



HAL
open science

Exact converging bounds for Stochastic Dual Dynamic Programming via Fenchel duality

Vincent Leclère, Pierre Carpentier, Jean-Philippe Chancelier, Arnaud Lenoir,
François Pacaud

► **To cite this version:**

Vincent Leclère, Pierre Carpentier, Jean-Philippe Chancelier, Arnaud Lenoir, François Pacaud. Exact converging bounds for Stochastic Dual Dynamic Programming via Fenchel duality. 2018. hal-01744035v1

HAL Id: hal-01744035

<https://enpc.hal.science/hal-01744035v1>

Preprint submitted on 27 Mar 2018 (v1), last revised 18 Apr 2018 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exact converging bounds for Stochastic Dual Dynamic Programming via Fenchel duality

Vincent Leclère,^{*}Pierre Carpentier,[†]Jean-Philippe Chancelier,[‡]
Arnaud Lenoir[§]and François Pacaud[¶]

March 27, 2018

Abstract

The Stochastic Dual Dynamic Programming (SDDP) algorithm has become one of the main tools to address convex multistage stochastic optimal control problem. Recently a large amount of work has been devoted to improve the convergence speed of the algorithm through cut-selection and regularization, or to extend the field of applications to non-linear, integer or risk-averse problems. However one of the main downside of the algorithm remains the difficulty to give an upper bound of the optimal value, usually estimated through Monte Carlo methods and therefore difficult to use in the algorithm stopping criterion.

In this paper we present a dual SDDP algorithm that yields a converging exact upper bound for the optimal value of the optimization problem. Incidentally we show how to compute an alternative control policy based on an inner approximation of Bellman value functions instead of the outer approximation given by the standard SDDP algorithm. We illustrate the approach on an energy production problem involving zones of production and transportation links between the zones. The numerical experiments we carry out on this example show the effectiveness of the method.

1 Introduction

In this paper, we consider a multistage stochastic optimization problem, with continuous decision variables. We adopt the stochastic optimal control point of view, that is, we work with explicit control and state variables in order to deal with an explicit dynamics of the system and to obtain an interpretation of the multipliers associated to the dynamics.

1.1 Stochastic optimization problem in discrete time

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space, where Ω is the set of possible outcomes, \mathcal{A} the associated σ -field and \mathbb{P} the probability measure. We denote by $\llbracket 0, T \rrbracket$ the discrete optimization horizon $\{0, 1, \dots, T\}$, and we define upon it three processes $\mathbf{X} = \{\mathbf{X}_t\}_{t \in \llbracket 0, T \rrbracket}$, $\mathbf{U} = \{\mathbf{U}_t\}_{t \in \llbracket 1, T-1 \rrbracket}$ and $\boldsymbol{\xi} = \{\boldsymbol{\xi}_t\}_{t \in \llbracket 1, T \rrbracket}$ where for all t , $\mathbf{X}_t : \Omega \rightarrow \mathbb{R}^n$, $\mathbf{U}_t : \Omega \rightarrow \mathbb{R}^m$ and $\boldsymbol{\xi}_t : \Omega \rightarrow \mathbb{R}^\ell$ are random

^{*}Université Paris-Est, CERMICS (ENPC) – vincent.leclere@enpc.fr

[†]UMA, ENSTA ParisTech, Université Paris-Saclay – pierre.carpentier@ensta-paristech.fr

[‡]Université Paris-Est, CERMICS (ENPC) – chancelier@cermics.enpc.fr

[§]EDF – arnaud.lenoir@edf.fr

[¶]CERMICS (ENPC) - Efficacity – francois.pacaud@enpc.com

variables representing respectively the state, the control and the noise variables. The state process \mathbf{X} is assumed to follow the linear dynamics

$$\begin{aligned}\mathbf{X}_0 &= x_0, \\ \mathbf{X}_{t+1} &= A_t \mathbf{X}_t + B_{t+1} \mathbf{U}_{t+1} + C_{t+1} \boldsymbol{\xi}_{t+1} \quad \forall t \in \llbracket 0, T-1 \rrbracket,\end{aligned}$$

where x_0 is the given initial state at time 0 and where $A_t \in \mathbb{R}^{n \times n}$, $B_{t+1} \in \mathbb{R}^{n \times m}$ and $C_{t+1} \in \mathbb{R}^{n \times \ell}$ are given matrices. We moreover assume that both the control and the state variables are subject to bound constraints, that is,

