SDDP FOR SOME INTERSTAGE DEPENDENT RISK-AVERSE PROBLEMS AND APPLICATION TO HYDRO-THERMAL PLANNING

VINCENT GUIGUES

Abstract. We consider interstage dependent stochastic linear programs where both the random right-hand side and the model of the underlying stochastic process have a special structure. Namely, for equality constraints (resp. inequality constraints) the right-hand side is an affine function (resp. a given function b_t) of the process value for the current time step t. As for m-th component of the process at time step t, it depends on previous values of the process through a function h_{tm}.

For this type of problem, to obtain an approximate policy under some assumptions for functions b_t and h_{tm}, we detail a stochastic dual dynamic programming algorithm. Our analysis includes some enhancements of this algorithm such as the definition of a state vector of minimal size, the computation of feasibility cuts without the assumption of relatively complete recourse, as well as efficient formulas for sharing optimality and feasibility cuts between nodes of the same stage. The algorithm is given for both a non-risk-averse and a risk-averse model. We finally provide preliminary results comparing the performances of the recourse functions corresponding to these two models for a real-life application.

AMS subject classifications: 90C15, 91B30.

1. Introduction

The use of decomposition methods for solving linear multi-stage stochastic programs dates back to the nested decomposition (ND) algorithm [2]. This method assumes that the number of realizations of the process over the optimization period is finite (these realizations can be organized in a finite scenario tree). At each iteration and in each node of the scenario tree, the algorithm updates lower bounding approximations for the corresponding recourse functions. However, for many applications, the number of scenarios is so large that this method entails prohibitive computational efforts. Monte Carlo sampling-based algorithms constitute an interesting alternative in such situations. One of these algorithms adapted for multistage stochastic linear programs whose number of immediate descendant nodes is small but with many stages [13] consists in sampling in the forward pass of the ND. This sampling-based variant of the ND is the so-called Stochastic Dual Dynamic Programming (SDDP) algorithm.

To our knowledge, this algorithm has been described so far with the assumption of relatively complete recourse for stochastic linear programs where the right-hand side is an affine function of the process values. Moreover, in general, it is assumed that the process is stagewise independent or that it affinely depends on previous values. In this paper, we detail the SDDP algorithm for a larger class of problems where relatively complete recourse does not hold. More precisely, we consider a

Key words and phrases. Stochastic programming and Risk-averse optimization and Decomposition algorithms and Interstage dependency and Monte Carlo sampling.

feasible T-stage stochastic optimization problem of form

$$
(1) \qquad \left\{ \begin{array}{lll} \inf \ \mathbb{E}[\sum_{t=1}^{T} f_t(x_t)] & & \\ A_t x_t \geq b_t(\xi_t) - B_t x_{t-1}, \ \text{a.s.}, & t = 1, \ldots, T, & \texttt{INEQ} \\ C_t x_t = D_t \xi_t - E_t x_{t-1}, \ \text{a.s.}, & t = 1, \ldots, T, & \texttt{EQ} \\ x_t \geq 0, \ \text{a.s.}, \ x_t \in L_p(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^{k_t}), & t = 1, \ldots, T, & \end{array} \right.
$$

where x_0 is given, (ξ_t) is an interstage dependent stochastic process with natural filtration $\mathcal{F}_t = \sigma(\xi_1, \ldots, \xi_t)$, f_t is a polyhedral cost function:

$$
(2) \qquad f_t(x_t) = \left\{ \begin{array}{ll} \max\limits_{1 \leq j \leq J_t} \alpha_{tj} + x_t^\top \beta_{tj} & \text{if } x_t \in X_t = \{x \,:\, c_{tk}^\top x \leq d_{tk}, k = 1, \ldots, K_t\}, \\ +\infty & \text{otherwise,} \end{array} \right.
$$

and $b_t(x) = (b_{t1}(x), \ldots, b_{t\ell_t}(x))^\top$ for given functions $b_{ti} : \mathbb{R}^M \to \mathbb{R}$.

Regarding the stochastic process, we assume that each component $\xi_t(m)$ is a general function of past values, i.e., for every $m = 1, \ldots, M$, and $t \in \mathbb{Z}$, we have

$$
(3) \qquad \xi_t(m) = h_{tm}(\xi_{t-1}(m), \ldots, \xi_{t-p_t(m)}(m), \eta_t(m))
$$

for some lag $p_t(m) \in \mathbb{N}$ and some function $h_{tm} : \mathbb{R}^{p_t(m)+1} \to \mathbb{R}$, where (η_t) is an interstage independent process (for any stage t, correlations between the components of η_t are however allowed).[1] The need to consider this more general framework is motivated by some applications; see Examples 2.1 and 2.2 below for instance. In that context, to preserve the convexity of the recourse functions, either (i) the functions h_{tm} are affine or (ii) satisfy some assumptions given in Section 2 and there are no equality constraints. In case (i), the right-hand side of equality constraints must be an affine function of the process value to preserve the convexity of the recourse functions.

In our interstage dependent context (3), the recourse functions depend on some past realizations of the process. We define for each time step t a vector $\xi_{[t]}$ containing the minimal number of past realizations needed to implement the SDDP algorithm. Next, under some assumptions on functions b_t and h_{tm} that guarantee the convexity of the recourse functions for (1), we provide formulas for the cuts that are built in the backward pass of the SDDP algorithm to approximate these recourse functions. Such cuts can be shared between nodes of the same stage. For an interstage dependent process with affine functions h_{tm}, this was first observed in [10]. However, when each component $(\xi_t(m))$ is a generalized autoregressive process (of form (8) below), the formulas we obtain for the cuts in Corollary 2.5 can be in some cases (depending on the application) more economic (in terms of memory allocation) compared to those in [10]. Moreover, since we do not assume relatively complete recourse, we also provide formulas for feasibility cuts that are needed to build sequences of feasible states in the forward pass of the algorithm. We show that in our statistical framework, these cuts can also be shared between nodes of the same stage. To the best of our knowledge, the description of the SDDP algorithm in the general framework (1) for processes satisfying (3) has not been done so far. When relatively complete recourse does not hold and when the underlying stochastic process is interstage dependent, we are also not aware of a previous work explaining how to build and share feasibility cuts (in the forward pass of the SDDP algorithm) between nodes of the same stage.

Next, we consider a risk-averse formulation of (1) using a multiperiod risk measure proposed in [5], [6] that allows us to apply SDDP to approximate the corresponding risk-averse recourse functions. For the class of problems considered in this paper, we provide formulas for the cuts built in this risk-averse version of SDDP.

Finally, we provide a first set of numerical results that compares the use of the aforementioned risk-averse and non-risk-averse recourse functions for the mid-term Brazilian hydro-thermal planning problem.

[1] If all lags $p_t(m)$ are null, we recover the case when process (ξ_t) is interstage independent.

The outline of the paper is as follows. In Section 2, we study the non-risk-averse version of SDDP while Section 3 considers the risk-averse case. Numerical results are reported in Section 4.

The paper is quite technical. The reader can in a first reading skip the theorems, propositions, and their corollaries and focus on the examples to get the main ideas. However, the theorems and propositions provide formulas for the cuts that will be useful to the practionner interested in implementing the SDDP algorithm for the type of problems we consider. The proofs are collected in the appendix.

We start setting some notation:

- e is a column vector of all ones whose dimension may vary upon the context;
- If A is an $m_1 \times n$ matrix and B an $m_2 \times n$ matrix, $(A; B)$ denotes the $(m_1 + m_2) \times n$ matrix $\begin{pmatrix} A \\ B \end{pmatrix}$;
- I_n is the $n \times n$ identity matrix and $0_{m\times n}$ is an $m \times n$ matrix of zeros;
- For real numbers $x_1, \ldots, x_n$, we denote by $\mathrm{Diag}(x_1, \ldots, x_n)$ the $n \times n$ diagonal matrix whose entry at position (i, i) is x_i;
- For a continuous random variable X representing a cost, the Conditional Value-at-Risk of level $\varepsilon \in [0, 1]$ of X [16] is given by $\mathrm{CVaR}^{\varepsilon}(X) := \mathbb{E}[X|X \geq F_X^{-1}(1 - \varepsilon)]$, where $F_X(\cdot)$ is the cumulative distribution function (CDF) of X;
- For $t_2 \geq t_1$, the short form $v_{t_1:t_2}$ stands for the concatenation $(v_{t_1}, v_{t_1+1}, \ldots, v_{t_2})$;
- $\mathcal{Q}_{t+1}$ denotes a (generic) recourse function used at time step $t = 1, \ldots, T$, i.e., $\mathcal{Q}_{T+1} \equiv 0$ and if $t < T$ then $\mathcal{Q}_{t+1}(x_t, \xi_{[t]})$ represents a cost over the period $t + 1, \ldots, T$. Various recourse functions at t will be defined using the same notation $\mathcal{Q}_{t+1}$. Which $\mathcal{Q}_{t+1}$ is relevant will be clear from the context.

As is usually done in the stochastic programming (SP) literature and to alleviate notation, we use the same notation for a random variable and for a particular realization of this random variable, the context allowing us to know which concept is being referred to.

2. SDDP for a class of non-risk-averse interstage dependent stochastic programs

In its risk-neutral version, SDDP aims at providing approximations of the recourse functions for problem (1). These recourse functions $\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}), t = 1, \ldots, T$, satisfy the dynamic programming (DP) relations

$$[LP_t] \quad \mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}) = \mathbb{E}_{\xi_t|\xi_{[t-1]}} \left[\begin{array}{l} \inf_{x_t} f_t(x_t) + \mathcal{Q}_{t+1}(x_t, \xi_{[t]}) \\ A_t x_t \geq b_t(\xi_t) - B_t x_{t-1} \\ C_t x_t = D_t \xi_t - E_t x_{t-1} \\ x_t \geq 0 \end{array} \right] \tag{4}$$

for $t = 1, \ldots, T$, with $\mathcal{Q}_{T+1} \equiv 0$. In the above relations, $\xi_{[t]}$ denotes the available and useful history of the process at time step t; see Section 2.2 for details.

A solution $(x_1(\cdot), \ldots, x_T(\cdot))$ of (1) is called a policy. Such policy is nonanticipative, i.e., $x_t(\cdot)$ is a function of available realizations at time step t. Using the approximate recourse functions obtained with SDDP, we obtain an approximate policy for (1).

We will assume that at the beginning of the optimization period, the realizations of $\xi_j, j \leq 1$, are available.

We make the following assumptions:

(A1) The function b_{ti} is convex for every $t = 1, \ldots, T$, and $i = 1, \ldots, \ell_t$.

(A2) The support Ω_t of the distribution of η_t in (3) is discrete and finite:

$$\Omega_t = \{\eta_{tj},\;\; j=1,\ldots,q_t<\infty\} \text{ with } \eta_{tj}\in\mathbb{R}^M \text{ and } \mathbb{P}(\eta_t=\eta_{tj})=p(t,j)>0. \tag{5}$$

(A3) For $t=2,\ldots,T$, for every $(t-1)$-stage scenario $(\xi_1,\xi_2,\ldots,\xi_{t-1})$, and for every state x_{t-1} feasible on this scenario for stage $t-1$, the set

$$\{x_t \;:\; A_t x_t \geq b_t(\xi_{tj}) - B_t x_{t-1},\; C_t x_t = D_t \xi_{tj} - E_t x_{t-1},\; x_t \geq 0\}$$

is nonempty and bounded for every $j=1,\ldots,q_t$ where the vector $\xi_{tj}\in\mathbb{R}^M$ is given by

$$\xi_{tj}(m) = h_{tm}(\xi_{t-1}(m),\ldots,\xi_{t-p_t(m)}(m),\eta_{tj}(m)),\; m=1,\ldots,M. \tag{6}$$

Assumption (A3) holds, in particular, if problem (1) is feasible and if at each stage, all decision variables are bounded, almost surely. Such is the case of the real-life application we consider in Section 4.

We will consider two special classes of processes referred to as the *convex process model* and the *affine process model* in the sequel. More precisely, in the case of the convex process model, we assume the following:

(A4) For every $t=1,\ldots,T$, and $i=1,\ldots,\ell_t$, for every $x,y\in\mathbb{R}^M$ such that $x\geq y$, we have $b_{ti}(x)\geq b_{ti}(y)$.
(A5) For every $m=1,\ldots,M$, and $t\in\mathbb{Z}$, relation (3) holds for some lag $p_t(m)\in\mathbb{N}$, some convex function $h_{tm}:\mathbb{R}^{p_t(m)+1}\to\mathbb{R}$, where (η_t) is an interstage independent process.
(A6) For every $t=1,\ldots,T$, and $m=1,\ldots,M$, for every $x,y\in\mathbb{R}^{p_t(m)+1}$ such that $x\geq y$, function $h_{tm}(x)$ from Assumption (A5) satisfies $h_{tm}(x)\geq h_{tm}(y)$.

If there are no equality constraints in (1), i.e., if constraints `EQ` are absent, our results will be derived making Assumptions (A1), (A2), (A3), (A4), (A5), and (A6) which guarantee, in particular, the convexity of recourse functions $\mathcal{Q}_t(\cdot)$ from (4). These assumptions as well as problem structure (1) have been used to model various applications.

Example 2.1 (Production management)**.** *Consider a production management problem aiming at minimizing the expected production cost where the system uncertainty is captured by demand $\mathcal{D}_t(m)$ in period t for type of client or geographical zone m. In this context, we have $\xi_t=(\mathcal{D}_t(1),\ldots,\mathcal{D}_t(M))^\top$ and demand satisfaction constraints can be written as* `INEQ` *with $b_{ti}(x_1,\ldots,x_M)=x_i,\; i=1,\ldots,\ell_t=M$ satisfying Assumptions (A1) and (A4). Since demand realizations are positive, instead of an affine function for h_{tm} one may prefer a model formulated as*

$$\xi_t(m) = \mathcal{D}_t(m) = f(\sum_{j=1}^{p_t(m)} \phi_t^j(m)\mathcal{D}_{t-j}(m) + \eta_t(m)) = h_{tm}(\xi_{t-1}(m),\ldots,\xi_{t-p_t(m)}(m),\eta_t(m)) \tag{7}$$

where f is a positive valued function: $f:\mathbb{R}\to\mathbb{R}_+$; which ensures positivity of demands for any distribution of noises η_t. As an example, taking for f the functions $f(x)=\max(x,0)$ or $f(x)=\exp(x)$, the corresponding functions h_{tm} in (7) given by $h_{tm}(x_1,\ldots,x_{p_t(m)+1})=\max(0,\sum_{j=1}^{p_t(m)}\phi_t^j(m)x_j+x_{p_t(m)+1})$ or $h_{tm}(x_1,\ldots,x_{p_t(m)+1})=\exp(\sum_{j=1}^{p_t(m)}\phi_t^j(m)x_j+x_{p_t(m)+1})$ are convex, i.e., Assumption (A5) holds. Moreover, in these cases, if all coefficients $\phi_t^j(m)$ are nonnegative, Assumption (A6) also holds. The max operator for f above can provide a model to obtain positive inflows for the application described in Example 2.2 below.

Assumption (A5) states that at stage t, m-th component of the process value depends on $p_t(m)$ previous values of this component through a convex function h_{tm}. As a special case, we will consider the affine process model where this function h_{tm} is affine:

(A7) For every $m = 1, \ldots, M$, and $t \in \mathbb{Z}$, we have

$$\frac{\xi_t(m) - \mu_t(m)}{\sigma_t(m)} = \sum_{j=1}^{p_t(m)} \phi_t^j(m) \left(\frac{\xi_{t-j}(m) - \mu_{t-j}(m)}{\sigma_{t-j}(m)} \right) + \eta_t(m) \tag{8}$$

where $\mu_t(m) = \mathbb{E}[\xi_t(m)]$, $\sigma_t^2(m) = Var[\xi_t(m)]$, lag $p_t(m) \in \mathbb{N}$, $\phi_t^{p_t(m)}(m) \neq 0$, and where (η_t) is an interstage independent process (for any stage t, correlations between the components of η_t are however allowed).[2]

In this context, Assumption (A4) will not be needed and subsequent developments hold under Assumptions (A1), (A2), (A3), and (A7). An example of a problem that can be modeled as (1) where Assumptions (A1), (A2), and (A7) hold is the hydro-thermal planning problem described in [8]. We recall in Example 2.2 which follows the uncertain constraints of a simplified version of this problem.

