in the dual, which is the tool used in [8] to deal with this difficulty. Alternatively, we could incorporate feasibility cuts in the algorithm.

2.4. Bounding the dual state

With our boundedness assumption, we have relatively complete recourse in the dual. To prove convergence, we still need to ensure that the dual state remains bounded.

By assumption, 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 π (see Problem (10)) states that, for each node, the optimal dual π_n is a subgradient of the primal value function *for* $\gamma_n = 1$. In particular, π_n/γ_n can be bounded by the Lipschitz constant of the primal value function V_n. In the independent setting, assuming that V_t is L_t-Lipschitz continuous on its domain, we can add the constraint $|\pi_j| \leq \gamma_j L_{t+1}$ to (12) for each j, without changing its value. This method is similar to the compactification process through Lipschitz-regularization used in [9].

Therefore, we use the compactified recursion presented in (13). Since it has RCR and bounded states, the SDDP algorithm on this recursion converges. This is illustrated in section 4.

$$
\begin{array}{rll}
D_t(\pi_t, \gamma_t) = \displaystyle\sup_{\zeta,\gamma_j,\lambda_j,\pi_j,\xi_j} & -\bar{x}_t^\top \zeta + \displaystyle\sum_{j\in[J]} -d_j^\top \lambda_j - \bar{y}_{t+1}^\top \xi_j + D_{t+1}(\pi_j, \gamma_j) & \\
\text{s.t.} & \gamma \in \gamma_t \mathcal{Q} & \\
& \zeta + \sum_j B_j^\top \lambda_j \geq \pi_t & \\
& \pi_j + A_j^\top \lambda_j = 0 & \forall j \in [J_t] \\
& \gamma_j c_j + T_j^\top \lambda_j + \xi_j \geq 0 & \forall j \in [J_t] \\
& |\pi_j| \leq \gamma_j L_{t+1} & \forall j \in [J_t] \\
& \zeta \geq 0, \xi_j \geq 0 &
\end{array}
\tag{13}
$$

3. 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 2.

3.1. Homogeneous Fenchel duality

Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a proper lower semicontinuous convex function. Recall (see [5] for more details) that the *perspective function* of f, denoted $\tilde{f}$, is a convex, lower-semicontinuous function of $\mathbb{R}^{n+1}$, such that $\tilde{f}(x, \gamma) = \gamma f(x/\gamma)$ for any positive number γ.

Recall that the Fenchel conjugate of f is

$$
f^\star : \mathbb{R}^n \to \overline{\mathbb{R}} : \psi \mapsto \sup_{x\in\mathbb{R}^n} \psi^\top x - f(x). \tag{14}
$$

Inspired by the recurrences in (6) and (9), we introduce the *coperspective function*:

Definition 2. *Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$. The coperspective of f is the perspective of the Fenchel conjugate, that is $(f^\star)^\sim$, that we denote $f^\boxtimes$. In particular, for $\psi \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}_{++}$, we have*

$$
f^\boxtimes(\psi, \gamma) := \sup_{x\in\mathbb{R}^n} \psi^\top x - \gamma f(x). \tag{15}
$$

Remark 3. The coperspective is jointly convex in (ψ, γ), lower semicontinuous, and a positively homogeneous function of degree 1: for all $t > 0$,

$$
f^\boxtimes(t \cdot \psi, t \cdot \gamma) = t \cdot f^\boxtimes(\psi, \gamma).
$$

Remark 4. 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

$$
\begin{aligned}
\tilde{f}(x, \gamma) = \gamma \cdot f(x/\gamma) &\geq \gamma f(x_0) + \gamma g^\top(x/\gamma - x_0) \\
&\geq \gamma f(x_0) + g^\top(x - \gamma \cdot x_0) \\
&\geq \theta \cdot \gamma + g^\top x
\end{aligned}
$$

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

3.2. Duality and conjugate value functions

Consider a polyhedral risk measure ρ and the associated risk-averse Bellman operator $\mathcal{B}$ that, to any cost-to-go function V and initial state x_{t-1} associates the value of Problem (8).