$$\begin{aligned}\underline{u}_{t+1} \leq \mathbf{U}_{t+1} \leq \bar{u}_{t+1} & \quad \forall t \in \llbracket 0, T-1 \rrbracket, \\ \underline{x}_{t+1} \leq \mathbf{X}_{t+1} \leq \bar{x}_{t+1} & \quad \forall t \in \llbracket 0, T-1 \rrbracket,\end{aligned}$$

and thus take values in compact subsets of \mathbb{R}^n and \mathbb{R}^m respectively. We assume that the problem has a Hazard-Decision¹ information structure, that is, the decision at time t is taken knowing the noise that affects the system between t and $t+1$. Accordingly, the decision \mathbf{U}_{t+1} is a function of the uncertainties up to time $t+1$, which means that \mathbf{U}_{t+1} has to be measurable with respect to the σ -field \mathcal{F}_{t+1} generated by the uncertainties $(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_{t+1})$. We write this *non anticipativity constraint* as

$$\mathbf{U}_{t+1} \preceq \mathcal{F}_{t+1} \quad \forall t \in \llbracket 0, T-1 \rrbracket.$$

Furthermore, we assume that the state variable \mathbf{X}_t and the control variable \mathbf{U}_t satisfy a linear coupling constraint

$$D_t \mathbf{X}_t + E_{t+1} \mathbf{U}_{t+1} + G_{t+1} \boldsymbol{\xi}_{t+1} \leq 0 \quad \forall t \in \llbracket 0, T-1 \rrbracket,$$

where $D_t \in \mathbb{R}^{p \times n}$, $E_{t+1} \in \mathbb{R}^{p \times m}$ and $G_{t+1} \in \mathbb{R}^{p \times \ell}$ are given matrices. Finally, the cost incurred at each time $t \in \llbracket 0, T-1 \rrbracket$ is a linear function

$$a_t^\top \mathbf{X}_t + b_{t+1}^\top \mathbf{U}_{t+1},$$

with $a_t \in \mathbb{R}^n$ and $b_{t+1} \in \mathbb{R}^m$, and the cost incurred at the final time T is

$$K(\mathbf{X}_T),$$

where K is a polyhedral, hence convex lower semi-continuous, function.

Gathering all these elements, we get the following stochastic optimization problem:

$$\min_{\mathbf{X}, \mathbf{U}} \mathbb{E} \left[\sum_{t=0}^{T-1} (a_t^\top \mathbf{X}_t + b_{t+1}^\top \mathbf{U}_{t+1}) + K(\mathbf{X}_T) \right], \quad (1a)$$

$$\text{s.t. } \mathbf{X}_0 = x_0, \quad (1b)$$

$$\mathbf{X}_{t+1} = A_t \mathbf{X}_t + B_{t+1} \mathbf{U}_{t+1} + C_{t+1} \boldsymbol{\xi}_{t+1} \quad \forall t \in \llbracket 0, T-1 \rrbracket, \quad (1c)$$

$$\underline{u}_{t+1} \leq \mathbf{U}_{t+1} \leq \bar{u}_{t+1} \quad \forall t \in \llbracket 0, T-1 \rrbracket, \quad (1d)$$

$$\underline{x}_{t+1} \leq \mathbf{X}_{t+1} \leq \bar{x}_{t+1} \quad \forall t \in \llbracket 0, T-1 \rrbracket, \quad (1e)$$

$$D_t \mathbf{X}_t + E_{t+1} \mathbf{U}_{t+1} + G_{t+1} \boldsymbol{\xi}_{t+1} \leq 0 \quad \forall t \in \llbracket 0, T-1 \rrbracket, \quad (1f)$$

$$\mathbf{U}_{t+1} \preceq \mathcal{F}_{t+1} \quad \forall t \in \llbracket 0, T-1 \rrbracket. \quad (1g)$$

We make the following assumption throughout the paper.

¹Wait-and-see in the Stochastic Programming terminology

Assumption 1 (discrete white noise). *Assume that $N\xi_{t \in [1, T]}$ is a sequence of independent variable with finite support.*

Independence is of paramount importance to obtain Dynamic Programming equation, while finiteness of the support is required both to be able to compute exactly the expectation and for theoretical convergence reasons.