Example 2.2 (Hydro-thermal planning)**.** *We have NS subsystems, each subsystem i containing an hydroplant, with hydro generation $u_t(i)$ for time step t, and its water reservoir. We denote by $V_t(i)$ the volume of this reservoir at the end of time step t. Such volume depends on the volume of the reservoir at the end of the previous period, on the turbined outflow, and on inflows $\mathcal{I}_t(i)$ in subsystem i for period t. The corresponding dynamics is given by*

$$V_t(i) = V_{t-1}(i) - u_t(i) + \gamma_t(i)\mathcal{I}_t(i),$$

where $\gamma_t(i) \in (0,1)$ is the portion of inflows that comes to the reservoir; the remaining portion being directly converted into energy by run-of-river plants. However, due to limits in the run-of-river capacities, not all these inflows may be converted into energy. The corresponding losses are modelled by some convex loss function $\mathcal{L}_t$ in such a way that for each subsystem i and time step t, demand satisfaction constraints write

$$u_t(i) + df_t(i) + (1 - \gamma_t(i))\,\mathcal{I}_t(i) - \mathcal{L}_t\left((1 - \gamma_t(i))\mathcal{I}_t(i)\right) \geq \mathcal{D}_t(i)$$

where $\mathcal{D}_t(i)$ (resp. $df_t(i)$) denotes the demand (resp. unsatisfied demand).[3] *Setting $x_t = (V_t(1), \ldots, V_t(NS), u_t(1), \ldots, u_t(NS), df_t(1), \ldots, df_t(NS))^\top$ and $\xi_t = (\mathcal{I}_t(1), \ldots, \mathcal{I}_t(NS), \mathcal{D}_t(1), \ldots, \mathcal{D}_t(NS))^\top$, we see that these constraints can be written as* `EQ` *and* `INEQ` *with*

$$\begin{array}{ll} C_t = [I_{NS}, I_{NS}, 0_{NS \times NS}], & E_t = [-I_{NS}, 0_{NS \times 2NS}], \\ D_t = [\mathrm{Diag}(\gamma_t(1), \ldots, \gamma_t(NS)), 0_{NS \times NS}], & A_t = [0_{NS \times NS}, I_{NS}, I_{NS}], \end{array}$$

$B_t = 0$, and where b_{ti} is the convex function

$$b_{ti}(x_1, \ldots, x_{2NS}) = x_{NS+i} - (1 - \gamma_t(i))x_i + \mathcal{L}_t((1 - \gamma_t(i))x_i), \text{ for } i = 1, \ldots, \ell_t = NS.$$

Moreover, for this problem, the process of inflows is commonly modeled by a Periodic Autoregressive (PAR) process of form (8)*, see* [9], [7], [11] *for instance. As a result, assuming also a PAR process for the demand in each subsystem, Assumption (A7) is satisfied and ξ_t has $M = 2NS$ components. The corresponding approximations of the recourse functions* (4) *are obtained discretizing the distributions of noises η_t. In this context, Assumption (A2) also holds.*

Our main results will be illustrated using simple hydro-thermal problems.

For didactic reasons, we start our developments describing in the next subsection the SDDP algorithm in a simplified framework: we consider a problem of form (1) without equality constraints,

---

[2] The generalized autoregressive model is written using normalized random variables. For numerical reasons, it is recommended to use such formulation for the calibration of the model.

[3] Exchanges between subsystems can also be considered. Demand satisfaction constraints can be written as `INEQ` in this case too.

with f_t linear, h_{tm} affine, b_t is a max function, and we assume relatively complete recourse. The next subsections consider the general case (problem of form (1)).

2.1. **SDDP for some interstage dependent problems.** In this subsection, we consider the case where there are no equality constraints, $b_t(\xi_t) = \max(\xi_t, 0)$, $f_t(x_t) = c_t^\top x_t$ is linear, and h_{tm} is affine. More precisely, (ξ_t) is a (PAR) process of the form (3):

$$\xi_{2t} = \frac{1}{2}\xi_{2t-1} + \eta_{2t} \text{ and } \xi_{2t-1} = \frac{1}{3}(\xi_{2t-2} + \xi_{2t-3} + \xi_{2t-4}) + \eta_{2t-1}, \quad \forall\, t \in \mathbb{Z}, \tag{9}$$

and we will assume, without loss of generality, that T is odd. We also assume in that subsection that relatively complete recourse holds. As a result, the recourse functions satisfy

$$[LP_t] \quad \mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}) = \mathbb{E}_{\xi_t|\xi_{[t-1]}} \left[\begin{array}{l} \inf_{x_t} c_t^\top x_t + \mathcal{Q}_{t+1}(x_t, \xi_{[t]}) \\ A_t x_t \geq \max(\xi_t, 0) - B_t x_{t-1} \\ x_t \geq 0 \end{array} \right] \tag{10}$$

for $t = 2, \ldots, T$, with $\mathcal{Q}_{T+1} \equiv 0$ while the optimization problem (1) can be written

$$[LP_1] \quad \begin{array}{l} \inf_{x_1} c_1^\top x_1 + \mathcal{Q}_2(x_1, \xi_{[1]}) \\ A_1 x_1 \geq \max(\xi_1, 0) - B_1 x_0 \\ x_1 \geq 0. \end{array}$$

One may think at first sight that since ξ_t depends on 1 or 3 past values of the process, depending if t is even or odd, it would be sufficient to store 1 (resp. 3) past realizations in the state vector for stage t if t is even (resp. odd). In fact, this choice does not define an appropriate state vector. Indeed, let us look at DP equations (10). Since they are written backward in time, the state vectors are also defined backward in time. Recalling that $\mathcal{Q}_{T+1} \equiv 0$, $\xi_{[T-1]}$ gathers the realizations of the process before stage T upon which ξ_T depends. Since, T is odd, ξ_T depends on 3 past realizations ξ_{T-1}, ξ_{T-2}, and ξ_{T-3}. It follows that $\xi_{[T-1]} = (\xi_{T-1}, \xi_{T-2}, \xi_{T-3})$. Looking again at DP equations (10), now with $t = T-1$, we see that $\xi_{[T-2]}$ should gather the terms in $\xi_{[T-1]}$ that correspond to realizations of the process before stage $T-1$ (these terms are ξ_{T-2} and ξ_{T-3}) and the realizations of the process before stage $T-1$ upon which ξ_{T-1} depends (since $T-1$ is even, this realization is ξ_{T-2}). It follows that $\xi_{[T-2]} = (\xi_{T-2}, \xi_{T-3})$. Proceeding iteratively, we obtain

$$\xi_{[t]} = \begin{cases} (\xi_t, \xi_{t-1}, \xi_{t-2}) & \text{if } t \text{ is even,} \\ (\xi_t, \xi_{t-1}) & \text{if } t \text{ is odd.} \end{cases} \tag{11}$$

The SDDP algorithm exploits the convexity of the recourse functions to build lower bounding approximations of these functions. At iteration i, a convex polyhedral lower bounding approximate function $\mathcal{Q}_t^i$ is built for the convex recourse function $\mathcal{Q}_t$:

$$\mathcal{Q}_t^i(x_{t-1}, \xi_{[t-1]}) = \max_{j=0,1,\ldots,iH} [-E_{t-1}^j x_{t-1} + \tilde{E}_{t-1}^j \xi_{[t-1]} + e_{t-1}^j]$$

where H is the number of cuts (hyperplanes lying below the recourse function) computed for $\mathcal{Q}_t$ at each iteration (see below).

For convenience, we denote by $\overrightarrow{E}_{t-1}^{\,i}$ (resp. $\overrightarrow{\tilde{E}}_{t-1}^{\,i}$ and $\overrightarrow{e}_{t-1}^{\,i}$) the matrix whose $(j+1)$-th line (for $j = 0, \ldots, iH$) is the row vector E_{t-1}^j (resp. $\tilde{E}_{t-1}^j$ and e_{t-1}^j).[4]

Given $(x_0, \xi_{[1]})$, in each iteration $i = 1, 2, \ldots$, a forward pass computes H feasible states $(x_t^k, \xi_{[t]}^k)$, $k = (i-1)H+1, \ldots, iH$, for time steps $t = 1, \ldots, T$, as follows. Given $\xi_{[1]}$, the scenarios $(\eta_2^k, \ldots, \eta_T^k)$, $k = (i-1)H+1, \ldots, iH$ are sampled from the distribution of $(\eta_2, \ldots, \eta_T)$. These scenarios induce the scenarios $(\xi_1^k, \ldots, \xi_T^k)$, $k = (i-1)H+1, \ldots, iH$ via (9) as well as $(\xi_{[1]}^k, \ldots, \xi_{[T]}^k)$ via (11)

[4] We use notation from [3] and [15]

(recall that since ξ_1 is known, $\xi_1^k = \xi_1$ for all scenario k). The forward pass then computes x_t^k solving the following approximate problem $[AP_t^{i,k}]$ of $[LP_t]$ for $t = 1, \ldots, T$, and $k = (i-1)H + 1, \ldots, iH$, obtained replacing $\mathcal{Q}_{t+1}$ by $\mathcal{Q}_{t+1}^{i-1}$:

$$[AP_t^{i,k}] \quad \mathcal{Q}_t^i(x_{t-1}^k, \xi_{[t-1]}^k) = \left\{ \begin{array}{l} \inf\limits_{x_t^k, \theta_t^k} c_t^\top x_t^k + \theta_t^k \\ A_t x_t^k \geq \max(\xi_t^k, 0) - B_t x_{t-1}^k \\ \overrightarrow{E}_t^{i-1} x_t^k + \theta_t^k e \geq \overrightarrow{\tilde{E}}_t^{i-1} \xi_{[t]}^k + \overrightarrow{e}_t^{i-1} \quad (a) \\ x_t^k \geq 0, \end{array} \right.$$

with $x_0^k = x_0$. In the above expression, constraints (a) are optimality cuts. Note that, for $t = 1$, since $[AP_t^{i,k}] = [AP_1^{i,k}]$ does not depend on k (because $\xi_1^k = \xi_1$ and $\xi_{[1]}^k = \xi_{[1]}$ for all k), we can write $[AP_1^i]$ instead of $[AP_1^{i,k}]$.

A backward pass then builds H cuts for each recourse function $\mathcal{Q}_t$ at $(x_{t-1}^k, \xi_{[t-1]}^k), k = (i-1)H + 1, \ldots, iH$. As a result, the forward pass consists of obtaining decisions on H scenarios replacing the recourse functions in $[LP_t]$ by the lower bounding approximations of these recourse functions obtained in the end of the backward pass of the previous iteration. A lower bound on $\mathcal{Q}_t(x_{t-1}^k, \xi_{[t-1]}^k)$ is obtained on scenario k and iteration i solving problem $[AP_t^{i,k}]$. In particular, a lower bound on the optimal value of $[LP_1]$ is given by the optimal value of $[AP_1^i]$.

In [15], conditions are given which guarantee the convergence of a set of sampling-based decomposition algorithms that include SDDP. Under such conditions, an optimal solution to $[AP_1^i]$ converges with probability one to an optimal solution to $[LP_1]$ in a finite number of iterations.

We now detail the computations of optimality cuts in the backward pass. The optimality cuts are computed for time step $T+1$ down to time step 2. For time step $T+1$, since $\mathcal{Q}_{T+1}^i = \mathcal{Q}_{T+1} = 0$, we have for $\mathcal{Q}_{T+1}$ the cuts $E_T^k = \tilde{E}_T^k = e_T^k = 0$ for $k = (i-1)H + 1, \ldots, iH$. At time step $t = 2, \ldots, T$, the cuts for $\mathcal{Q}_t$ are computed having at hand the approximation $\mathcal{Q}_{t+1}^i$ of $\mathcal{Q}_{t+1}$ which satisfies

$$\mathcal{Q}_{t+1}(x_t, \xi_{[t]}) \geq \mathcal{Q}_{t+1}^i(x_t, \xi_{[t]}). \tag{12}$$

The above relation indeed holds for $t = T$. Assuming that it holds for some $t \in \{2, \ldots, T\}$, we build $\mathcal{Q}_t^i$ as follows, in such a way that (12) holds with $t+1$ substituted by t.

First, it is convenient to skip from conditional expectations to unconditional ones computed with respect to the distributions of $\eta_t, t = 2, \ldots, T$. This can be done expressing $\xi_{[t]}$ and ξ_t as a function of η_t and $\xi_{[t-1]}$ using equations (9) and state vectors (11): we obtain

$$\xi_t = \Phi_t \xi_{[t-1]} + \eta_t \quad \text{and} \quad \xi_{[t]} = \tilde{\Phi}_t \xi_{[t-1]} + \tilde{\Psi}_t \eta_t \tag{13}$$

where

$$\left\{ \begin{array}{ll} \Phi_{2t} = \left(\frac{1}{2} I_M \ \ 0_{M \times M} \right), & \Phi_{2t-1} = \left(\frac{1}{3} I_M \ \ \frac{1}{3} I_M \ \ \frac{1}{3} I_M \right), \\ \tilde{\Phi}_{2t} = \begin{pmatrix} \frac{1}{2} I_M & 0_{M \times M} \\ I_M & 0_{M \times M} \\ 0_{M \times M} & I_M \end{pmatrix}, & \tilde{\Phi}_{2t-1} = \begin{pmatrix} \frac{1}{3} I_M & \frac{1}{3} I_M & \frac{1}{3} I_M \\ I_M & 0_{M \times M} & 0_{M \times M} \end{pmatrix}, \\ \tilde{\Psi}_{2t} = \begin{bmatrix} I_M \\ 0_{M \times M} \\ 0_{M \times M} \end{bmatrix}, & \tilde{\Psi}_{2t-1} = \begin{bmatrix} I_M \\ 0_{M \times M} \end{bmatrix}, \end{array} \right.$$

for every positive integer t.

Plugging (12) into (10) and using (13), we obtain $\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}) \geq \mathbb{E}_{\eta_t}\left[\mathcal{Q}_t^i(x_{t-1}, \xi_{[t-1]}, \eta_t)\right]$ with

$$
\mathcal{Q}_t^i(x_{t-1}, \xi_{[t-1]}, \eta_t) = \left\{ \begin{array}{l} \inf\limits_{x_t, \theta_t} \; c_t^\top + \theta_t \\ A_t x_t \geq \max(\Phi_t \xi_{[t-1]} + \eta_t, 0) - B_t x_{t-1} \\ \overrightarrow{E}_t^i x_t + \theta_t e \geq \overrightarrow{E}_t^i \left(\tilde{\Phi}_t \xi_{[t-1]} + \tilde{\Psi}_t \eta_t\right) + \overrightarrow{e}_t^i \\ x_t \geq 0. \end{array} \right. \tag{14}
$$

On scenario k and time step t, the above problem is solved for $(x_{t-1}, \xi_{[t-1]}, \eta_t) = (x_{t-1}^k, \xi_{[t-1]}^k, \eta_{tj})$, $j = 1, \ldots, q_t$. Since Assumption (A3) holds, the optimal values of these linear programs are finite and both the primal and the dual have the same optimal value. We denote by π_{t2}^{kj} and ρ_t^{kj} the (row vectors) optimal Lagrange multipliers associated to respectively the first and second group of constraints for problem $\mathcal{Q}_t^i(x_{t-1}^k, \xi_{[t-1]}^k, \eta_{tj})$.

Next, for any $x, x_0 \in \mathbb{R}^M$, let $s(x_0)$ be a matrix such that

$$
\max(x, 0) \geq \max(x_0, 0) + s(x_0)(x - x_0),
$$

i.e., the transpose of the m-th row of $s(x_0)$ is a subgradient of the function $\max(x(m), 0)$ at $x_0(m)$.

Setting

$$
\xi_{tj}^k = \Phi_t \xi_{[t-1]}^k + \eta_{tj},
$$

for $j = 1, \ldots, q_t$, the following cuts are computed for $\mathcal{Q}_t$ at iteration i in the backward pass:

$$
\begin{array}{lcl}
E_{t-1}^k & = & \sum_{j=1}^{q_t} p(t,j) \pi_{t2}^{kj} B_t, \\
\tilde{E}_{t-1}^k & = & \sum_{j=1}^{q_t} p(t,j) \left[\rho_t^{kj} \overrightarrow{E}_t^i \tilde{\Phi}_t + \pi_{t2}^{kj} s(\xi_{tj}^k) \Phi_t \right], \\
e_{t-1}^k & = & \sum_{j=1}^{q_t} p(t,j) \left[\rho_t^{kj} (\overrightarrow{e}_t^i + \overrightarrow{E}_t^i \tilde{\Psi}_t \eta_{tj}) + \pi_{t2}^{kj} \left(\max(\xi_{tj}^k, 0) - s(\xi_{tj}^k) \Phi_t \xi_{[t-1]}^k \right) \right],
\end{array}
$$

for $t = 2, \ldots, T$, $k = (i-1)H + 1, \ldots, iH$.

Finally, after each backward pass (run after a forward pass), a stopping test, discussed in Section 2.6, is called for. We now consider problems of the form (1) and start defining the relevant history $\xi_{[t]}$ of the process to be included in the state vector in our interstage dependent framework.

2.2. **State vector definition.** For simplicity, in the SP literature, the state vector $\xi_{[t]}$ involved in (4) is in general either not specified or $\xi_{[t]} = (\xi_1, \ldots, \xi_{t-1})$ is chosen ([17], [18] for instance). However, for some processes and time steps, this history may not be enough, or, on the contrary, may be too rich. It is important, especially for algorithmic purposes, to keep track of all the necessary history of the process but also to try and find the "minimal" history of the process that needs to be included in the state vectors for a stochastic problem of form (1) and an underlying stochastic process (ξ_t) satisfying Assumption (A5).