The coperspective of $\mathcal{B}(V)$ can be calculated using (9). Leveraging positive homogeneity, for $\psi_0 \in \mathbb{R}^n$ and $\gamma_0 > 0$, we get that $\mathcal{B}(V)^\boxtimes(\psi_0, \gamma_0)$ is given by

$$
\begin{aligned}
&\sup_{x_0}\ \psi_0^\top x_0 \\
&\quad + \inf_{\substack{\gamma,\lambda,\mu\\ \zeta,\xi}} \sum_{j\in[J]} \lambda_j^\top (d_j - B_j x_0) + \xi_j^\top \bar{y}_{t+1} \\
&\qquad\qquad + \zeta_j^\top \bar{x}_{t+1} + V^\boxtimes(\mu_j - A_j^\top \lambda_j - \zeta_j, \gamma_j) \\
&\quad\ \text{s.t. } \gamma \in \gamma_0 \mathcal{Q} \\
&\qquad\ \gamma_j c_j + \xi_j + T_j^\top \lambda_j \geq 0 \qquad \forall j \\
&\qquad\ \mu_j, \zeta_j, \xi_j \geq 0 \qquad \forall j.
\end{aligned}
\tag{16}
$$

Note that, if V is polyhedral, so are its Fenchel dual and its perspective. Thus, by linear programming duality, we can interchange sup and inf to obtain

$$
\begin{aligned}
&[\mathcal{B}(V)]^{\boxtimes}(\psi_0, \gamma_0) = \qquad (17)\\
&\inf_{\substack{\gamma,\lambda\\ \mu,\zeta,\xi}} \quad \sum_{j\in[J]} \lambda_j^\top d_j + \xi_j^\top \bar{y}_{t+1} + \zeta_j^\top \bar{x}_{t+1} + V^{\boxtimes}(\psi_j, \gamma_j)\\
&\quad \text{s.t.} \quad \textstyle\sum_j B_j^\top \lambda_j = \psi_0\\
&\qquad\quad \gamma \in \gamma_0 \mathcal{Q}\\
&\qquad\quad \gamma_j c_j + \xi_j + T_j^\top \lambda_j \geq 0 \quad \forall j\\
&\qquad\quad \psi_j = \mu_j - A_j^\top \lambda_j - \zeta_j \quad \forall j.
\end{aligned}
$$

This equation defines a risk-*neutral* LBO $\mathcal{B}^{\boxtimes}$ that takes a *homogeneous* recourse function $V^{\boxtimes}$ and returns another homogeneous convex function of the same dimension. We call this operator the *projective dual Bellman operator* associated to $\mathcal{B}$.

Comparing (12) and (17), we notice the decomposition is not done at the same time-step for all variables: in the first one, ζ is a single variable, relaxing the incoming dual state constraint; whereas in the second, it relaxes the outgoing dual state constraint. Substituting $\pi_j = \psi_j + \zeta_j - \mu_j$, we obtain the following proposition, linking the coperspectives of the primal value functions with the value functions of the dual problem.

Proposition 5. *For $t \in [T]$, if the dual value function D_t is defined by* (12)*, and V_t is the primal value function defined by* (3) *then*

$$
D_t(\pi_t, \gamma_t) = -\inf_{\substack{\zeta_t + \psi_t \geq \pi_t\\ \zeta_t \geq 0}} \bar{x}_t^\top \zeta_t + V_t^{\boxtimes}(\psi_t, \gamma_t).
$$

In particular, D_t is a concave, positively homogeneous, one-sided Lipschitz regularization of $V_t^{\boxtimes}$.

Further, the value of primal Problem (1) *is* $\sup_{\pi_0} \quad \pi_0^\top x_0 + D_0(\pi_0, 1)$.

This proposition paves the way to a dual SDDP algorithm. Indeed, it was shown in [9] that SDDP can be applied to any sequence of functions linked through linear Bellman operators (LBO) like $\mathcal{B}^{\boxtimes}$.

4. Examples

In this section, we provide an algorithm, in the lineage of SDDP, for the risk-averse dual problem given by the recursion (13). Then, we close with one numerical example from a real-world problem. A more comprehensive discussion on the algorithm, including implementation details, can be found at the companion. There, one will also find further results on the application of our algorithm.

4.1. A dual risk-averse algorithm

The recursion of (perspective) value functions D_t given by (13) can be solved by recursively constructing piecewise linear (upper) approximations, which we call $\mathfrak{D}_t$. As usual, one needs to ensure that the domain of the state variables π_t and γ_t remains bounded. Since all γ_t remain in $[0, 1]$, we only need bounds for π_t, which we assume are given by the user as the Lipschitz constants L_t for the primal value functions V_t. In our experiments, the Lipschitz constant estimation was not critical: Increasing L_t by a factor 10 or 100 had a negligible impact after 50 iterations, as can be seen in section C of the companion. Moreover, one needs a starting upper bound for $\mathfrak{D}_t$. These can be obtained, for example, choosing $\pi_t = 0$ and $\gamma_t = 1$, and constructing cuts from $t = T - 1$ back to $t = t_0$.