1.2 Stochastic Dual Dynamic Programming and its weaknesses

Thanks to white noise Assumption 1, we can solve Problem (1) by the Dynamic Programming approach (see the two reference books Bellman [1957] and Bertsekas [2005] for further details). This approach leads to the so-called Bellman's value functions V_t , such that $V_t(x)$ is the optimal value of the problem when starting at time t with state $\mathbf{X}_t = x$. These functions are obtained by solving the following recursive Bellman equation:

$$\begin{aligned} V_T(x) &= K(x) , \\ V_t(x) &= \mathbb{E} \left[\inf_{u_{t+1}, x_{t+1}} a_t^\top x + b_{t+1}^\top u_{t+1} + V_{t+1}(x_{t+1}) \right] , \\ \text{s.t. } \quad x_{t+1} &= A_t x + B_{t+1} u_{t+1} + C_{t+1} \xi_{t+1} , \\ \underline{u}_{t+1} &\leq u_{t+1} \leq \bar{u}_{t+1} , \\ \underline{x}_{t+1} &\leq x_{t+1} \leq \bar{x}_{t+1} , \\ D_t x + E_{t+1} u_{t+1} + G_{t+1} \xi_{t+1} &\leq 0 . \end{aligned}$$

Under further assumptions, we can show that V_t is finite on X_t . In the discrete state case, we can solve the Bellman equation by exhaustive exploration of the state, yielding the exact solution of the problem. In the continuous linear-convex case, we have to rely on polyhedral approximations. The Stochastic Dual Dynamic Programming algorithm (SDDP) builds polyhedral approximations \underline{V}_t of the functions V_t by using a nested Benders decomposition (see Philpott and Guan [2008], Girardeau et al. [2014], Guigues [2016] and Guigues [2017] for the convergence of this approach). In particular the polyhedral value functions \underline{V}_t computed by SDDP are *outer approximations* of the functions V_t at each stage, that is, $\underline{V}_t \leq V_t$, so that the value $\underline{v}_0 = \underline{V}_0(x_0)$ is an exact (deterministic) lower bound on the optimal value $v_0 = V_0(x_0)$ of Problem (1).

The polyhedral value functions \underline{V}_t obtained from SDDP can also be used to derive an admissible strategy, whose associated expected cost \bar{v}_0 gives an upper bound of the optimization problem value. Unfortunately, computing the expectation is usually out of reach, and we need to rely on some approximate computation. The most common way to perform that task is based on the Monte Carlo approach: it consists in simulating the control strategy induced by the polyhedral value functions \underline{V}_t along a (large) number M of noise scenarios, and then computing the arithmetic mean \hat{v}_0^M of the scenarios cost and the associated empirical standard deviation $\hat{\sigma}_0^M$. The value \hat{v}_0^M is an approximate (statistical) upper bound of the optimal value of Problem (1). Moreover, it is easy to obtain a α -confidence interval $[\hat{v}_0^M - z_\alpha \hat{\sigma}_0^M, \hat{v}_0^M + z_\alpha \hat{\sigma}_0^M]$. Here $1 - \alpha \in [0, 1]$ is a chosen confidence level and $z_\alpha = \Phi^{-1}(1 - \alpha)$, Φ being the cumulative distribution function of the standard normal distribution.

The classical way to use this statistical upper bound in SDDP, as presented in Pereira and Pinto [1991], consists in testing at each iteration of the algorithm if the available exact lower bound \underline{v}_0 is greater than the α -confidence lower bound $\hat{v}_0^M - z_\alpha \hat{\sigma}_0^M$, and to stop the algorithm in that case. Such a stopping criterion raises at least two difficulties:

1. the Monte Carlo simulation increases the computational burden of SDDP,

2. the stopping test does not give any guarantee of convergence of the algorithm.