The construction of this state vector is first illustrated on a small example depicted in Figure 1. In this example, ξ_t has only one component and there are $T = 8$ stages. From top to bottom, the different graphs in this figure illustrate the dependence of respectively ξ_8, ξ_7, ξ_6, and ξ_5 with respect to previous values. More precisely, ξ_8 depends on ξ_5, ξ_6, and ξ_7; ξ_7 only depends on ξ_6; ξ_6 depends on ξ_4 and ξ_5 while ξ_5 depends on ξ_2, ξ_3, and ξ_4. Recalling that $\xi_{[t-1]}$ is an argument of $\mathcal{Q}_t$ (see DP equations (4)) and that $\mathcal{Q}_9$ is null; considering (4) written for $t = 8$, we see that $\xi_{[7]}$ gathers realizations of the process upon which ξ_8 depends, i.e., (ξ_5, ξ_6, ξ_7). Considering equation (4) written for $t = 7$, we see that $\xi_{[6]}$ appearing as an argument of $\mathcal{Q}_7$ not only needs to contain past values of the process upon which ξ_7 depends but also the values in $\xi_{[7]}$ (appearing in the objective function) corresponding to time steps lower than or equal to 6. Considering Figure 1, among the arrows starting at the current time step or at future time steps $j > 7$, we look at the one which reaches the smallest past time step. As a result, $\xi_{[6]} = (\xi_5, \xi_6)$. Reasoning similarly and going backward in time, we obtain $\xi_{[5]} = (\xi_4, \xi_5)$ and $\xi_{[4]} = (\xi_2, \xi_3, \xi_4)$.

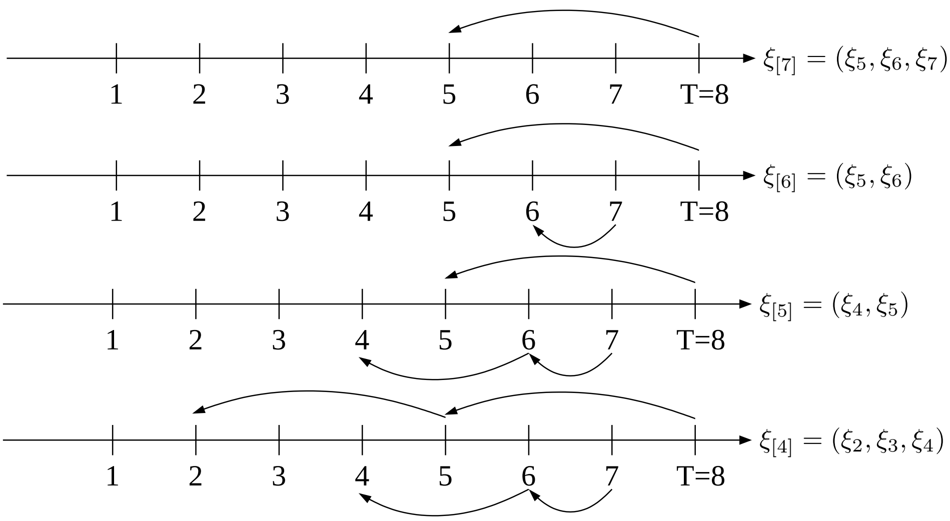

Figure 1. State vector definition on a simple example.

Let us consider as another example the hydro-thermal application of Section 4 where the lags for the PAR models for each subsystem are given in Table 1. We see that the lags range from 1 to 4 for the South-East and South subsystems and from 1 to 5 for the North-East and North subsystems. It is thus sufficient to use a state vector that stores 4 past inflow realizations for the South-East and South subsystems and 5 past inflow realizations for the North-East and North subsystems. However, this choice is not optimal. Let us see what minimal information is needed. One may think at first sight that since $\xi_t(m)$ depends on $p_t(m)$ past values, it would be sufficient, together with x_{t-1}, to store $p_t(m)$ past realizations of component m in the state vector for stage t. In fact, it is not difficult to see that this does not define in general an appropriate state vector. To convince us, let us see what minimal information is needed in that example for the South-East subsystem if the optimization horizon is a civil year (January-December) with monthly time step. Since the DP equations are written backward in time, we start with the last stage and since the lag for December is $Lag_{December} = 4$, we see that we need $Size_{December} = 4$ past realizations for December. For November, the lag is $Lag_{November} = 1$ so we need at least $Lag_{November} = 1$ past realization for November. Moreover, the cost-to-go function $\mathcal{Q}_{t+1}$ used for November takes as argument the state vector for December. Since this argument contains the realizations of inflows for November, October, September, and August (recall that $Size_{December} = 4$), we need at least $Size_{December} - 1 = 3$ past realizations for November (the realizations for October, September, and August). Gathering our observations, we see that we need to store for November $Size_{November} = \max(Size_{December} - 1, Lag_{November}) = 3$ past realizations. Similarly, we need to store for October $Size_{October} = \max(Size_{November} - 1, Lag_{October}) = \max(2, 3) = 3$ past realizations. Reasoning similarly, we obtain the minimal number of past realizations to include in the state vector for all stages and subsystems reported in Table 2. From this table, we see that the state vector for stage t can need more than $p_t(m)$ past realizations of inflows for component m.

From these examples, we see that in the general case, vector $\xi_{[t]}$ should gather the realizations of the process up to time t upon which depend $\xi_{t+1}, \xi_{t+2}, \ldots, \xi_T$ (see DP equations (4)). For time step

Subsystem	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
South-East	1	1	1	2	3	1	3	1	1	3	1	4
South	1	1	1	1	1	1	4	1	1	1	1	1
North-East	5	2	1	1	1	1	2	1	3	3	2	5
North	1	4	1	1	2	1	3	2	5	3	5	1

TABLE 1. Lags for the PAR models of inflows for the South-East, South, North-East, and North subsystems. Column i gives the lags for the i-th month of the year: for instance the lags for January are 1, 1, 5, and 1 for respectively the South-East, South, North-East, and North subsystems.

Subsystem	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
South-East	1	1	1	2	3	2	3	1	2	3	3	4
South	1	1	1	1	2	3	4	1	1	1	1	1
North-East	5	2	1	1	1	1	2	2	3	3	4	5
North	3	4	1	1	2	2	3	4	5	4	5	1

TABLE 2. Minimal number of past realizations to include in the state vector for the South-East, South, North-East, and North subsystems when the optimization horizon is a civil year (January-December). Column i gives this number for the i-th month of the year: for instance the number of past realizations to include in the state vector for January are 1, 1, 5, and 3 for respectively the South-East, South, North-East, and North subsystems.

$T-1$, these realizations are $\xi_{[T-1]} = (\xi_{T-1}(m),\, \ldots, \xi_{T-p_T(m)}(m), m=1,\ldots,M)$. For time step $T-2$, the values of the process upon which depend ξ_{T-1} and ξ_T are $(\xi_{t-1}(m), \ldots, \xi_{t-p_t(m)}(m))$, $m=1,\ldots,M$, for $t = T-1, T$. As a result, we have $\xi_{[T-2]} = (\xi_{T-2}(m), \ldots, \xi_{T-1-s_{T-1,m}}(m), m = 1,\ldots,M)$, with $T-1-s_{T-1,m} = \min(T-1-p_{T-1}(m), T-p_T(m))$. Proceeding iteratively, we see that $\xi_{[t-1]}$ should gather $(\xi_{t-1}(m), \xi_{t-2}(m), \ldots,\, \xi_{t-s_{t,m}}(m), m=1,\ldots,M)$, with $t - s_{t,m} = \min_{t \leq w \leq T}(w - p_w(m))$, i.e., $s_{t,m} = \max_{0 \leq w \leq T-t}(p_{t+w}(m) \, - w)$. The coefficients $s_{t,m}$ can also be defined iteratively by $s_{T+1,m} = -\infty$ and $s_{t,m} = \max(p_t(m),\, s_{t+1,m}-1),\ t=1,\ldots,T$. Finally, the state vector at time step $t+1$ is given by $(x_t^\top, \xi_{[t]}^\top)^\top$ with

$$\xi_{[t]}\Big(\sum_{j=1}^{m-1} s_{t+1,j} + k\Big) = \xi_{t+1-k}(m) \text{ for } m=1,\ldots,M, \text{ and } k = 1,\ldots,s_{t+1,m}. \tag{15}$$

2.3. **General overview of SDDP for problem** (1)**.** As mentionned in Section 2.1, the SDDP algorithm exploits the convexity of the recourse functions to build lower bounding approximations of these functions. This latter property holds under conditions given in the following lemma:

Lemma 2.3 (Convexity of recourse functions)**.** *Consider recourse functions defined by* (4)*. In each of the two situations below, these recourse functions are convex:*

(i) *Assumptions (A1) and (A7) hold;*
(ii) *for every time step, there are no equality constraints (matrices C_t, D_t, and E_t are absent) and Assumptions (A1), (A4), and (A5) hold.*

Note that for the class of problems we consider and in contrast with the framework usually considered to apply SDDP (see [13] and [15] for instance), our recourse functions are not in general polyhedral. However, at iteration i, a convex polyhedral lower bounding approximate function $\mathcal{Q}_t^i$ can still be built for the convex recourse function $\mathcal{Q}_t$:

$$\mathcal{Q}_t^i(x_{t-1}, \xi_{[t-1]}) = \max_{j=0,1,\ldots,iH} [-E_{t-1}^j x_{t-1} + \tilde{E}_{t-1}^j \xi_{[t-1]} + e_{t-1}^j]$$

where, as in Section 2.1, H is the number of cuts (hyperplanes lying below the recourse function) computed for $\mathcal{Q}_t$ at each iteration (see below).

The SDDP algorithm is made of a sequence of forward-backward passes that we detail now for problem (1).

Using the notation introduced in Section 2.1, the forward pass computes H feasible states $(x_t^k, \xi_{[t]}^k)$, $k = (i-1)H+1, \ldots, iH$, for time steps $t = 1, \ldots, T$. We recall that $(\xi_{[1]}^k, \ldots, \xi_{[T]}^k)$ is obtained from a sample $(\eta_2^k, \ldots, \eta_T^k)$, $k = (i-1)H+1, \ldots, iH$ of the noises $(\eta_2, \ldots, \eta_T)$, using (3) and (15). The forward pass then computes x_t^k solving the following approximate problem $[AP_t^{i,k}]$ of $[LP_t]$ obtained replacing $\mathcal{Q}_{t+1}$ by $\mathcal{Q}_{t+1}^{i-1}$:

$$[AP_t^{i,k}] \quad \mathcal{Q}_t^i(x_{t-1}^k, \xi_{[t-1]}^k) = \left\{ \begin{array}{ll} \inf\limits_{x_t^k, \theta_t^k} f_t(x_t^k) + \theta_t^k & \\ A_t x_t^k \geq b_t(\xi_t^k) - B_t x_{t-1}^k & \\ C_t x_t^k = D_t \xi_t^k - E_t x_{t-1}^k & \\ \overrightarrow{E}_t^{i-1} x_t^k + \theta_t^k e \geq \overrightarrow{\tilde{E}}_t^{i-1} \xi_{[t]}^k + \overrightarrow{e}_t^{i-1} & (a) \\ \overrightarrow{F}_t x_t^k \geq \overrightarrow{\tilde{F}}_t \xi_{[t]}^k + \overrightarrow{f}_t & (b) \\ x_t^k \geq 0, & \end{array} \right.$$

for $t = 1, \ldots, T$, and $k = (i-1)H+1, \ldots, iH$. In the above expression, constraints (a) are optimality cuts while constraints (b) are feasibility cuts. Matrices $\overrightarrow{F}_t$, $\overrightarrow{\tilde{F}}_t$, and vector $\overrightarrow{f}_t$ can be modified various times in a given iteration. As a result, constraints (b) correspond to the feasibility cuts for x_t that are available so far. For more details on the computation of feasibility cuts, see Sections 2.5 and 2.6.

We now detail the computations of optimality and feasibility cuts in our interstage dependent framework.

2.4. **SDDP: backward pass.** The optimality cuts are computed for time step $T+1$ down to time step 2.

We start our computations when Assumption (A7) holds. Proceeding as in Section 2.1, we first express $\xi_{[t]}$ and ξ_t as a function of η_t and $\xi_{[t-1]}$. For this, we introduce the $M \times \sum_{k=1}^M s_{t,k}$ matrix Φ_t defined by

$$\begin{array}{ll} \Phi_t(m, \sum_{k=1}^{m-1} s_{t,k} + 1 : \sum_{k=1}^{m-1} s_{t,k} + p_t(m)) = \Phi_t(m), & \\ \Phi_t(m, j) = 0 & \text{otherwise,} \end{array}$$

for $m = 1, \ldots, M$, where $\Phi_t(m) = (\Phi_t^1(m), \Phi_t^2(m), \ldots, \Phi_t^{p_t(m)})$ with $\Phi_t^j(m) = \frac{\sigma_t(m)}{\sigma_{t-j}(m)} \phi_t^j(m), j = 1, \ldots, p_t(m)$. Setting Ψ_t the $M \times M$ matrix $\Psi_t = \mathrm{Diag}(\sigma_t(1), \ldots, \sigma_t(M))$ and Θ_t the M-vector given by $\Theta_t(m) = \mu_t(m) - \sum_{j=1}^{p_t(m)} \Phi_t^j(m) \mu_{t-j}(m)$, $m = 1, \ldots, M$, we have

$$\xi_t = \Phi_t \xi_{[t-1]} + \Psi_t \eta_t + \Theta_t. \tag{16}$$

Next, let $\tilde{\Phi}_t$ be the $\sum_{k=1}^M s_{t+1,k} \times \sum_{k=1}^M s_{t,k}$ matrix whose non-zero elements are given by

$$\begin{array}{l} \tilde{\Phi}_t(\sum_{k=1}^{m-1} s_{t+1,k} + 1, \sum_{k=1}^{m-1} s_{t,k} + 1 : \sum_{k=1}^{m-1} s_{t,k} + p_t(m)) = \Phi_t(m), \quad m = 1, \ldots, M, \\ \tilde{\Phi}_t(\sum_{k=1}^{m-1} s_{t+1,k} + j, \sum_{k=1}^{m-1} s_{t,k} + j - 1) = 1, \quad m = 1, \ldots, M, \ j = 2, \ldots, s_{t+1,m}. \end{array}$$

Notice that in the expression above, $1 \leq j-1 \leq s_{t+1,m}-1 \leq s_{t,m}$. Finally, $\tilde{\Psi}_t$ and $\tilde{\Theta}_t$ are respectively the $\sum_{k=1}^{M} s_{t+1,k} \times M$ matrix and the $\sum_{k=1}^{M} s_{t+1,k}$-vector whose non-zero elements are given by

$$\begin{array}{ll} \tilde{\Psi}_t(\sum_{k=1}^{m-1} s_{t+1,k}+1, m) = \sigma_t(m) & m=1,\ldots,M, \\ \tilde{\Theta}_t(\sum_{k=1}^{m-1} s_{t+1,k}+1) = \mu_t(m) - \sum_{j=1}^{p_t(m)} \Phi_t^j(m)\mu_{t-j}(m) & m=1,\ldots,M. \end{array}$$

With this notation, we obtain

$$\xi_{[t]} = \tilde{\Phi}_t \xi_{[t-1]} + \tilde{\Psi}_t \eta_t + \tilde{\Theta}_t. \tag{17}$$

To alleviate notation and without loss of generality, we assume in (2) that $X_t = \mathbb{R}^{k_t}$. Plugging (12) (still valid) into (4) and using (16) and (17), we obtain $\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}) \geq \mathbb{E}_{\eta_t}\left[Q_t^i(x_{t-1}, \xi_{[t-1]}, \eta_t)\right]$ with

$$Q_t^i(x_{t-1}, \xi_{[t-1]}, \eta_t) = \left\{ \begin{array}{l} \inf\limits_{x_t, \theta_{t1}, \theta_{t2}} \theta_{t1} + \theta_{t2} \\ A_t x_t \geq b_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_t + \Theta_t) - B_t x_{t-1} \\ C_t x_t = D_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_t + \Theta_t) - E_t x_{t-1} \\ -\beta_t x_t + \theta_{t1} e \geq \alpha_t \\ \overrightarrow{E}_t^i x_t + \theta_{t2} e \geq \overrightarrow{E}_t^i \left(\tilde{\Phi}_t \xi_{[t-1]} + \tilde{\Psi}_t \eta_t + \tilde{\Theta}_t\right) + \overrightarrow{e}_t^i \\ x_t \geq 0 \end{array} \right. \tag{18}$$

where β_t is the matrix $\beta_t = [\beta_{t1}^\top; \ldots; \beta_{tJ_t}^\top]$ and α_t is the vector $\alpha_t = (\alpha_{t1}, \ldots, \alpha_{tJ_t})^\top$. On scenario k and time step t, the above problem is solved for $(x_{t-1}, \xi_{[t-1]}, \eta_t) = (x_{t-1}^k, \xi_{[t-1]}^k, \eta_{tj})$, $j = 1, \ldots, q_t$. The forward pass ensures that the feasible sets of these optimization problems are nonempty. Since Assumption (A3) holds, the optimal values of these linear programs are finite and both the primal and the dual have the same optimal value. We denote by $\pi_{t2}^{kj}, \pi_{t1}^{kj}, \lambda_t^{kj}$, and ρ_t^{kj} the (row vectors) optimal Lagrange multipliers associated to respectively the first, second, third, and fourth group of constraints for problem $\mathcal{Q}_t^i(x_{t-1}^k, \xi_{[t-1]}^k, \eta_{tj})$. Finally, $s_{ti}^b(x)$ (resp. $s_{ti}^h(x)$) will denote a subgradient of convex function b_{ti} (resp. h_{ti}) at x. The following theorem provides the cuts computed for $\mathcal{Q}_t$ at iteration i:

Theorem 2.4 (Optimality cuts-Affine process model)**.** *Consider recourse functions $\mathcal{Q}_t$ from* (4) *and let Assumptions (A1), (A2), (A3), and (A7) hold. Let ξ_{tj}^k be the M-vector given by*

$$\xi_{tj}^k = \Phi_t \xi_{[t-1]}^k + \Psi_t \eta_{tj} + \Theta_t. \tag{19}$$

In the backward pass of iteration i of the SDDP algorithm, H valid cuts for $\mathcal{Q}_t, t = 2, \ldots, T$, are given by $E_{t-1}^k = \sum_{j=1}^{q_t} p(t,j) E_{t-1}^{kj}$, $\tilde{E}_{t-1}^k = \sum_{j=1}^{q_t} p(t,j)\tilde{E}_{t-1}^{kj}$, and $e_{t-1}^k = \sum_{j=1}^{q_t} p(t,j) e_{t-1}^{kj}$ where

$$\begin{array}{l} E_{t-1}^{kj} = \pi_{t1}^{kj} E_t + \pi_{t2}^{kj} B_t, \\ \tilde{E}_{t-1}^{kj} = \rho_t^{kj} \overrightarrow{E}_t^i \tilde{\Phi}_t + \pi_{t1}^{kj} D_t \Phi_t + \pi_{t2}^{kj} s_t^b(\xi_{tj}^k)\Phi_t, \\ e_{t-1}^{kj} = \lambda_t^{kj}\alpha_t + \rho_t^{kj}(\overrightarrow{e}_t^i + \overrightarrow{E}_t^i(\tilde{\Psi}_t \eta_{tj} + \tilde{\Theta}_t)) + \pi_{t1}^{kj} D_t(\Psi_t \eta_{tj} + \Theta_t) + \pi_{t2}^{kj}(b_t(\xi_{tj}^k) - s_t^b(\xi_{tj}^k)\Phi_t \xi_{[t-1]}^k), \end{array}$$

for $t = 2, \ldots, T$, $k = (i-1)H+1, \ldots, iH$, and $E_T^k = \tilde{E}_T^k = e_T^k = 0$ for $k = (i-1)H+1, \ldots, iH$. In these expressions, $s_t^b(x)$ is the matrix $s_t^b(x) = (s_{t1}^b(x)^\top; \ldots; s_{t\ell_t}^b(x)^\top)$.

For implementation purposes, it is convenient to decompose $\tilde{E}_{t-1}^k$ as $\tilde{E}_{t-1}^k = (\tilde{E}_{t-1,1}^k, \ldots, \tilde{E}_{t-1,M}^k)$ with $\tilde{E}_{t-1,m}^k \in \mathbb{R}^{1 \times s_{t,m}}$ and to get rid of (large size and sparse) matrices Φ_t and $\tilde{\Phi}_t$ in the formula for $\tilde{E}_{t-1,m}^k$. Such decomposition is provided in the following corollary of Theorem 2.4:

Corollary 2.5. *Let $\tilde{E}^k_{t-1}$ be defined as in Theorem 2.4 and let us decompose $\tilde{E}^k_{t-1}$ as $\tilde{E}^k_{t-1} = (\tilde{E}^k_{t-1,1}, \ldots, \tilde{E}^k_{t-1,M})$ with $\tilde{E}^k_{t-1,m} \in \mathbb{R}^{1\times s_{t,m}}$. We have $\tilde{E}^k_{t-1,m} = \sum_{j=1}^{q_t} p(t,j)\tilde{E}^{kj}_{t-1,m}$ where the transpose of $\tilde{E}^{kj}_{t-1,m}$ is given by*

$$\left[\begin{array}{c}\left(\sum_w \pi^{kj}_{t1}(w)D_t(w,m) + \sum_{w=0}^{iH} \rho^{kj}_t(w)\tilde{E}^w_{t,m}(1) + \sum_{w=1}^{\ell_t}\pi^{kj}_{t2}(w)s^b_{tw}(\xi^k_{tj})(m)\right)\Phi_t(m)^\top \\ 0_{(s_{t,m}-p_t(m))\times 1}\end{array}\right]$$
$$+\left[\begin{array}{c}\sum_{w=0}^{iH}\rho^{kj}_t(w)\tilde{E}^w_{t,m}(2:s_{t+1,m})^\top \\ 0_{(s_{t,m}-s_{t+1,m}+1)\times 1}\end{array}\right].$$

Also, in Theorem 2.4, e^{kj}_{t-1} can be expressed as

$$\lambda^{kj}_t\alpha_t + \rho^{kj}_t \overrightarrow{e}^{\,i}_t + \sum_{m=1}^{M}(\Theta_t(m)+\sigma_t(m)\eta_{tj}(m))\left(\sum_w \pi^{kj}_{t1}(w)D_t(w,m) + \sum_{w=0}^{iH}\rho^{kj}_t(w)\tilde{E}^w_{t,m}(1)\right)$$
$$+\pi^{kj}_{t2}b_t(\xi^k_{tj}) - \sum_{m=1}^{M}\sum_{w=1}^{\ell_t}\sum_{v=1}^{p_t(m)}\pi^{kj}_{t2}(w)s^b_{tw}(\xi^k_{tj})(m)\Phi^v_t(m)\xi^k_{t-v}(m).$$

Remark 2.6. *Using the convexity of Q_t, we can also express e^{kj}_{t-1} as*

$$Q^i_t(x^k_{t-1},\xi^k_{[t-1]},\eta_{tj}) + \left(\pi^{kj}_{t2}B_t + \pi^{kj}_{t1}E_t\right)x^k_{t-1} - \left(\pi^{kj}_{t2}s^b_t(\xi^k_{tj})\Phi_t + \rho^{kj}_t \overrightarrow{E}^{\,i}_t\tilde{\Phi}_t + \pi^{kj}_{t1}D_t\Phi_t\right)\xi^k_{[t-1]}. \tag{20}$$

The previous corollary shows that for some interstage dependent processes, it is possible to share optimality cuts between nodes of the same stage. This was first observed in [10]. We mention some differences between the optimality cuts derived in [10] and those written in the above corollary. In [10], the affine model is written in the vectorial form:

$$\xi_t = \sum_{j=1}^{t-1}(R^t_j\xi_j + S^t_j\eta_j), \tag{21}$$

whereas we consider separate models for each process component. On the one hand, the above vectorial form allows for dependences between different components and noises $\eta_j, j < t$, but on the other hand, the number of terms in the sum is not a parameter of the model as is the case for model (8). When ξ_t has many components and for many stages, many large size matrices R^t_j, S^t_j will be involved in the formulas for the cuts. Moreover, for large time steps, these formulas will provide a large number of cut coefficients (with possibly a large number of null coefficients) whereas we consider a minimal subset of such coefficients necessary for building the cuts. On the other hand, we do not provide as in [10] iterative formulas to compute the cut coefficients. As a result, depending on the application, one formulation or the other may be more interesting in terms of memory allocation. In the simple 3-stage example which follows (a small hydro-thermal problem), we show that our formulas for the cuts are more economic in terms of memory allocation.

Example 2.7 (Optimality cuts for a simple hydro-thermal problem)**.** *Consider a simplified hydro-thermal problem with $T = 3$ stages, 2 independent hydroplants, and a thermal plant. The stochastic process (ξ_t) corresponds to the process of inflows and satisfy $\xi_2(1) = \frac{1}{2}\xi_1(1)+\eta_2(1)$, $\xi_3(1) = \frac{1}{2}(\xi_2(1)+\xi_1(1)+\xi_0(1))+\eta_3(1)$ for the first reservoir and $\xi_t(2) = \frac{1}{2}\xi_{t-1}(2)+\eta_t(2)$, $t = 2,3$, for the second. For these models, vectors η_t are independent, and past inflows are given by $\xi_0 = (1;1)$ and $\xi_1 = (0;0)$. For each t, the possible realizations of η_t are $(0;0),(0;5),(5;0)$, and $(5;5)$, each with probability 0.25. Demands are known and set to respectively 14, 12, and 18 for the first three time steps. Reservoir* *volumes are nonnegative, initial reservoir levels are* $(7;7)$*, and unit thermal cost is 1 while hydro plants produce without cost. To avoid feasibility problems (discussed in the next section) we assume that* $H=1$ *and that the sampled scenario of inflows in the forward pass of the first iteration is the wet scenario* $(0;5;8)$ *for the first reservoir and* $(0;5;7.5)$ *for the second (obtained taking the realization* $(5;5)$ *for both* η_2 *and* η_3*). We denote by* V_t *the vector of reservoir levels at the end of time step* t*. In the backward pass, after some simple computations obtained following the previous developments of this section, we obtain for* $\mathcal{Q}_3$ *the cut*

$$\mathcal{Q}_3(V_2,\xi_2(1),\xi_1(1),\xi_0(1),\xi_2(2)) \geq 13 - e^\top V_2 - \frac{1}{2}\xi_2(2) - \frac{1}{2}(\xi_2(1)+\xi_1(1)+\xi_0(1))$$

and for $\mathcal{Q}_2$ *the cut* $\mathcal{Q}_2(V_1) \geq 7 - e^\top V_1$*. Writing the process of inflows in the form* (21)*, the cut for* $\mathcal{Q}_3$ *would be expressed in terms of matrices* $R_2^3 = \left(\begin{array}{cc} \frac{1}{2} & 0 \\ 0 & \frac{1}{2}\end{array}\right)$ *and* $R_1^3 = R_0^3 = \left(\begin{array}{cc} \frac{1}{2} & 0 \\ 0 & 0\end{array}\right)$ *as follows:* $\mathcal{Q}_3(V_2,\xi_2,\xi_1,\xi_0) \geq 13 - e^\top V_2 - e^\top(R_2^3\xi_2 + R_1^3\xi_1 + R_0^3\xi_0)$, *(note also that the latter formulation involves more arguments for* $\mathcal{Q}_3$*).*

We now provide in Theorem 2.8 the formulas for optimality cuts for the convex process model. In this case, we consider problems of form (1) without equality constraints. As in the proof of Theorem 2.4, we introduce $\mathcal{Q}_t^i(x_{t-1},\xi_{[t-1]},\eta_{tj})$ obtained using (3) and replacing recourse function $\mathcal{Q}_{t+1}$ by its lower bounding approximation $\mathcal{Q}_{t+1}^i$. We denote by $\pi_{t2}^{kj}, \lambda_t^{kj}$, and ρ_t^{kj} the (row vectors) optimal Lagrange multipliers associated to respectively the first, second, and third group of constraints for the corresponding problem $\mathcal{Q}_t^i(x_{t-1}^k,\xi_{[t-1]}^k,\eta_{tj})$.

Theorem 2.8 (Optimality cuts-Convex process model)**.** *Consider stochastic optimization problem* (1) *without equality constraints and the corresponding recourse functions* $\mathcal{Q}_t$ *from* (4)*. As in Corollary 2.5, let us decompose* $\tilde{E}_{t-1}^k$ *as* $\tilde{E}_{t-1}^k = (\tilde{E}_{t-1,1}^k,\ldots,\tilde{E}_{t-1,M}^k)$ *with* $\tilde{E}_{t-1,m}^k \in \mathbb{R}^{1\times s_{t,m}}$*. Let* ξ_{tj} *be defined as in Assumption (A3) and let* ξ_{tj}^k *be the* M*-vector defined by*

$$\xi_{tj}^k(m) = h_{tm}(\xi_{t-1}^k(m),\ldots,\xi_{t-p_t(m)}^k(m),\eta_{tj}(m)),\ m=1,\ldots,M. \tag{22}$$

Let Assumptions (A1), (A2), (A3), (A4), (A5), and (A6) hold. In the backward pass of iteration i *of the SDDP algorithm,* H *valid cuts for* $\mathcal{Q}_{T+1}$ *are given by* $E_T^k = \tilde{E}_T^k = e_T^k = 0$ *for* $k=(i-1)H+1,\ldots,iH$*. For* $t=2,\ldots,T$*,* H *valid cuts for* $\mathcal{Q}_t$ *are given by* $E_{t-1}^k = \sum_{j=1}^{q_t} p(t,j)\pi_{t2}^{kj}B_t$*,* $\tilde{E}_{t-1,m}^k = \sum_{j=1}^{q_t} p(t,j)\tilde{E}_{t-1,m}^{kj}$*, and* $e_{t-1}^k = \sum_{j=1}^{q_t} p(t,j)e_{t-1}^{kj}$ *where the transpose of* $\tilde{E}_{t-1,m}^{kj}$ *is given by*

$$\begin{bmatrix} \left(\displaystyle\sum_{w=0}^{iH}\rho_t^{kj}(w)\tilde{E}_{t,m}^w(1) + \sum_{\ell=1}^{\ell_t}\pi_{t2}^{kj}(\ell)s_{t\ell}^b(\xi_{tj}^k)(m)\right) s_{tm}^h\left(\xi_{t-1:t-p_t(m)}^k(m),\eta_{tj}(m)\right)(1:p_t(m)) \\ 0_{(s_{t,m}-p_t(m))\times 1}\end{bmatrix} + \begin{bmatrix} \displaystyle\sum_{\ell=0}^{iH}\rho_t^{kj}(\ell)\tilde{E}_{t,m}^\ell(2:s_{t+1,m})^\top \\ 0_{(s_{t,m}-s_{t+1,m}+1)\times 1}\end{bmatrix} \tag{23}$$

and where $e_{t-1}^{kj} - \lambda_t^{kj}\alpha_t - \rho_t^{kj}\overrightarrow{e}_t^{\,i} - \pi_{t2}^{kj}b_t(\xi_{tj}^k)$ *is given by*

$$\begin{array}{l} -\displaystyle\sum_{m=1}^{M}\sum_{w=1}^{p_t(m)}\sum_{\ell=1}^{\ell_t}\pi_{t2}^{kj}(\ell)s_{t\ell}^b(\xi_{tj}^k)(m)s_{tm}^h\left(\xi_{t-1:t-p_t(m)}^k(m),\eta_{tj}(m)\right)(w)\xi_{t-w}^k(m) \\ +\displaystyle\sum_{m=1}^{M}\sum_{w=0}^{iH}\rho_t^{kj}(w)\tilde{E}_{t,m}^w(1)\left(\xi_{tj}^k(m) - \sum_{u=1}^{p_t(m)} s_{tm}^h\left(\xi_{t-1:t-p_t(m)}^k(m),\eta_{tj}(m)\right)(u)\xi_{t-u}^k(m)\right).\end{array} \tag{24}$$

Following the lines of DOASA algorithm introduced in [15], we can reduce the per-iteration computational effort choosing for each time step t and scenario k a nonempty subset Ω_t^k of Ω_t. Next, in the backward pass, at iteration i, for each scenario $k = (i-1)H+1, \ldots, iH$, instead of solving all subproblems $\mathcal{Q}_t^i(x_{t-1}^k, \xi_{[t-1]}^k, \eta_{tj}), j = 1 \ldots, q_t$, only subproblems $\mathcal{Q}_t^i(x_{t-1}^k, \xi_{[t-1]}^k, \eta_{tj})$, with $\eta_{tj} \in \Omega_t^k$ are solved. After solving these problems, optimal dual multipliers are stored, i.e., the following set $\mathcal{M}_t^i$ of multipliers is updated:

$$\mathcal{M}_t^i = \{(\lambda_t^{kj}, \pi_{t1}^{kj}, \pi_{t2}^{kj}, \rho_t^{kj}), k = (i-1)H+1, \ldots, iH, \; j \; : \; \eta_{tj} \in \Omega_t^k\},$$

if there are equality constraints in (1) and

$$\mathcal{M}_t^i = \{(\lambda_t^{kj}, \pi_{t2}^{kj}, \rho_t^{kj}), k = (i-1)H+1, \ldots, iH, \; j \; : \; \eta_{tj} \in \Omega_t^k\}$$

otherwise. Next, for every $k = (i-1)H+1, \ldots, iH$, and for every j such that η_{tj} belongs to Ω_t but not to Ω_t^k, the "missing" multipliers are replaced by

$$(\lambda_t^{kj}, \pi_{t1}^{kj}, \pi_{t2}^{kj}, \rho_t^{kj}) \in \mathrm{Argmax}_{(\lambda_t, \pi_{t1}, \pi_{t2}, \rho_t) \in \mathcal{M}_t^i} \; g_1(\lambda_t, \pi_{t1}, \pi_{t2}, \rho_t, \eta_{tj}) \tag{25}$$

if there are equality constraints in (1) and

$$(\lambda_t^{kj}, \pi_{t2}^{kj}, \rho_t^{kj}) \in \mathrm{Argmax}_{(\lambda_t, \pi_{t2}, \rho_t) \in \mathcal{M}_t^i} \; g_2(\lambda_t, \pi_{t2}, \rho_t, \eta_{tj}) \tag{26}$$

otherwise. In this context, the next proposition provides the cuts computed in the backward pass:

Proposition 2.9 (Optimality cuts for the affine and convex process models with DOASA)**.** *Consider stochastic optimization problem* (1) *and the corresponding recourse functions* $\mathcal{Q}_t$ *from* (4)*.*

If there are no equality constraints in (1) *(resp. if there are some equality constraints), assume that Assumptions (A1), (A2), (A3), (A4), (A5), and (A6) hold or that Assumptions (A1), (A2), (A3), and (A7) (resp. Assumptions (A1), (A2), (A3), and (A7)) hold. Next, for every* $k = (i-1)H+1, \ldots, iH$*, and for every* $j = 1, \ldots, q_t$*, let* $(\lambda_t^{kj}, \pi_{t2}^{kj}, \rho_t^{kj})$ *(resp.* $(\lambda_t^{kj}, \pi_{t1}^{kj}, \pi_{t2}^{kj}, \rho_t^{kj})$*) given by* (26) *(resp.* (25)*).*

With these values of $(\lambda_t^{kj}, \pi_{t2}^{kj}, \rho_t^{kj})$ *(resp.* $(\lambda_t^{kj}, \pi_{t1}^{kj}, \pi_{t2}^{kj}, \rho_t^{kj})$*), Theorem 2.8 (resp. Theorem 2.4 and Corollary 2.5) defines valid cuts for recourse functions* $\mathcal{Q}_t, t = 2, \ldots, T+1$*.*

2.5. **SDDP: forward pass and feasibility cuts.** Before the first forward pass, in an initialization phase, trivial cuts (given by available lower bounding functions for $\mathcal{Q}_t$) are built, taking for instance a constant value for e_{t-1}^0 and null values for E_{t-1}^0 and $\tilde{E}_{t-1}^0$.

In the forward pass, at stage t and given a $(t-1)$-stage scenario $(\xi_1^k, \ldots, \xi_{t-1}^k)$, before proceeding forward to the next stage, we need to check if x_{t-1}^k yields a feasible $[AP_t^{i,k}]$ and that cuts can be built at $(x_{t-1}^k, \xi_{[t-1]}^k)$ in the next backward step. If this is not the case then either $t = 1$ and the problem is infeasible or a feasibility cut (an additional constraint) needs to be added to stage $(t-1)$ subproblems. For interstage dependent stochastic linear programs (IDSLP), the subtrees rooted at the different nodes of a given time step are in general different and feasibility cuts cannot in general be shared between these nodes. However, we will show that in our context the feasibility cut sharing property holds. More precisely, when infeasibility arises in the course of the algorithm at a node of stage t that belongs to some sampled path, rows F_{t-1}, $\tilde{F}_{t-1}$, and f_{t-1} are added to respectively $\overrightarrow{F}_{t-1}$, $\overrightarrow{\tilde{F}}_{t-1}$, and $\overrightarrow{f}_{t-1}$. These rows added are such that given the history $\xi_{[t-1]}$ until stage $t-1$, if x_{t-1} is a feasible output state for stage $t-1$ then $F_{t-1}x_{t-1} \geq \tilde{F}_{t-1}\xi_{[t-1]} + f_{t-1}$. The methodology to build feasibility cuts is the same as with the Nested Decomposition algorithm (see [1] for instance). However, to the best of our knowledge the feasibility cut sharing property for SDDP in our interstage dependent context has not been explained so far in the literature.

To understand how feasibility cuts are built and can be shared in our context between nodes of the same stage, we consider a simple hydro-thermal planning problem. We have one reservoir with volume V_t at the end of period t and hydro generation u_t for this period. Thermal generation is denoted by w_t for period t and thermal capacity is very large, i.e., (deterministic) demand can be satisfied by thermal power at each stage. We consider $T = 4$ stages and impose the restrictions $V_1 \geq 0, V_2 \geq 0$, $V_3 \geq 0$, and $V_4 \geq 7$ on reservoir levels, knowing that $V_0 = 7$. The vector of demands d is $d = (7; 1; 1; 1)$ (in the same energy unit as u_t and w_t) and thermal unit cost is one. Let the model for the inflows be of form (8): $\xi_t = 0.5\xi_{t-1} + \eta_t$ with $\xi_1 = 0$ and $\mathbb{P}(\eta_t = 0) = 0.5$, $\mathbb{P}(\eta_t = 5) = 0.5$. The corresponding scenarios for the first 4 stages are represented in Figure 2.

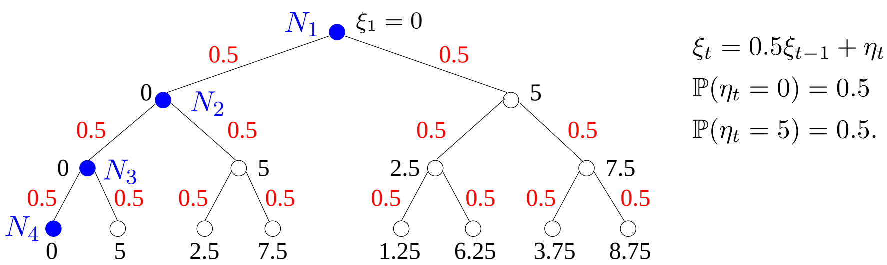

FIGURE 2. Scenario tree of inflows.

In this context, the problem is feasible since on each scenario we can use a fully thermal production plan. At a given stage $t-1$ and for a given history $\xi_{[t-1]}$, output state x_{t-1} is said to be feasible if for any future scenario there exist feasible decisions for every time step.

Let us take $H = 1$ scenario per iteration and let us see how the forward pass of the first iteration of SDDP is like if the first sampled scenario is the pessimistic (dry) scenario of inflows $(\xi_1^1, \xi_2^1, \xi_3^1, \xi_4^1) = (0, 0, 0, 0)$ (nodes N_1, N_2, N_3, and N_4 of the scenario tree in Figure 2). Let us start the algorithm with null approximations of the recourse functions, i.e., E_{t-1}^0, $\tilde{E}_{t-1}^0$, and e_{t-1}^0 are null (costs are nonnegative). Problem $[AP_1^{11}]$ (see Section 2.3) reads

$$
\begin{array}{l}
\min\ w_t^1 \\
V_t^1 = V_{t-1}^1 - u_t^1 + \xi_t^1,\ u_t^1 + w_t^1 \geq d_t, \\
u_t^1 \geq 0,\ w_t^1 \geq 0,\ V_t^1 \geq V_t^{\min},
\end{array}
\tag{27}
$$

with $t = 1$, $V_t^{\min} = 0$, and $V_0^1 = V_0 = 7$. The solution is given by $u_1^1 = 7, w_1^1 = 0$, and $V_1^1 = 0$. For the second stage, we solve problem $[AP_2^{11}]$ which is of form (27) with $t = 2, V_t^{\min} = 0$, $d_t = 1$, and $V_1^1 = 0$. The solution is given by $u_2^1 = 0, w_2^1 = 1$, and $V_2^1 = 0$. For stage 3, we consider problem $[AP_3^{11}]$ which is of form (27) with $t = 3, V_t^{\min} = 0$, $d_t = 1$, $V_2^1 = 0$. The solution is given by $u_3^1 = 0, w_3^1 = 1$, and $V_3^1 = 0$. For the last stage, $[AP_4^{11}]$ is of form (27) with $t = 4, V_t^{\min} = 7$, $d_t = 1$, $V_3^1 = 0$. This problem is infeasible. We thus see that relatively complete recourse does not hold. As a result, a feasibility cut needs to be built at node N_3. Such feasibility cut is a constraint satisfied by all feasible reservoir levels V_3 at node N_3, at the end of stage 3. If state V_3 is feasible for node N_3 then the optimal value of the following linear program is 0:

$$
\begin{array}{l}
\min\ \sum_{i=1}^4\ x_i \\
w_t^1 + u_t^1 + x_1 \geq d_t \\
V_t^1 + x_2 - x_3 + u_t^1 = V_{t-1} + \xi_t, \\
u_t^1 \geq 0,\ w_t^1 \geq 0,\ V_t^1 \geq 0,\ V_t^1 + x_4 \geq V_t^{\min},\ x_i \geq 0,\ i = 1, \ldots, 4,
\end{array}
\tag{28}
$$

where $t=4$ and where ξ_t is the inflow at N_4, i.e., $\xi_t=0$. Now observe that on the dry scenario, we have $\xi_4=\frac{1}{2}\xi_3=0$. As a result, problem (28) with $t=4$ can be written as

$$
\begin{array}{l}
\min \sum_{i=1}^{4} x_i\\
w_t^1+u_t^1+x_1\geq d_t\\
V_t^1+x_2-x_3+u_t^1=V_{t-1}+\frac{1}{2}\xi_{t-1},\\
u_t^1\geq 0,\ w_t^1\geq 0,\ V_t^1\geq 0,\ V_t^1+x_4\geq V_t^{\min},\ x_i\geq 0,\ i=1,\ldots,4,
\end{array}
\tag{29}
$$

where $\xi_{t-1}=0$ is the inflow at node N_3 (father of N_4). Next, observe that for any node n of stage 3, there is a son node where the realization of η_4 is 0, i.e., a node such that the inflow is half the inflow for his father node n. As a result, for any node of stage 3, if V_3 is feasible then the optimal value of (29) is 0. Moreover, wee see that a dual solution to (29) written with $V_{t-1}=0$ and $\xi_{t-1}=0$ is feasible for the dual of (29) written with given V_{t-1} and ξ_{t-1}. Consequently, using a dual solution to (29) written with $V_{t-1}=0$ and $\xi_{t-1}=0$ (problem solved at N_4), we obtain that if V_3 is feasible at stage 3 then $0\geq V_4^{\min}-(V_3+\frac{1}{2}\xi_3)$, i.e., $V_3\geq 7-\frac{1}{2}\xi_3$.

We then go back to node N_3 and solve (27) with $t=3$, $V_t^{\min}=0, V_2^1=0$, and with the additional cut $V_3\geq 7$ valid for node N_3. This problem is not feasible. To build a feasibility cut for V_2, we use the fact that if state V_2 is feasible for node N_2 then the optimal value of the following linear program is 0:

$$
\begin{array}{l}
\min \sum_{i=1}^{5} x_i\\
w_t^1+u_t^1+x_1\geq d_t\\
V_t^1+x_2-x_3+u_t^1=V_{t-1}+\xi_t,\\
V_t^1\geq 0,\ V_t^1+x_4\geq V_t^{\min},\ V_t^1+x_5\geq 7-\frac{1}{2}\xi_t,\\
u_t^1\geq 0,\ w_t^1\geq 0,\ x_i\geq 0,\ i=1,\ldots,5,
\end{array}
\tag{30}
$$

where $t=3$ and where $\xi_t=0$ is the inflow at node N_3. Next, we observe that at N_3 we have $\xi_t=\frac{1}{2}\xi_{t-1}$ where ξ_{t-1} is the inflow for father node N_2. As before, we also note that for any node n of stage 2, there is a son node where the realization of η_3 is 0, i.e., a node such that the inflow is half the inflow for its father node n. As a result, for any node of stage 2, if V_2 is feasible then the optimal value of (30) with ξ_t replaced by $\frac{1}{2}\xi_{t-1}$ is 0. As before, using a dual solution to (30) with V_{t-1} and ξ_t null, we obtain the cut $0\geq 7-\frac{1}{4}\xi_{t-1}-(V_{t-1}+\frac{1}{2}\xi_{t-1})$ with $t=3$, i.e., $V_2\geq 7-\frac{3}{4}\xi_2$. This cut is valid for all nodes of stage 2.

We then go back to node N_2 solving (27) with $t=2$, $V_t^{\min}=0, V_1^1=0$, and with the additional cut $V_2\geq 7$ valid for node N_2. This problem is not feasible. Reasoning as before, we use the fact that if state V_2 is feasible for a node of stage 2, then the optimal value of the following linear program is 0:

$$
\begin{array}{l}
\min \sum_{i=1}^{5} x_i\\
w_t^1+u_t^1+x_1\geq d_t\\
V_t^1+x_2-x_3+u_t^1=V_{t-1}+\frac{1}{2}\xi_{t-1},\\
V_t^1\geq 0,\ V_t^1+x_4\geq V_t^{\min},\ V_t^1+x_5\geq 7-\frac{3}{8}\xi_{t-1},\\
u_t^1\geq 0,\ w_t^1\geq 0,\ x_i\geq 0,\ i=1,\ldots,5,
\end{array}
\tag{31}
$$

where $t=2$ and where $\xi_{t-1}=0$ is the inflow at node N_1. We obtain the cut $V_1\geq 7-\frac{7}{8}\xi_1=7$, i.e., there will be no hydro generation at the root node.

We go back to the first stage (node N_1) and solve $[AP_1^{11}]$ with $t=1$, $V_t^{\min}=0$, $V_0^1=V_0=7$, and the cut $V_1\geq 7$. We find the optimal solution $u_1^1=0, w_1^1=7$, and $V_1^1=7$. No more feasibility cuts are needed at this iteration and we find the solutions $u_t^1=0, w_t^1=1$, and $V_t^1=7,\ t=2,3,4$.

The feasibility cuts constructed in this example are easily interpreted. Indeed, let us consider a given stage t and a node of this stage with inflow ξ_t. The level of the reservoir V_t at the end of stage t at this node plus the minimal future inflow needs to be above $V_T^{\min}=V_0=7$. The minimal future inflow is obtained when all future realizations of the noises are 0. On this dry scenario, inflows are $\frac{1}{2}\xi_t$ at stage $t+1$, $\frac{1}{4}\xi_t$ at stage $t+2$, ..., and $\frac{1}{2^{T-t}}\xi_t$ at stage T. We thus obtain the cut $V_t + \sum_{i=1}^{T-t} \frac{\xi_t}{2^i} \geq V_T^{\min}$, i.e., $V_t \geq V_T^{\min} - (1 - \frac{1}{2^{T-t}})\xi_t$. When $T=4$ and $t=1,2$, or 3, we obtain the cuts previously built.

We now consider the general case and provide formulas for the feasibility cuts built in the forward pass of the algorithm.

Let us start with the affine process model. On a given node of stage $t-1$ with history $\xi_{[t-1]}$, if state x_{t-1} is feasible then all subproblems for all sons of this node must be feasible, i.e., the optimal value of the optimization problem

$$(32) \qquad \left\{ \begin{array}{l} \displaystyle\min_{x_t, v_{t1}, v_{t2}, v_{t3}, v_{t4}} \|(v_{t1}^\top, v_{t2}^\top, v_{t3}^\top, v_{t4}^\top)^\top\|_1 \\ A_t x_t + v_{t1} \geq b_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_{tj} + \Theta_t) - B_t x_{t-1} \\ C_t x_t + v_{t2} - v_{t3} = D_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_{tj} + \Theta_t) - E_t x_{t-1} \\ \overrightarrow{F}_t x_t + v_{t4} \geq \overrightarrow{F}_t(\tilde{\Phi}_t \xi_{[t-1]} + \tilde{\Psi}_t \eta_{tj} + \tilde{\Theta}_t) + \overrightarrow{f}_t \\ x_t, v_{t1}, v_{t2}, v_{t3}, v_{t4} \geq 0 \end{array} \right.$$

must be 0 for every $j = 1, \ldots, q_t$. In the forward pass, on scenario $\xi_{1:T}^k$, to know if x_{t-1}^k yields feasible problems for all son nodes, problems (32) for $j = 1, \ldots, q_t$, are solved with $\xi_{[t-1]}$ and x_{t-1} respectively replaced by $\xi_{[t-1]}^k$ and x_{t-1}^k. If one of these problems is not feasible then a feasibility cut is built as explained in the following theorem.