The first stage problem, corresponding to $t = t_0$, is slightly different. It is obtained as the fusion of the "zero-th stage" containing π_{n_0} as a decision variable, and the first stage in (13). Furthermore, since x_{n_0} is fixed, there's no corresponding slack variables μ_{n_0} and ζ_{n_0}, so it must satisfy

$$
\sum_{j\in[J_0]} B_j^\top \lambda_j = \pi_0. \qquad (18)
$$

With this, we can now present how one can perform Bellman iterations on the recursion defined by (13) to obtain convergence. We highlight the following differences with the primal SDDP:

- Computing $D_t(\pi, \gamma)$ cannot be decomposed by realization of $\boldsymbol{\omega}_t$ due to the coupling constraint $\zeta + \sum_{j\in[J_t]} B_j^\top \lambda_j \geq \pi_t$. In particular, the forward pass is as demanding as the backward pass, and yields cuts. Furthermore, we have one next-state variable per possible realization of $\boldsymbol{\omega}_t$, which means that, when adding a single cut to the approximation of D_{t+1}, we are adding J_t constraints.

- In the forward step, we choose the realization j according to a (smoothed) "importance sampling" procedure, with weight $\gamma_j + \varepsilon$.

- By homogeneity, we normalize the state variables (π_j, γ_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 4, we ensure that, for every cut, its parameter β is always zero.

Algorithm 1: Dual Risk-Averse SDDP

Data: upper bounds $\mathfrak{D}_t^0 \geq D_t$ and bounds L_t for $|\pi_t|$
Result: upper bound on the value of (13)

for $k = 0$ **to** N **do**
    Solve the first stage problem to obtain π_0, and set $\gamma_0 = 1$
    if $k == N$ **then** Return upper bound

    for $t = 0$ **to** $T - 1$ **do** `// forward pass`
        Solve problem (13) with $\mathfrak{D}_{t+1}^k$ instead of D_{t+1}
        Compute a cut for D_t using the optimal multipliers for π_t and γ_t
        Choose a branch $\hat{\jmath}$ according to probabilities $\gamma_j + \varepsilon$
        if $\gamma_{\hat{\jmath}} > 0$ **then**
            Set $\pi_{t+1} \leftarrow \pi_{\hat{\jmath}}/\gamma_{\hat{\jmath}}$, and $\gamma_{t+1} \leftarrow 1$
        else
            Set $\pi_{t+1} \leftarrow \pi_{\hat{\jmath}}$, and $\gamma_{t+1} \leftarrow 0$

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 computing 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.

4.2. Numerical experiments

We present here a numerical example. Further details and other results are given in the companion, and the implementation in julia, along with other examples, can be found at `https://github.com/bfpc/DualSDDP.jl`.

This 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 12 stages and 82 inflow realizations per stage (thus 82^{12} scenarios). 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 $\beta\mathbb{E} + (1-\beta)\text{AV@R}_\alpha$. In this problem, the highest marginal cost is given by load shedding, which yields estimates for the Lipschitz constants we use.

In Figure 1, we present the evolution of the bounds obtained by the primal SDDP, our dual SDDP algorithm, as well the one shot backward bounds of [11] (Philpott UB), computed every 50 iterations based on the trajectories from primal SDDP, and the upper and lower bounds provided by the problem-child method of [3] (Baucke UB / LB). This is done for various level of risk aversion. Note that, on this problem, the dual upper bound always outperform the problem-child method. It also slightly beat the primal one-shot upper bound in the most risk-averse case. This is also observed on the other numerical experiments available at `https://github.com/bfpc/DualSDDP.jl`.

Finally, we noticed that each iteration of the dual is between 30 and 15 times slower than primal iteration, being larger for higher branching sizes.

# branches	P-SDDP	D-SDDP	Problem Child
10	0.023	0.166	0.109
20	0.054	0.523	0.224
40	0.113	2.366	0.402
80	0.274	5.739	0.813

Table 1: Single iteration time in sec (around $it = 100$)

This is expected, since each problem in the dual formulation includes all inflow realizations and a linking constraint among all of them, whereas the primal problem also allows decomposing each time step in separate problems for each branch.