The first difficulty can be tackled by parallelizing the M simulations involved in the evaluation of the upper bound, and also by calculating the empirical mean \widehat{v}_0 over the last \bar{k} iterations of the algorithm, thus enlarging the sample size from M to $\bar{k}M$ without additional computation (see [Shapiro et al., 2012, §3.2]). The second weakness induced by this stopping criterion has been analyzed in Shapiro [2011]: the larger the standard deviation $\widehat{\sigma}_0$ and the confidence $(1 - \alpha)$ are, the sooner the algorithm will be stopped. The author proposes another criterion based on the difference between the α -confidence upper bound $\widehat{v}_0 + z_\alpha \widehat{\sigma}_0$ and the exact lower bound \underline{v}_0 up to a prescribed accuracy level ε . An interesting view on the class of stopping criteria in terms of statistical hypothesis tests has been given in Homem-de Mello et al. [2011]: the authors compare different hypothesis tests of optimality² and so they find the stopping criteria proposed by Pereira and Pinto [1991], Shapiro [2011], as well as another one which ensures an upper bound on the probability of incorrectly claiming convergence (type II error). Moreover, the simulation scenarios are obtained using Quasi-Monte Carlo or Latin Hypercube Sampling rather than raw Monte carlo, so that the accuracy of the upper bound is increased. Nevertheless, all these stopping criteria are based on a statistical evaluation and it is thus not possible to obtain a guaranteed convergence criterion.

A different approach consists in building polyhedral inner approximations \overline{V}_t of the Bellman functions V_t at each stage, that is, $\overline{V}_t \geq V_t$. A deterministic upper bound $\overline{V}_0(x_0)$ of the optimal value of Problem (1) thus becomes available, and it is then possible to perform a stopping test of SDDP on the (deterministic) gap $\overline{V}_0(x_0) - \underline{V}_0(x_0)$. Such a test, giving a guarantee on the algorithm convergence has been investigated in Philpott et al. [2013]. More precisely, starting from a polyhedral inner approximation \overline{V}_{t+1} at time $t + 1$, and choosing an arbitrary sequence of points $x_t^1, \dots, x_t^{J_t}$, the authors show how to compute a value q_t^j at each point x_t^j such that $q_t^j \geq V_t(x_t^j)$. The inner polyhedral approximation \overline{V}_t is then obtained from the pairs $\{(x_t^j, q_t^j)\}_{j \in \llbracket 1, J_t \rrbracket}$. A delicate issue when devising the inner approximations is the choice of the points x_t^j defining the polyhedral functions \overline{V}_t . The authors suggest to use points generated by some other algorithm, such as SDDP. Another approach involving inner and outer approximations of the Bellman functions is described in Baucke et al. [2017]. The main feature of this last reference is that the algorithm presented herein is fully deterministic.

1.3 Contents of the paper

In Sect. 2, we introduce the formalism of linear Bellman operators for a large class of optimization problem, we define its dual linear Bellman operator and we enlighten the relationships between them thanks to the Fenchel conjugate. We also present the SDDP algorithm in this abstract formalism. We apply in Sect. 3 the conjugacy results obtained in Sect. 2 to obtain a dual SDDP algorithm associated to the standard SDDP algorithm used for solving Problem (1). The main result of this section is that we eventually obtain an exact upper bound over the value of Problem (1). In Sect. 4, we show how to build an inner approximation of Bellman functions associated to Problem (1) thanks to the outer approximation functions computed by the dual SDDP algorithm, and we prove that the expected cost induced by the control strategy based on these inner approximation Bellman functions is lower than the exact upper bound obtained in Sect. 3. Ultimately, in Sect. 5, we illustrate all the presented methodology on an energy management problem inspired by Électricité de France, at the European scale. The results show, on the one hand that having at disposal an exact upper bound in SDDP allows to devise a more efficient stopping test for SDDP than the usual ones based on a Monte Carlo approach, and

²such as: $(H_0: \bar{v}_0 = \underline{v}_0)$ against $(H_1: \bar{v}_0 \neq \underline{v}_0)$

on the other hand that the strategy based on the inner approximation of the Bellman functions outperforms the usual strategy obtained using standard outer approximations.