Theorem 2.10 (Feasibility cuts-Affine process model)**.** *Consider optimization problem* (32) *for some $j \in \{1, \ldots, q_t\}$ and with $\xi_{[t-1]}$ and x_{t-1} respectively replaced by $\xi_{[t-1]}^k$ and x_{t-1}^k. Assume that the optimal value of this problem is positive and that Assumptions (A1), (A2), and (A3) hold. Let σ_t^{kj} be a row vector of optimal dual variables for feasibility cuts and let π_{t1}^{kj} (resp. π_{t2}^{kj}) be a row vector of optimal dual variables for the equality constraints (resp. the remaining inequality constraints). A feasibility cut can be built for x_{t-1} adding respectively*

$$\begin{array}{ll} \text{row vector } \pi_{t1}^{kj} E_t + \pi_{t2}^{kj} B_t & \text{to } \overrightarrow{F}_{t-1}, \\ \text{row vector } \pi_{t1}^{kj} D_t \Phi_t + \pi_{t2}^{kj} s_t^b(\xi_{tj}^k)\Phi_t + \sigma_t^{kj} \overrightarrow{F}_t \tilde{\Phi}_t & \text{to } \overrightarrow{F}_{t-1}, \\ \text{scalar } \sigma_t^{kj}(\overrightarrow{f}_t + \overrightarrow{F}_t(\tilde{\Psi}_t \eta_{tj} + \tilde{\Theta}_t)) + \pi_{t1}^{kj} D_t(\Psi_t \eta_{tj} + \Theta_t) + \pi_{t2}^{k}(b_t(\xi_{tj}^k) - s_t^b(\xi_{tj}^k)\Phi_t \xi_{[t-1]}^k) & \text{to } \overrightarrow{f}_{t-1}, \end{array}$$

where we recall that $\xi_{tj}^k = \Phi_t \xi_{[t-1]}^k + \Psi_t \eta_{tj} + \Theta_t$ in the affine process model.

For the convex process model, on a given node of stage $t-1$ with history $\xi_{[t-1]}$, if state x_{t-1} is feasible then all subproblems for all sons of this node must be feasible, i.e., the optimal value of the optimization problem

$$(33) \qquad \left\{ \begin{array}{l} \displaystyle\min_{x_t, v_{t1}, v_{t4}} \|(v_{t1}^\top, v_{t4}^\top)^\top\|_1 \\ A_t x_t + v_{t1} \geq b_t(\xi_{tj}) - B_t x_{t-1} \\ \overrightarrow{F}_t x_t + v_{t4} \geq \overrightarrow{F}_t \xi_{[t]j} + \overrightarrow{f}_t \\ x_t, v_{t1}, v_{t4} \geq 0 \end{array} \right.$$

must be 0 for every $j = 1, \ldots, q_t$, where ξ_{tj} is given by (6) and where $\xi_{[t]j}$ is the useful history of the process at time step t given the history $\xi_{[t-1]}$ up to time step $t-1$ and the realization of ξ_t obtained with history $\xi_{[t-1]}$ and realization η_{tj} of η_t. Feasibility cuts for the convex process model are given in the following proposition:

Theorem 2.11 (Feasibility cuts-Convex process model)**.** *Consider optimization problem* (33) *for some $j \in \{1, \ldots, q_t\}$ and with $\xi_{[t-1]}$ and x_{t-1} respectively replaced by $\xi_{[t-1]}^k$ and x_{t-1}^k for some*

$k \in \{(i-1)H+1, \ldots, iH\}$. *Assume that the optimal value of this problem is positive and that Assumptions (A1), (A2), (A3), (A4), (A5), and (A6) hold. Let* σ_t^{kj} *be a row vector of optimal dual variables for feasibility cuts and let* π_{t2}^{kj} *be a row vector of optimal dual variables for the inequality constraints. A feasibility cut can be built for* x_{t-1} *as follows. We add to* $\overrightarrow{F}_{t-1}$ *the row* $\pi_{t2}^{kj} B_t$. *We add to* $\overrightarrow{F}_{t-1}$ *the row vector* $(\tilde{F}_{t-1,1}, \ldots, \tilde{F}_{t-1,M})$ *with* $\tilde{F}_{t-1,m} \in \mathbb{R}^{1 \times s_{t,m}}$ *and where* $\tilde{F}_{t-1,m}$ *is given by the expression* (23) *where* E, ρ, *and* iH *are respectively replaced by* F, σ, *and the current number of feasibility cuts minus 1. Finally, we add to* $\overrightarrow{f}_{t-1}$ *the quantity* $\sigma_t \overrightarrow{f}_t + \pi_{t2}^{kj} b_t(\xi_{tj}^k)$ *plus the expression* (24) *where* E, ρ, *and* iH *are respectively replaced by* F, σ, *and the current number of feasibility cuts minus 1. In this expression,* ξ_{tj}^k *is given by* (22).

Remark 2.12. *We see that the feasibility cut sharing property is possible due to the interstage independence of* η_t*: we have used the fact that all the nodes of a given stage* t *have the same set of realizations of* η_{t+1} *at their children nodes.*

We mention that infeasibility in the forward pass could also be handled by penalization of slack variables. Slack variables are added in such a way that the modified problem satisfies relatively complete recourse. As a result, feasibility cuts are not necessary anymore for this problem. However, unless guided by some physical interpretation intrinsic to the particular application (for instance when the penalty corresponds to a fee paid by the company to the clients or to the government for each unit of unsatisfied demand), the choice of the penalty parameters remains a delicate matter and can substantially distort the recourse functions. Using feasibility cuts amounts to eliminating all solutions with postive slack variables which makes sense when such solutions cannot be implemented.

In the absence of relatively complete recourse, the approximate policy obtained with SDDP may not be feasible, even with the feasibility cuts built. The feasibility cuts will simply allow us to avoid infeasible states when building the policy. However, when simulating the policy on a set of scenarios, infeasibility can arise. To remedy that, we cannot use feasibility cuts in the simulation phase since we only go forward. We thus need to use slack variables penalized in the objective in the simulation phase. In that context, numerical results could be performed to compare the quality of policies built using on the one hand feasibility cuts and on the other hand slack variables penalized in the objective.

2.6. **SDDP: stopping rule and algorithm.** In the backward pass, for the first time step, the first stage problem is solved using the recourse function $\mathcal{Q}_2^i \leq \mathcal{Q}_2$. Since Assumption (A3) holds, the optimal value of this problem is finite and provides a lower bound $z_{\inf}$ for the optimal mean cost. In our numerical experiments in Section 4 and in Figure 3 representing the SDDP algorithm, the algorithm is stopped after this lower bound $z_{\inf}$ has stabilized. Two other stopping criterion are discussed in [19] and [4].

Using the previous developments, DOASA algorithm for solving (1) that handles infeasibilities in the forward step and with an interstage dependent process of form (8) for (ξ_t) can be formulated as in Figure 3. In this figure, the fast-forward fast-backward tree traversing strategy [21] is used. A discussion on alternative tree traversing strategies (*Shuffle*, *Cautious*) can be found in [12] for instance.

3. SDDP for some risk-averse interstage dependent stochastic programs

For general stochastic programs, two recent papers [19], [5] have introduced risk-averse recourse functions and have proposed to use SDDP to obtain approximations of these functions in the special case of stochastic linear programs. In [19], the recourse functions are based on a risk-averse nested formulation of the problem defined in terms of conditional risk mappings. This methodology is applied in [14] (resp. [20]) to an hydro-thermal scheduling problem in the New Zealand (resp.

Step 0: **INITIALIZATION.** Set $i = 1$ (iteration number), $z_{\texttt{inf\_new}} = z_{\texttt{inf\_old}} = -\infty$, and all E_t^0, $\tilde{E}_t^0$, and e_t^0 to 0 for $t = 2, \ldots, T+1$. Fix a confidence level $\varepsilon > 0$. Go to Step 1.

Step 1: **FORWARD PASS.**
Sample H scenarios $(\xi_1^k, \ldots, \xi_T^k), k = (i-1)H+1, \ldots, iH$.
For $k = (i-1)H+1, \ldots, iH$,
  $t = 1$.
  While $(t \le T)$
    Back=FALSE. %We check the feasibility of x_{t-1}^k:
    For $j = 1, \ldots, q_t$,
      Solve optimization problem (32) with $\xi_{[t-1]}$ and x_{t-1} respectively replaced by $\xi_{[t-1]}^k$ and x_{t-1}^k.
      If the optimal value of this problem is positive and $t = 1$ then stop: the problem is infeasible.
      Else if this optimal value is positive and $t > 1$ then build a feasibility cut for x_{t-1} at stage $t-1$ using Theorem 2.10.
        Back=TRUE.
      End If
    End For
    If (Back=TRUE)
      $t \leftarrow t - 1$.
    Else
      Solve problem $[AP_t^{i,k}]$, store an optimal solution x_t^k of this problem and do $t \leftarrow t + 1$.
    End If
  End While
End For

Step 2: **BACKWARD PASS.**
For $t = T+1$ down to 2,
  For $k = (i-1)H+1, \ldots, iH$,
    If $(t = T+1)$ then set $E_{t-1}^k, \tilde{E}_{t-1}^k$, and e_{t-1}^k to 0.
    Else
      For each $j \in \{1, \ldots, q_t\}$ such that $\eta_{tj} \in \Omega_t^k$,
        Compute $\mathcal{Q}_t^i(x_{t-1}^k, \xi_{[t-1]}^k, \eta_{tj})$ given by (18) and store optimal dual multipliers $(\lambda_t^{kj}, \pi_{t1}^{kj}, \pi_{t2}^{kj}, \rho_t^{kj})$ in $\mathcal{M}_t^i$.
      End For
    End If
  End For
  If $(t \le T)$ then
    For $k = (i-1)H+1, \ldots, iH$,
      For each $j \in \{1, \ldots, q_t\}$ such that $\eta_{tj} \notin \Omega_t^k$,
        Compute $(\lambda_t^{kj}, \pi_{t1}^{kj}, \pi_{t2}^{kj}, \rho_t^{kj})$ given by (25).
      End For
      Form a cut for $\mathcal{Q}_t$ of the form $\theta_{t-1}^k + E_{t-1}^k x_{t-1} \ge \tilde{E}_{t-1}^k \xi_{[t-1]} + e_{t-1}^k$ with $E_{t-1}^k, \tilde{E}_{t-1}^k$, and e_{t-1}^k given in Proposition 2.9.
    End For
  End If
End For
$z_{\texttt{inf\_old}} = z_{\texttt{inf\_new}}$.
Set $z_{\texttt{inf\_new}}$ to the optimal value $z_{\inf}$ of the first stage problem.
Go to Step 3.

Step 3: **STOPPING RULE.**
If $z_{\texttt{inf\_new}} < z_{\texttt{inf\_old}}(1 + \varepsilon)$ then stop.
Else $i \leftarrow i + 1$ and go to Step 1.
End If

FIGURE 3. DOASA algorithm without relatively complete recourse for solving ID-SLP (1) with a stochastic process (ξ_t) satisfying (8).

Brazilian) electricity system. In [5], the class of multiperiod extended polyhedral risk measures is introduced and studied. In particular, this class is shown to be appropriate for deriving risk-averse DP equations. Conditions are also given on the multiperiod risk measure chosen to guarantee the convergence of SDDP in this risk- averse setting. Taking as a special case of multiperiod risk measure, a convex combination of the expectation of the total cost and of Conditional Value-at-Risks of partial costs, we obtain the risk-averse problem

$$
\begin{array}{l}
\inf f_1(x_1) + \Gamma_1 \mathbb{E}[\sum_{t=2}^{T} f_t(x_t)] + \sum_{t=2}^{T} \Gamma_t \mathrm{CVaR}^{\varepsilon_t}(\sum_{k=2}^{t} f_k(x_k)) \\
A_t x_t \geq b_t(\xi_t) - B_t x_{t-1}, \text{ a.s., } t = 1, \ldots, T, \\
C_t x_t = D_t \xi_t - E_t x_{t-1}, \text{ a.s., } t = 1, \ldots, T, \\
x_t \geq 0, \text{ a.s., } , x_t \in L_p(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^{k_t}), \ t = 1, \ldots, T,
\end{array} \tag{34}
$$

where confidence levels $\varepsilon_t \in (0,1)$ and coefficients Γ_t are nonnegative and sum up to one[5]. In the case when $\Gamma_1 = 1$, problem (34) boils down to non-risk-averse problem (1) considered in Section 2. Using the minimization formula from [16] for the CVaR, [5] provides for model (34) the DP equations

$$
\begin{array}{l}
\inf\limits_{x_1, w_{2:T}} f_1(x_1) + \sum_{t=2}^{T} \Gamma_t w_t + \mathcal{Q}_2(x_1, \xi_{[1]}, z_1, w_2, \ldots, w_T) \\
A_1 x_1 \geq b_1(\xi_1) - B_1 x_0, \ C_1 x_1 = D_1 \xi_1 - E_1 x_0, \\
x_1 \geq 0, \ \ w_t \in \mathbb{R}, \ t = 2, \ldots, T,
\end{array} \tag{35}
$$

with $z_1 = 0$ and where for $t = 2, \ldots, T$, $\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}, z_{t-1}, w_{t:T})$ is given by

$$
\mathbb{E}_{\xi_t | \xi_{[t-1]}} \left(\begin{array}{l}
\inf\limits_{x_t, z_t} \delta_{tT} \Gamma_1 z_t + \dfrac{\Gamma_t}{\varepsilon_t}(z_t - w_t)^+ + \mathcal{Q}_{t+1}(x_t, \xi_{[t]}, z_t, w_{t+1:T}) \\
A_t x_t \geq b_t(\xi_t) - B_t x_{t-1}, \ C_t x_t = D_t \xi_t - E_t x_{t-1} \\
x_t \geq 0, \ \ z_t = z_{t-1} + f_t(x_t)
\end{array} \right) \tag{36}
$$

where δ_{tT} is the Kronecker delta and $\mathcal{Q}_{T+1} \equiv 0$. In [5], cuts are provided for these recourse functions for interstage independent SLP. The adaptations to our interstage dependent context are easily done using the developments of the previous section. In particular, note that for our risk-averse SDDP, the stopping criterion is a simple adaptation of the stopping criterion in the risk-neutral case. The interested reader can look at Figure 1 in [6] which gives a detailed description of a risk-averse SDDP for a model more general than (34). That description (written for an interstage independent process) provides in particular the computation of the stopping criterion.

In the next section, devoted to numerical simulations, approximations of these risk-averse recourse functions are used on a real-life application with an affine process model. For this reason and for the sake of completeness, the cuts needed to obtain approximations of the risk-averse recourse functions using SDDP are derived in Theorem 3.1 which follows for the affine process model. Before stating this theorem, we need some more notation and remarks. First, lower bounding approximations $\mathcal{Q}_t^i$ of $\mathcal{Q}_t$ now have the form

$$
\mathcal{Q}_t^i(x_{t-1}, \xi_{[t-1]}, z_{t-1}, w_{t:T}) = \max_{j=0,1,\ldots,iH} [-E_{t-1}^j x_{t-1} + \tilde{E}_{t-1}^j \xi_{[t-1]} - Z_{t-1}^j z_{t-1} + \sum_{\tau=1}^{T-t+1} W_{t-1}^{j\tau} w_{t+\tau-1} + e_{t-1}^j]
$$

with $Z_{t-1}^j, W_{t-1}^{j\tau} \in \mathbb{R}$. Next, notice that risk-averse DP equations (35)-(36) involve additional first stage variables $w_2, \ldots, w_T$ as well as partial cost variables $z_1, \ldots, z_T$. When applying SDDP on these DP equations, with respect to the previous section, at iteration i of SDDP, the forward pass

[5]Risk measures are defined on random variables representing costs, contrary to [5] where they are defined on random variables representing incomes. We easily switch from one setting to another since an income is the opposite of a cost.