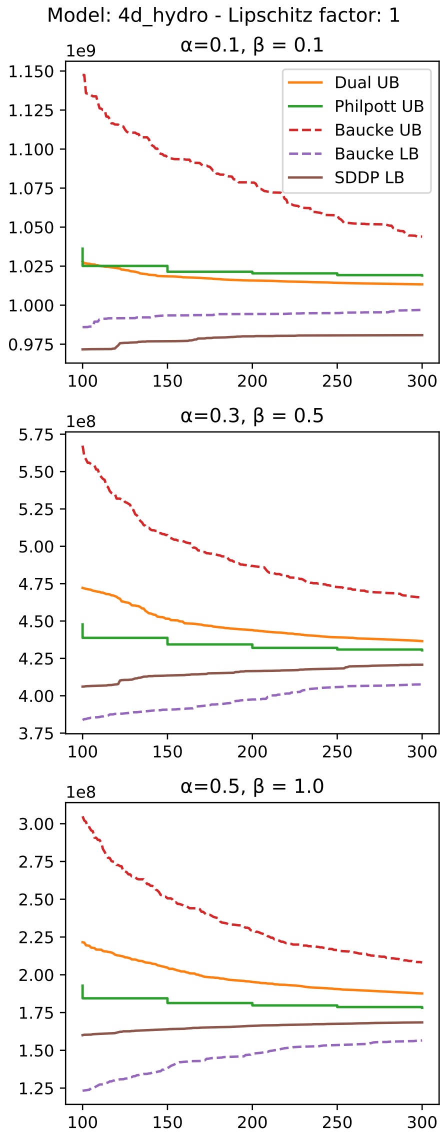

Figure 1: Bounds evolution for hydrothermal problem.

Acknowledgements. We would like to thank the editor and an anonymous referee for their suggestions which improved the presentation of our results. We also thank Lucas Merabet for his comments.

The first author is partly supported by project COPPETEC-23145. The second author benefited from the support of FMJH-PGMO and from EDF.

References

[1] Shabbir Ahmed, Filipe Goulart Cabral, and Bernardo Freitas Paulo da Costa. Stochastic lipschitz dynamic programming. *Mathematical Programming*, pages 1–39, 2020.

[2] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

[3] Regan Baucke, Anthony Downward, and Golbon Zakeri. A deterministic algorithm for solving multistage stochastic programming problems. *Optimization Online*, 2017.

[4] Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1 & 2. Athena Scientific Belmont, MA, 3rd edition, 2005.

[5] Patrick L Combettes. Perspective functions: Properties, constructions, and examples. *Set-Valued and Variational Analysis*, 26(2):247–264, 2018.

[6] Anthony Downward, Oscar Dowson, and Regan Baucke. Stochastic dual dynamic programming with stagewise-dependent objective uncertainty. *Operations Research Letters*, 48(1):33–39, 2020.

[7] Horand Gassmann and William T Ziemba. *Stochastic Programming: applications in finance, energy, planning and logistics*, volume 4. World Scientific, 2013.

[8] Vincent Guigues, Alexander Shapiro, and Yi Cheng. Duality and sensitivity analysis of multistage linear stochastic programs. *arXiv preprint arXiv:1911.07080*, 2019.

[9] Vincent Leclère, Pierre Carpentier, Jean-Philippe Chancelier, Arnaud Lenoir, and François Pacaud. Exact converging bounds for stochastic dual dynamic programming via fenchel duality. *SIAM Journal on Optimization*, 30(2):1223–1250, 2020.

[10] Mario VF Pereira and Leontina MVG Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, 52(1-3):359–375, 1991.

[11] Andrew Philpott, Vitor de Matos, and Erlon Finardi. On solving multistage stochastic programs with coherent risk measures. *Operations Research*, 61(4):957–970, 2013.

[12] Hamed Rahimian, Güzin Bayraksan, and Tito Homem-de Mello. Identifying effective scenarios in distributionally robust stochastic programs with total variation distance. *Mathematical Programming*, 173(1):393–430, 2019.

[13] Andrzej Ruszczyński. Risk-averse dynamic programming for markov decision processes. *Mathematical programming*, 125(2):235–261, 2010.

[14] Alexander Shapiro. Minimax and risk averse multistage stochastic programming. *European Journal of Operational Research*, 219(3):719–726, 2012.

[15] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.

[16] Alexander Shapiro, Wajdi Tekaya, Joari Paulo da Costa, and Murilo P Soares. Risk neutral and risk averse stochastic dual dynamic programming method. *European Journal of Operational Research*, 224(2):375–391, 2013.

[17] Jikai Zou, Shabbir Ahmed, and Xu Andy Sun. Stochastic dual dynamic integer programming. *Mathematical Programming*, 175(1):461–502, 2019.