1.4 Notations

- $\llbracket i, j \rrbracket$ denotes the set of integer between i and j .
- Ω denotes a finite set of cardinality $|\Omega|$ supporting a probability distribution \mathbb{P} : $\Omega = \{\omega_1, \dots, \omega_{|\Omega|}\}$ and $\mathbb{P} = (\pi_1, \dots, \pi_{|\Omega|})$.
- Random variables are denoted using bold uppercase letters (such as $\mathbf{Z} : \Omega \rightarrow \mathbb{Z}$), and their realizations are denoted using lowercase letters ($z \in \mathbb{Z}$).
- $\mathbf{X} : \Omega \rightarrow \mathbb{X}$ corresponds to the state, $\mathbf{U} : \Omega \rightarrow \mathbb{U}$ to the control, $\boldsymbol{\xi} : \Omega \rightarrow \Xi$ to the noise.
- $V_t : \mathbb{X}_t \rightarrow \mathbb{R}$ is the Bellman value function associated to Problem (1) at time t .
- $\mathcal{D}_t = [V_t]^*$ is the dual value function associated to Problem (1) at time t .
- \mathcal{B} is a generic linear Bellman operator, with associated solution operator \mathcal{S} , and dual operator denoted \mathcal{B}^\ddagger .
- \mathcal{T} is the Bellman operator associated to Problem (1), it's dual being denoted \mathcal{T}^\ddagger .
- Underlined notation (e.g. \underline{V}) corresponds to a lower approximation of a function (e.g V). Overlined notation (e.g. \overline{V}) denotes a upper approximation.

2 Linear Bellman operators

This self-contained section is devoted to the definition and properties of linear Bellman operators (LBOs). In §2.1 we present the abstract formalism of LBOs that allows to write Dynamic Programming equations in a compact manner. In §2.2 we show that the Fenchel transform of a LBO is also a LBO. In §2.3 we present an abstract version of the SDDP algorithm.

2.1 Linear Bellman operator

We first introduce the notion of *linear Bellman operator*, which is a particular class of Bellman operators (see Bertsekas [2005]) associated to stochastic optimal control problems where costs and constraints are linear.

We consider an abstract probability space $(\Omega, \mathcal{A}, \mathbb{P})$. Recall that Ω is a finite set (see Assumption 1) and assume that the σ -field \mathcal{A} is generated by all the elements of Ω . We denote by $F(\mathbb{R}^{n_x})$ the set of functions defined on \mathbb{R}^{n_x} and taking values in $[-\infty, +\infty]$, and by $\mathcal{L}^0(\mathbb{R}^{n_x})$ the space of \mathbb{R}^{n_x} -valued measurable functions defined on $(\Omega, \mathcal{A}, \mathbb{P})$.

Remark 2. Since we suppose that the set Ω is finite, every function defined on Ω is measurable and a property that holds almost surely is a property that holds for every $\omega \in \Omega$. In particular $\mathcal{L}^0(\mathbb{R}^n)$ is also a finite-dimensional space. \diamond

Definition 3. An operator $\mathcal{B} : F(\mathbb{R}^{n_x}) \rightarrow F(\mathbb{R}^{n_x})$ is said to be a linear Bellman operator (LBO) if it is defined as follows

$$\begin{aligned} \mathcal{B} : F(\mathbb{R}^{n_x}) &\rightarrow F(\mathbb{R}^{n_x}) \\ R &\mapsto \mathcal{B}(R) : x \mapsto \inf_{(\mathbf{U}, \mathbf{Y}) \in \mathcal{L}^0(\mathbb{R}^{n_x} \times \mathbb{R}^{n_u})} \mathbb{E} \left[\mathbf{C}^\top \mathbf{U} + R(\mathbf{Y}) \right], \quad (2a) \\ &\text{s.t. } Tx + \mathcal{W}_u(\mathbf{U}) + \mathcal{W}_y(\mathbf{Y}) \leq \mathbf{H}, \quad (2b) \end{aligned}$$

where $\mathcal{W}_u : \mathcal{L}^0(\mathbb{R}^{n_u}) \rightarrow \mathcal{L}^0(\mathbb{R}^{n_c})$ and $\mathcal{W}_y : \mathcal{L}^0(\mathbb{R}^{n_x}) \rightarrow \mathcal{L}^0(\mathbb{R}^{n_c})$ are two linear operators. Here, \mathbf{U} and \mathbf{Y} are two decision random variables respectively defined on \mathbb{R}^{n_u} and \mathbb{R}^{n_x} . The two random variables $\mathbf{C} : \Omega \rightarrow \mathbb{R}^{n_u}$ and $\mathbf{H} : \Omega \rightarrow \mathbb{R}^{n_c}$ are exogenous uncertainties in Problem (2) and we note $\boldsymbol{\xi} = (\mathbf{C}, \mathbf{H})$. Deterministic matrix $T \in \mathbb{R}^{n_c \times n_x}$ is given data.