additionally computes first stage decisions $w_2^i, \ldots, w_T^i$ as well as partial costs $z_1^k, \ldots, z_T^k$ on scenario $k = (i-1)H+1, \ldots, iH$. As a result, in the backward pass of iteration i, H cuts are computed for $\mathcal{Q}_t$ at $(x_{t-1}^k, \xi_{[t-1]}^k, z_{t-1}^k, w_t^i, \ldots, w_T^i), k = (i-1)H+1, \ldots, iH$. We can bound from below $\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}, z_{t-1}, w_{t:T})$ by $\mathbb{E}_{\eta_t}[\mathcal{Q}_t^i(x_{t-1}, \xi_{[t-1]}, z_{t-1}, w_{t:T}, \eta_t)]$ with $\mathcal{Q}_t^i(x_{t-1}, \xi_{[t-1]}, z_{t-1}, w_{t:T}, \eta_t)$ given as the optimal value of the following linear program:

$$
\begin{array}{l}
\inf\limits_{x_t, \theta_{t1}, \theta_{t2}, \theta_{t3}, v_t} \;\; \theta_{t3} + \dfrac{\Gamma_t}{\varepsilon_t} v_t + \theta_{t2} \\
\theta_{t3} - \delta_{tT}\Gamma_1 \theta_{t1} \geq \delta_{tT}\Gamma_1 z_{t-1} \\
A_t x_t \geq b_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_t + \Theta_t) - B_t x_{t-1} \\
C_t x_t = D_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_t + \Theta_t) - E_t x_{t-1} \\
-\beta_t x_t + \theta_{t1} e \geq \alpha_t \\
v_t - \theta_{t1} \geq z_{t-1} - w_t \\
\overrightarrow{E}_t^i x_t + \theta_{t2} e + \theta_{t1} \overrightarrow{Z}_t^i \geq \overrightarrow{E}_t^i \left(\tilde{\Phi}_t \xi_{[t-1]} + \tilde{\Psi}_t \eta_t + \tilde{\Theta}_t \right) - z_{t-1} \overrightarrow{Z}_t^i + \sum_{\tau=1}^{T-t} \overrightarrow{W}_t^{i\tau} w_{t+\tau} + \overrightarrow{e}_t^i \\
x_t \geq 0, v_t \geq 0
\end{array}
\tag{37}
$$

where $\overrightarrow{W}_t^{i\tau}$ (resp. $\overrightarrow{Z}_t^i$) is the column vector whose $(j+1)$th component is $W_t^{j\tau}$ (resp. Z_t^j) for $j = 0, \ldots, iH$. We denote by $\pi_{t0}^{kj}, \pi_{t2}^{kj}, \pi_{t1}^{kj}, \lambda_t^{kj}, \mu_t^{kj}$, and ρ_t^{kj}, the (row vectors) optimal Lagrange multipliers associated to respectively the first 6 groups of constraints for the problem defining $Q_t^i(x_{t-1}^k, \xi_{[t-1]}^k, z_{t-1}^k, w_{t:T}^i, \eta_{tj})$. With this notation, the following theorem provides the cuts computed for $\mathcal{Q}_t$ at iteration i:

Theorem 3.1. *Let $\mathcal{Q}_t, t = 2, \ldots, T+1$, be the risk-averse recourse functions given by (36) and let Assumptions (A1), (A2), (A3), and (A7) hold. In the backward pass of iteration i of the SDDP algorithm, H valid cuts for these recourse functions are given as follows: for $t = T+1$, we set $E_{t-1}^k, \tilde{E}_{t-1}^k, Z_{t-1}^k, W_{t-1}^{k\tau}$, and e_{t-1}^k to 0 for $k = (i-1)H+1, \ldots, iH$. For $t = 2, \ldots, T$ and $k = (i-1)H+1, \ldots, iH$, $E_{t-1}^k, \tilde{E}_{t-1}^k$, and e_{t-1}^k are given by Theorem 2.4 and*

$$
Z_{t-1}^k = \sum_{j=1}^{q_t} p(t,j) \left[-\pi_{t0}^{kj} \delta_{tT} \Gamma_1 - \mu_t^{kj} + \rho_t^{kj} \overrightarrow{Z}_t^i \right], \;\; W_{t-1}^{k1} = -\sum_{j=1}^{q_t} p(t,j) \mu_t^{kj}, \tag{38}
$$

$$
W_{t-1}^{k\tau} = \sum_{j=1}^{q_t} p(t,j) \rho_t^{kj} \overrightarrow{W}_t^{i\tau-1}, \; \tau = 2, \ldots, T-t+1. \tag{39}
$$

4. Numerical experiment

4.1. **Power system data and policies.** We consider a hydro-thermal power system operating over an horizon of 10 years, discretized in $T = 120$ time steps, from January 2005 to December 2014. Most of the data was made available by CEPEL and corresponds to part of Brazil's power system, represented by 4 different subsystems that can trade energy in the form of import-export exchanges. Each subsystem, South-East (SE), South (S), North-East (NE), and North (N), corresponds to a geographical region; some energy exchanges between the N, NE, and SE subsystems make use of a fifth, fictitious, node (F). In a specific subsystem, a single reservoir aggregates all the hydro-power, while thermal generation is considered individually: there are 24, 14, 6, and 0 thermal plants in the SE (the largest one), S, NE, and N subsystems, respectively.

The total monthly demand is 54900 MWMonth [6], taken constant over the horizon. Each reservoir critical level was set to 20% of the maximum level of the reservoir, for all time steps.

[6] We adopt the convention 1 MWMonth= $\frac{365.25 \times 24}{12}$ MWh= 730.5 MWh

The objective function is given by the total thermal operating cost (ranging between R$ 6.27 per MWh and R$ 1047 per MWh) plus load shedding (set at R$ 4170.44 per MWh). Hydro plants operating cost is negligible while unnecessary spillage and exchanges are avoided by introducing penalties and trading costs between subsystems.

Following the lines of [11], the inflows in each reservoir are modeled by a periodic autoregressive model of form (8) (Assumption (A7) holds). The parameters of each model were estimated based on historical data from 1931 to 2005, with one important modification, relative to standard deviations. Namely, we reduced the estimated value of $\sigma_t^{\eta}(m)$ because, with the original estimations, the model generated too many negative water inflows that have no meaningful physical interpretation. Due to this modification, our results should be interpreted as an illustration of our methodology, rather than reflecting the real behavior of the Brazilian power system. This distribution of inflows is discretized to generate a scenario tree such that for all stage $t < T$, a given node of the scenario tree for this stage has 20 children nodes. With the notation of (5), we thus have $q_t = 20$ for $t = 2, \ldots, T$.

Our analysis compares two different policies:

- `SDDP`: a usual multistage risk-neutral policy that approximates recourse functions by SDDP as in Section 2;
- `RA-SDDP`: the risk-averse approach from Section 3 with $\varepsilon_t = 0.1$, $\Gamma_T = 1 - \Gamma_1 = 0.3$ (i.e., $\Gamma_t = 0,\ t = 2, \ldots, T-1$), and with risk-averse recourse functions $\mathcal{Q}_{t+1}$ approximated using SDDP.

The policies are compared in a simulation phase that uses 500 streamflow scenarios generated from the continuous distribution of inflows. Note that the approximate recourse functions are obtained solving by SDDP a Sample Average Approximation (SAA) problem associated to the scenario tree previously mentionned. As a result, with probability one, no scenario from the simulation phase is a scenario from the scenario tree.

Due to the high computational effort required by `RA-SDDP`, as in [14], SDDP is run for risk-averse model `RA-SDDP` taking the number of iterations necessary for SDDP to converge in the risk-neutral setting `SDDP`. At each iteration, $H = 200$ scenarios are generated in the forward passes of `SDDP` and `RA-SDDP`.

The implementation was done in Matlab, using Mosek's optimization library to solve linear programming problems (`http://www.matlab.com` and `http://www.mosek.com`).[7] The state vectors defined in Section 2.2 were used and convergence was obtained after 11 iterations[8] after observing a stabilization of the lower bound (increase of the lower bound inferior to 1%). The evolution of the upper bound (computed as in [19]) and of the lower bound along the iterations of the risk-neutral version of SDDP are reported in Figure 4. In this context, each approximate recourse function is built from 2201 cuts obtained in the backward passes.

4.2. **Distribution of the cost.** Since for model (34), we chose $\varepsilon_t = 0.1$ and $\Gamma_T = 1 - \Gamma_1 = 0.3$ for our simulations, `RA-SDDP` aims at decreasing the cost of the average of the $0.1 \times 500 = 50$ scenarios of highest cost. This model thus seeks to avoid peaks in the *total* cost (sum of the individual costs for all stages). For this reason, to measure the impact of the introduction of aversion to risk, we compare the distribution of the total cost for both policies.

We first provide for these policies in Table 3 the mean and the empirical standard deviation (s.d.) of the whole system total cost over the 500 scenarios, as well as the corresponding VaR p%, for $p = 1, 5, 10$, and 90, where VaR p% is the (1-p/100)-quantile of the empirical distribution of the cost. We observe that the mean total cost is higher for `RA-SDDP` (increase of about 23%). It can also

[7]The runs were done on a Dell PowerEdge 2900 server with 2 CPUs Intel Xeon E5345 (2.33 GHz, 8M of cache memory, 1333 MHz FSB), running under CentOS release 5, with 48 GB of RAM.

[8]The computational time was approximately 4 weeks.

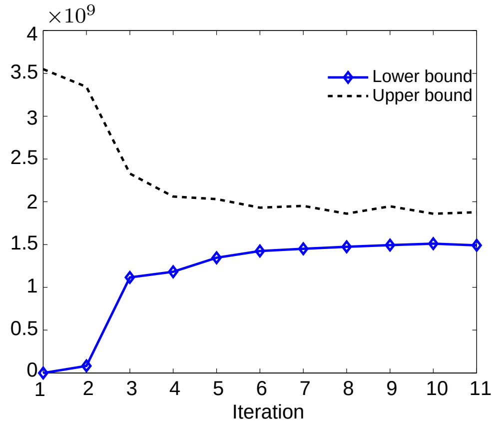

FIGURE 4. Upper and lower bounds evolution for the risk-neutral version of SDDP (at each iteration, 200 cuts are built).

TABLE 3. Measures of central tendency and of dispersion of the total cost (R$)

Output	SDDP	RA SDDP
Mean	1.693×10^9	2.087×10^9
s.d.	1.293×10^9	8.755×10^8
VaR 1%	6.864×10^9	4.543×10^9
VaR 5%	3.953×10^9	3.863×10^9
VaR 10%	3.205×10^9	3.358×10^9
VaR 90%	6.850×10^8	1.158×10^9

be seen that the risk-averse version of `SDDP` we tested results in lower standard deviation as well as lower VaR 1% and 5% of the total cost. Finally, with `RA-SDDP`, VaR 10% is slightly higher while VaR 90% is significantly higher. More precisely, from Figure 5 where the distributions of the policies are compared, we can add that on a majority of scenarios, `RA-SDDP` generation cost is higher than `SDDP` generation cost. However, with `SDDP`, there are more than 10 scenarios with cost above 5×10^9 whereas all scenarios have cost below this value for `RA-SDDP`. This is partly due to the fact that the portion of demand left unsatisfied is larger with `SDDP`: the percentage of unsatisfied demand is very small for each policy but is larger for `SDDP`: 2.62×10^{-3}% against 1.89×10^{-3}% for `RA-SDDP`.

As a result, the risk-averse model allows us to reduce the number of very high cost scenarios at the expense of an increase in the mean total cost.

4.3. **Reservoir levels.** Another important indicator to measure the impact of risk-aversion on the policy is the volume of the reservoirs. We report in Figure 6 the mean and 0.05- and 0.95-quantiles for the equivalent reservoir level (sum of all reservoir volumes). Over all time steps and scenarios, `RA-SDDP` uses less water than `SDDP`, especially after the third year. As a result, with the risk-averse policy, the mean and 0.95-quantiles of the equivalent reservoir level are much higher. This also explains that load shedding decreases with `RA-SDDP` since it occurs when the system does not use enough water and is not able to satisfy the demand with the remaining thermal plants (of limited capacity). Finally, since for all policies the demand is satisfied for nearly all time steps and scenarios, the thermal generation merely complements the hydro-generation to attain the demand level for each policy.

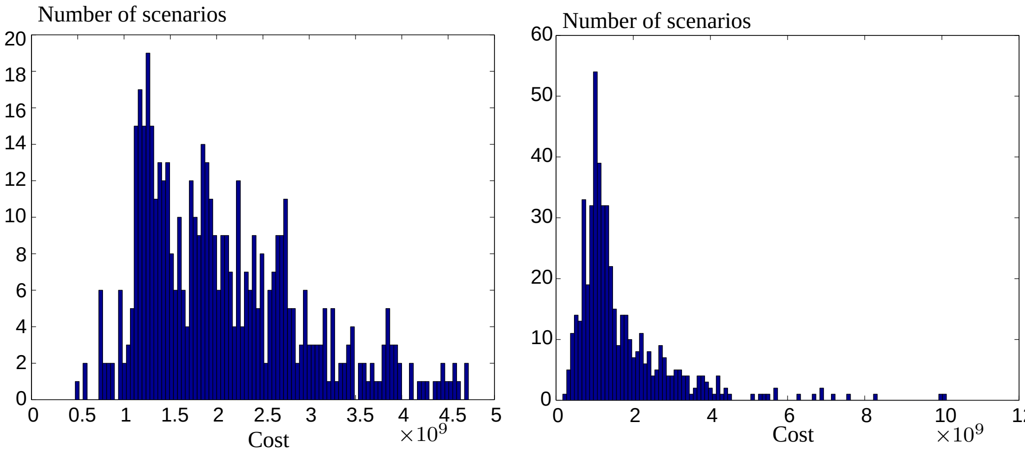

FIGURE 5. Empirical distribution of the total cost for `RA-SDDP` (on the left) and `SDDP` (on the right) policies.

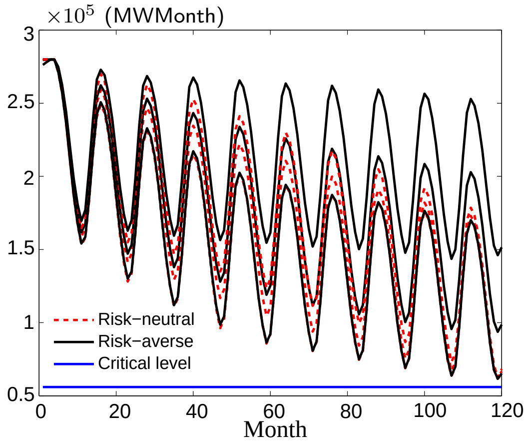

FIGURE 6. Equivalent reservoir level evolution (mean and 0.05- and 0.95- quantiles).

5. CONCLUSION

We have explained how to apply the SDDP algorithm both in risk-neutral and risk-averse settings for some interstage dependent stochastic linear programs for which relatively complete recourse does not hold. Considering two statistical frameworks for the underlying stochastic process, namely the affine process model and the convex process model, we provided conditions that guarantee the convexity of the recourse functions and gave formulas for the feasibility and optimality cuts that are built in respectively the forward and backward passes of the SDDP algorithm. We have also shown how to share these cuts (both feasibility and optimality) between nodes of the same stage.

We then presented numerical results that compare for a real-life application the performance of a risk-neutral model with risk-averse model (34) when recourse functions are approximated using SDDP. We have seen that the risk-averse model allows us to avoid high 0.99-quantiles and to decrease the standard deviation of the total cost. However, a price has to be paid for this risk aversion which is the increase in the policy average total cost. A visible effect of risk aversion for our application is also observed comparing the evolution of hydro reservoir levels. The risk-averse model keeps more water in the reservoirs, resulting in less load shedding, which appears as another appealing feature. Further numerical experiments could analyze the behavior of the risk-averse model for different values of parameters (ε_t) and (Γ_t) and compare this risk-averse version of SDDP with the one in [19].

In a forthcoming work, we intend to explain how one can extend the SDDP algorithm and related methods when the number of stages is random.

Acknowledgments

The author's research was partially supported by an FGV grant, PRONEX-Optimization, and CNPq grant No.382.851/07-4. We would like to thank the reviewers and the Associate Editor for beneficial comments and suggestions.

References

[1] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer-Verlag, New York, 1997.
[2] J.R. Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, 33:989–1007, 1985.
[3] J.R. Birge and C. J. Donohue. The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse. *Algorithmic of Operations Research*, 1:20–30, 2001.
[4] T. Homem de Mello, V.L. de Matos, and E. C. Finardi. Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling. *Energy Systems*, 2:1–31, 2011.
[5] V. Guigues and W. Römisch. Sampling-based decomposition methods for multistage stochastic programs based on extended polyhedral risk measures. *SIAM Journal on Optimization*, 22:286–312, 2012.
[6] V. Guigues and W. Römisch. SDDP for multistage stochastic linear programs based on spectral risk measures. *Operations Research Letters*, 40:313–318, 2012.
[7] V. Guigues and C. Sagastizábal. Exploiting the structure of autoregressive processes in chance-constrained multistage stochastic linear programs. *Operations Research Letters*, 40:478–483, 2012.
[8] V. Guigues and C. Sagastizábal. The value of rolling horizon policies for risk-averse hydro-thermal planning. *European Journal of Operational Research*, 217:129–140, 2012.
[9] V. Guigues and C. Sagastizábal. Risk-averse feasible policies for large-scale multistage stochastic linear programs. *Mathematical Programming*, 138:167–198, 2013.
[10] G. Infanger and D. Morton. Cut sharing for multistage stochastic linear programs with interstage dependency. *Mathematical Programming*, 75:241–256, 1996.
[11] M.E. Maceira and J. Damázio. The use of PAR(p) model in the stochastic dual dynamic programming optimization scheme used in the operation planning of the Brazilian hydropower system. *VIII International Conference on Probabilistic Methods Applied to Power Systems*, pages 397–402, Sept. 2004.
[12] D.P. Morton. An enhanced decomposition algorithm for multistage stochastic hydroelectric scheduling. *Annals of Operations Research*, 64:211–235, 1996.
[13] M.V.F. Pereira and L.M.V.G Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52:359–375, 1991.
[14] A. Philpott and V. de Matos. Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. *European Journal of Operational Research*, 218:470–483, 2012.
[15] A. B. Philpott and Z. Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36:450–455, 2008.
[16] R.T Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *J. of Banking and Finance*, 26(7):1443–1471, 2002.
[17] A. Ruszczyński and A. Shapiro. Conditional risk mappings. *Mathematics of Operations Research*, 31:544–561, 2006.
[18] A. Ruszczyński and A. Shapiro. Optimization of convex risk functions. *Mathematics of Operations Research*, 31:433–452, 2006.
[19] A. Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209:63–72, 2011.
[20] A. Shapiro, W. Tekaya, J.P. da Costa, and M.P. Soares. Risk neutral and risk averse stochastic dual dynamic programming method. *European Journal of Operational Research*, 224:375–391, 2013.

[21] R.J. Wittrock. Advances in a nested decomposition algorithm for solving staircase linear programs. *Technical Report SOL 83-2, Systems Optimization Laboratory, Department of Operations Research, Stanford University (Stanford, CA, 1983).*

Appendix

Proof of Theorem 2.4. By duality, $Q_t^i(x_{t-1}, \xi_{[t-1]}, \eta_{tj})$ may be expressed as the optimal value of the following linear program (due to Assumption (A3) the dual and the primal have the same finite optimal value):

$$
\begin{array}{l}
\displaystyle\max_{\lambda_t, \pi_{t1}, \pi_{t2}, \rho_t} g_1(\lambda_t, \pi_{t1}, \pi_{t2}, \rho_t, \eta_{tj}) \\
C_t^\top \pi_{t1}^\top + A_t^\top \pi_{t2}^\top + \overrightarrow{E}_t^{i\,\top} \rho_t^\top \leq \beta_t^\top \lambda_t^\top \\
\lambda_t e = 1,\ \rho_t e = 1,\ \rho_t \geq 0,\ \lambda_t \geq 0,\ \pi_{t2} \geq 0
\end{array} \tag{40}
$$

where the objective function is given by

$$
\begin{array}{l}
g_1(\lambda_t, \pi_{t1}, \pi_{t2}, \rho_t, \eta_{tj}) = \lambda_t \alpha_t + \rho_t \left(\overrightarrow{e}_t^i + \overrightarrow{\tilde{E}}_t^i (\tilde{\Phi}_t \xi_{[t-1]} + \tilde{\Psi}_t \eta_{tj} + \tilde{\Theta}_t) \right) \\
+\pi_{t2} \left(b_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_{tj} + \Theta_t) - B_t x_{t-1} \right) + \pi_{t1} \left(D_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_{tj} + \Theta_t) - E_t x_{t-1} \right).
\end{array} \tag{41}
$$

For problem (40), optimal solutions are extremal points of the feasible set. Further, the feasible set neither depends on x_{t-1} nor on $\xi_{[t-1]}$ and for any $(x_{t-1}, \xi_{[t-1]})$, row vectors $\lambda_t^{kj}, \pi_{t1}^{kj}, \pi_{t2}^{kj}, \rho_t^{kj}$ are extremal points of the feasible set of problem $Q_t^i(x_{t-1}, \xi_{[t-1]}, \eta_{tj})$ expressed as (40). It follows that $Q_t^i(x_{t-1}, \xi_{[t-1]}, \eta_{tj})$ is bounded from below by

$$
\begin{array}{l}
g_1(\lambda_t^{kj}, \pi_{t1}^{kj}, \pi_{t2}^{kj}, \rho_t^{kj}, \eta_{tj}) = \lambda_t^{kj} \alpha_t + \rho_t^{kj} \left(\overrightarrow{e}_t^i + \overrightarrow{\tilde{E}}_t^i (\tilde{\Phi}_t \xi_{[t-1]} + \tilde{\Psi}_t \eta_{tj} + \tilde{\Theta}_t) \right) \\
+\pi_{t2}^{kj} \left(b_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_{tj} + \Theta_t) - B_t x_{t-1} \right) + \pi_{t1}^{kj} \left(D_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_{tj} + \Theta_t) - E_t x_{t-1} \right)
\end{array} \tag{42}
$$

for $j = 1, \ldots, q_t$. Next, from the convexity of b_{ti}, we obtain

$$
b_{ti}(\Phi_t \xi_{[t-1]} + \Psi_t \eta_{tj} + \Theta_t) \geq b_{ti}(\xi_{tj}^k) + s_{ti}^b(\xi_{tj}^k)^\top \Phi_t(\xi_{[t-1]} - \xi_{[t-1]}^k),\ i = 1, \ldots, \ell_t, \tag{43}
$$

and since $\pi_{t2}^{kj} \geq 0$, we have

$$
\pi_{t2}^{kj} b_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_{tj} + \Theta_t) \geq \pi_{t2}^{kj} \left[b_t(\xi_{tj}^k) + s_t^b(\xi_{tj}^k) \Phi_t(\xi_{[t-1]} - \xi_{[t-1]}^k) \right]. \tag{44}
$$

Plugging (44) into lower bound (42) for $Q_t^i(x_{t-1}, \xi_{[t-1]}, \eta_{tj})$ and since $\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]})$ is bounded from below by $\mathbb{E}_{\eta_t}[Q_t^i(x_{t-1}, \xi_{[t-1]}, \eta_t)] = \sum_{j=1}^{q_t} p(t,j) Q_t^i(x_{t-1}, \xi_{[t-1]}, \eta_{tj})$, we obtain a cut of the form $\theta_{t-1}^k + E_{t-1}^k x_{t-1} \geq \tilde{E}_{t-1}^k \xi_{[t-1]} + e_{t-1}^k$ and the result follows. □

Proof of Theorem 2.8. We show by induction, from $t = T+1$ down to $t = 2$, that the announced cuts are valid and that $\tilde{E}_t^k \geq 0$ for $t = 1, \ldots, T$, and $k = 0, 1, \ldots, iH$. For $t = T$, we have $\mathcal{Q}_{T+1}^i = \mathcal{Q}_{T+1} \equiv 0$. As a result, all components of $E_T^k, \tilde{E}_T^k$, and e_T^k are null for $k = 0, 1, \ldots, iH$. In particular, we have that $\tilde{E}_t^k \geq 0$. This achieves the first step of the induction.

Let us now assume that for some $t \in \{2, \ldots, T\}$, valid cuts have been built for $\mathcal{Q}_{\ell+1}, \ell = t, \ldots, T$, according to the formulas given in the theorem for $E_\ell^k, \tilde{E}_\ell^k$, and e_ℓ^k, $\ell = t, \ldots, T$, $k = 0, 1, \ldots, iH$, with all $\tilde{E}_\ell^k \geq 0$.

We have $\mathcal{Q}_t(x_{t-1}, \xi_{[t-1]}) \geq \sum_{j=1}^{q_t} p(t,j) \mathcal{Q}_t^i(x_{t-1}, \xi_{[t-1]}, \eta_{tj})$ with

$$
\mathcal{Q}_t^i(x_{t-1}, \xi_{[t-1]}, \eta_{tj}) \geq g_2(\lambda_t^{kj}, \pi_{t2}^{kj}, \rho_t^{kj}, \eta_{tj}) = \lambda_t^{kj} \alpha_t + \rho_t^{kj} \overrightarrow{e}_t^i - \pi_{t2}^{kj} B_t x_{t-1} + U_{tj}^k + V_{tj}^k + W_{tj}^k \tag{45}
$$

where

$$U_{tj}^k = \sum_{m=1}^{M} \sum_{\ell=1}^{s_{t+1,m}-1} \left[\sum_{w=0}^{iH} \rho_t^{kj}(w) \tilde{E}_{t,m}^{w}(\ell+1) \right] \xi_{t-\ell}(m), \tag{46}$$

$$V_{tj}^k = \sum_{m=1}^{M} \sum_{w=0}^{iH} \rho_t^{kj}(w) \tilde{E}_{t,m}^{w}(1) \xi_{tj}(m), \tag{47}$$

$$W_{tj}^k = \sum_{\ell=1}^{\ell_t} \pi_{t2}^{kj}(\ell) b_{t\ell}(\xi_{tj}). \tag{48}$$

Let us first bound from below W_{tj}^k. Using the convexity of $b_{t\ell}$, we obtain

$$b_{t\ell}(\xi_{tj}) \geq b_{t\ell}(\xi_{tj}^k) + s_{t\ell}^b(\xi_{tj}^k)^\top (\xi_{tj} - \xi_{tj}^k)$$

for every $\ell = 1, \ldots, \ell_t$. Using these inequalities and the fact that $\pi_{t2}^{kj} \geq 0$, we have

$$W_{tj}^k = \pi_{t2}^{kj} b_t(\xi_{tj}) \geq \pi_{t2}^{kj} b_t(\xi_{tj}^k) + \pi_{t2}^{kj} s_t^b(\xi_{tj}^k)(\xi_{tj} - \xi_{tj}^k). \tag{49}$$

Similarly, using the convexity of h_{tm}, we have

$$\begin{array}{l} \xi_{tj}(m) - \xi_{tj}^k(m) = h_{tm}(\xi_{t-1:t-p_t(m)}(m), \eta_{tj}(m)) - h_{tm}(\xi_{t-1:t-p_t(m)}^k(m), \eta_{tj}(m)) \\ \geq \displaystyle\sum_{w=1}^{p_t(m)} s_{tm}^h \left(\xi_{t-1:t-p_t(m)}^k(m), \eta_{tj}(m) \right)(w) \left(\xi_{t-w}(m) - \xi_{t-w}^k(m) \right) \end{array} \tag{50}$$

for every $m = 1, \ldots, M$. Using Assumption (A4), each component of each subgradient of $b_{t\ell}$ is nonnegative. As a result, all elements in matrix $s_t^b(\xi_{tj}^k)$ are nonnegative. Using this observation and relations (49) and (50), we obtain for W_{tj}^k the lower bound $\pi_{t2}^{kj} b_t(\xi_{tj}^k)$ plus

$$\sum_{m=1}^{M} \sum_{w=1}^{p_t(m)} \left[\sum_{\ell=1}^{\ell_t} \pi_{t2}^{kj}(\ell) s_{t\ell}^b(\xi_{tj}^k)(m) s_{tm}^h \left(\xi_{t-1:t-p_t(m)}^k(m), \eta_{tj}(m) \right)(w) \right] (\xi_{t-w}(m) - \xi_{t-w}^k(m)). \tag{51}$$

Let us now bound from below V_{tj}^k. Using relation (50) and the nonnegativeness of $\tilde{E}_{t,m}^w(1)$ (induction hypothesis) and of $\rho_t^{kj}(w)$, we obtain for V_{tj}^k the lower bound

$$\begin{array}{l} \displaystyle\sum_{m=1}^{M} \sum_{w=0}^{iH} \rho_t^{kj}(w) \tilde{E}_{t,m}^w(1) \sum_{u=1}^{p_t(m)} s_{tm}^h \left(\xi_{t-1:t-p_t(m)}^k(m), \eta_{tj}(m) \right)(u) \left(\xi_{t-u}(m) - \xi_{t-u}^k(m) \right) \\ + \displaystyle\sum_{m=1}^{M} \sum_{w=0}^{iH} \rho_t^{kj}(w) \tilde{E}_{t,m}^w(1) \xi_{tj}^k(m). \end{array} \tag{52}$$

Plugging into (45) relation (46) as well as lower bounds (51) and (52) for respectively $W_{tj}^k - \pi_{t2}^{kj} b_t(\xi_{tj}^k)$ and V_{tj}^k, we obtain for $\mathcal{Q}_t$ a cut of form $-E_{t-1}^k x_{t-1} + \tilde{E}_{t-1}^k \xi_{[t]} + e_{t-1}^k$ with the desired values of $E_{t-1}^k, \tilde{E}_{t-1}^k$, and e_{t-1}^k. If remains to check that all components of $\tilde{E}_{t-1}^k$ are nonnegative. We had already observed that for all functions b_{ti}, all components of all subgradients are nonnegative, due to Assumption (A4). The same remark holds for functions h_{tm}, due to Assumption (A6). By induction hypothesis, all coefficients $(\tilde{E}_{tm}^j(\ell))_{j,\ell,m}$ are nonnegative. Using the nonnegativity of these coefficients, as well as the nonnegativity of row vectors ρ_t^{kj} and π_{t2}^{kj}, together with the formula for $\tilde{E}_{t-1}^k$, we obtain that $\tilde{E}_{t-1}^k \geq 0,\ k = 0, 1, \ldots, iH$. □

Proof of Proposition 2.9. We show the cuts are valid when there are both equality and inequality constraints. A similar proof can be done when there are not equality constraints. Let $k \in \{(i-1)H+1, \ldots, iH\}$ and $j \in \{1, \ldots, q_t\}$. If $\eta_{tj} \in \Omega_t^k$ then $Q_t^i(x_{t-1}, \xi_{[t-1]}, \eta_{tj})$ is bounded from below by $g_1(\lambda_t^{kj}, \pi_{t1}^{kj}, \pi_{t2}^{kj}, \rho_t^{kj}, \eta_{tj})$ where the expression of g_1 is given by (40). Next, for every j such that $\eta_{tj} \notin \Omega_t^k$, since all $(\lambda_t, \pi_{t1}, \pi_{t2}, \rho_t) \in \mathcal{M}_t^i$ belong to the feasible set of problem $Q_t^i(x_{t-1}, \xi_{[t-1]}, \eta_{tj})$, we have $Q_t^i(x_{t-1}, \xi_{[t-1]}, \eta_{tj}) \geq g_1(\lambda_t, \pi_{t1}, \pi_{t2}, \rho_t, \eta_{tj})$ for every $(\lambda_t, \pi_{t1}, \pi_{t2}, \rho_t) \in \mathcal{M}_t^i$. As a result,

$$\begin{array}{lll} Q_t^i(x_{t-1}, \xi_{[t-1]}, \eta_{tj}) & \geq & \max_{(\lambda_t, \pi_{t1}, \pi_{t2}, \rho_t) \in \mathcal{M}_t^i} \; g_1(\lambda_t, \pi_{t1}, \pi_{t2}, \rho_t, \eta_{tj}) \\ & = & g_1(\lambda_t^{kj}, \pi_{t1}^{kj}, \pi_{t2}^{kj}, \rho_t^{kj}, \eta_{tj}) \text{ using (25).} \end{array}$$

We then conclude as in the proof of Theorem 2.4. □

Proof of Theorem 2.10. Let x_{t-1} be a feasible state at the end of time step $t-1$ at a given node of this time step with history $\xi_{[t-1]}$. Since for one of the son nodes, the realization of η_t is η_{tj}, the optimal value of (32) is 0. As a result, the optimal value of the dual of (32) is 0. This dual problem can be written

$$\left\{ \begin{array}{l} \displaystyle\max_{\pi_{t1}, \pi_{t2}, \sigma_t} \; f(\pi_{t1}, \pi_{t2}, \sigma_t) \\ C_t^\top \pi_{t1}^\top + A_t^\top \pi_{t2}^\top + \overrightarrow{F}_t^\top \sigma_t^\top \leq 0 \\ -e \leq \pi_{t1}^\top \leq e, \; 0 \leq \pi_{t2}^\top \leq e, \; 0 \leq \sigma_t^\top \leq e, \end{array} \right. \tag{53}$$

where the objective function f is given by

$$\begin{array}{l} \pi_{t1} \left[D_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_{tj} + \Theta_t) - E_t x_{t-1} \right] + \pi_{t2} \left[b_t(\Phi_t \xi_{[t-1]} + \Psi_t \eta_{tj} + \Theta_t) - B_t x_{t-1} \right] \\ + \sigma_t \left[\overrightarrow{F}_t(\tilde{\Phi}_t \xi_{[t-1]} + \tilde{\Psi}_t \eta_{tj} + \tilde{\Theta}_t) + \overrightarrow{f}_t \right]. \end{array}$$

For this dual problem, since the optimal value is 0 and since $(\pi_{t1}^{kj}, \pi_{t2}^{kj}, \sigma_t^{kj})$ is feasible, we obtain

$$0 \geq f(\pi_{t1}^{kj}, \pi_{t2}^{kj}, \sigma_t^{kj}).$$

We conclude using (43) and (44). □

Proof of Proposition 2.11. We follow the proofs of Theorems 2.8 and 2.10. □

Proof of Theorem 3.1. The proof is similar to the proof of Theorem 2.4. □

Vincent Guigues: Escola de Matemática Aplicada, Fundação Getulio Vargas, Praia de Botafogo, 190, Rio de Janeiro, Brazil, vguigues@fgv.br, IMPA, Instituto de Matemática Pura e Aplicada, 110 Estrada Dona Castorina, Jardim Botanico, Rio de Janeiro, Brazil, and, UFRJ, Escola Politécnica, Departamento de Engenharia Industrial, Ilha do Fundão, CT, Bloco F, Rio de Janeiro, Brazil