We denote by $\mathcal{S}(R)$ the set valued mapping giving, for a given $x \in \mathbb{R}^{n_x}$, the set of solutions \mathbf{Y} of Problem (2):

$$\mathcal{S}(R) : \mathbb{R}^{n_x} \rightarrow \mathcal{L}^0(\mathbb{R}^{n_x}) \quad (3a)$$

$$x \mapsto \arg \min_{\mathbf{Y} \in \mathcal{L}^0(\mathbb{R}^{n_x})} \left(\inf_{\mathbf{U} \in \mathcal{L}^0(\mathbb{R}^{n_u})} \mathbb{E} \left[\mathbf{C}^\top \mathbf{U} + R(\mathbf{Y}) \right] \right), \quad (3b)$$

$$\text{s.t. } Tx + \mathcal{W}_u(\mathbf{U}) + \mathcal{W}_y(\mathbf{Y}) \leq \mathbf{H}. \quad (3c)$$

Let $\mathcal{G} : \mathbb{R}^{n_x} \rightrightarrows \mathcal{L}^0(\mathbb{R}^{n_u}) \times \mathcal{L}^0(\mathbb{R}^{n_x})$ be the set valued mapping defined by

$$\mathcal{G}(x) := \{(\mathbf{U}, \mathbf{Y}) \mid Tx + \mathcal{W}_u(\mathbf{U}) + \mathcal{W}_y(\mathbf{Y}) \leq \mathbf{H}\}. \quad (4a)$$

With domain $\text{dom}(\mathcal{G}) = \{x \in \mathbb{R}^{n_x} \mid \mathcal{G}(x) \neq \emptyset\}$. Further, we say that \mathcal{B} is compact if \mathcal{G} is compact-valued with non-empty compact domain.

Once the domain of \mathcal{G} has been introduced, we define the key notion of *relatively complete recourse*, introduced in Rockafellar and Wets [1976]. We use a definition of relatively complete recourse close to the one given in [Shapiro et al., 2009, §2.1.3].

Definition 4. Let $R \in F(\mathbb{R}^{n_x})$ and \mathcal{B} a LBO. We say that the pair (\mathcal{B}, R) satisfy a relatively complete recourse (RCR) assumption if

$$\forall x \in \text{dom}(\mathcal{G}), \exists (\mathbf{U}, \mathbf{Y}) \in \mathcal{G}(x) \text{ such that } \mathbf{Y} \in \text{dom}(R) \text{ } \mathbb{P}\text{-a.s.} \quad (5)$$

Remark 5. Note that the set valued function \mathcal{G} , and thus $\text{dom}(\mathcal{G})$, does not depend on the operator R . We have that $\text{dom}(\mathcal{B}(R)) \subset \text{dom}(\mathcal{G})$ but we may not have equality because the domain of R is also involved in $\text{dom}(\mathcal{B}(R))$.

However, if the pair (\mathcal{B}, R) satisfy the RCR assumption then $\text{dom}(\mathcal{G}) = \text{dom}(\mathcal{B}(R))$. Thus, if \mathcal{B} is compact with non-empty domain, and (\mathcal{B}, R) satisfy the RCR assumption, then $\mathcal{B}(R)$ is finite at some point x_0 . \diamond

Example 1. We give some classical examples of operators \mathcal{W}_u and \mathcal{W}_y involved in Definition 3 of \mathcal{B} . We stress out that \mathcal{W} is a linear operator over a space of random variables, and we describe the associated adjoint operator, that is, the linear operator \mathcal{W}^\dagger such that $\langle \mathbf{X}, \mathcal{W}(\mathbf{Y}) \rangle = \langle \mathcal{W}^\dagger(\mathbf{X}), \mathbf{Y} \rangle$, with $\langle \mathbf{X}, \mathbf{Y} \rangle = \mathbb{E}[\mathbf{X}^\top \mathbf{Y}]$.

- Linear point-wise operator:

$$\begin{aligned} \mathcal{W} : \mathcal{L}^0(\mathbb{R}^{n_x}) &\rightarrow \mathcal{L}^0(\mathbb{R}^{n_c}) \\ (\omega \mapsto \mathbf{Y}(\omega)) &\mapsto (\omega \mapsto A\mathbf{Y}(\omega)). \end{aligned}$$

Such an operator allows to encode an *almost sure constraint*, and $\mathcal{W}^\dagger(\mathbf{X}) = A^\top \mathbf{X}$.

- Linear expected operator:

$$\mathcal{W} : \begin{array}{ccc} \mathcal{L}^0(\mathbb{R}^{n_x}) & \rightarrow & \mathcal{L}^0(\mathbb{R}^{n_c}) \\ (\omega \mapsto \mathbf{Y}(\omega)) & \mapsto & (\omega \mapsto A \mathbb{E}(\mathbf{Y})) . \end{array}$$

Such an operator allows to encode a *constraint in expectation*, and $\mathcal{W}^\dagger(\mathbf{X}) = A^\top \mathbb{E}(\mathbf{X})$.

- Linear conditional operator: given a sub σ -field \mathcal{F} of \mathcal{A} ,

$$\mathcal{W} : \begin{array}{ccc} \mathcal{L}^0(\mathbb{R}^{n_x}) & \rightarrow & \mathcal{L}^0(\mathbb{R}^{n_c}) \\ (\omega \mapsto \mathbf{Y}(\omega)) & \mapsto & (\omega \mapsto A \mathbb{E}[\mathbf{Y}|\mathcal{F}](\omega)) , \end{array}$$

Such an operator allows to encode, for example, measurability constraints and $\mathcal{W}^\dagger(\mathbf{X}) = A^\top \mathbb{E}[\mathbf{X}|\mathcal{F}]$.

Of course, any mix between these three kinds of operator is possible. \triangle

We now turn to properties of LBOs and polyhedral functions. Recall that a polyhedral set of \mathbb{R}^{n_x} is the finite intersection of closed half spaces, and a polyhedral function is a function whose epigraph is a polyhedral set (see [Rockafellar, 1970, Section 19]). In particular a polyhedral function is convex l.s.c., but not necessarily proper³. A non-proper polyhedral function takes value $-\infty$ on a polyhedral set, and $+\infty$ elsewhere.

Proposition 6. *Let R be a function of $F(\mathbb{R}^{n_x})$ and let \mathcal{B} be a LBO. Then we have the following properties.*

1. *If R is convex, then $\mathcal{B}(R)$ is convex.*
2. *If R is polyhedral, then $\mathcal{B}(R)$ is polyhedral.*
3. *If $R \geq \tilde{R}$, then $\mathcal{B}(R) \geq \mathcal{B}(\tilde{R})$.*

Proof. The probability set Ω being finite, we denote by u (resp. y, c, h) the vectors concatenating all possible values of \mathbf{U} (resp. $\mathbf{Y}, \mathbf{C}, \mathbf{H}$) over the set Ω , that is $u = (u_1, \dots, \mathbf{U}_{|\Omega|})$. Then the extensive formulation of Constraint (2b) is

$$\tilde{T}x + \tilde{W}_u u + \tilde{W}_y y \leq h ,$$

where \tilde{T} , \tilde{W}_u and \tilde{W}_y are adequate matrices deduced from T , \mathcal{W}_u and \mathcal{W}_y . Problem (2) rewrites

$$\mathcal{B}(R)(x) = \inf_{u,y} J(R)(x, u, y) ,$$

with

$$J(R)(x, u, y) = \sum_{s=1}^{|\Omega|} \pi_s \left(c_s^\top u_s + R(y_s) \right) + \chi_{\{\tilde{T}x + \tilde{W}_u u + \tilde{W}_y y \leq h\}}(x, u, y) ,$$

where χ_A is the characteristic function of a set A :

$$\chi_A(x) = \begin{cases} 0 & \text{if } x \in A, \\ +\infty & \text{otherwise.} \end{cases} \quad (6)$$

1. *If R is convex, then $J(R)$ is jointly convex in (x, u, y) so that $\mathcal{B}(R)$ is a convex function.*

³An extended real valued function is proper if it never takes the value $-\infty$ and is not identically equal to $+\infty$.

2. If R is polyhedral, then $J(R)$ is polyhedral in (x, u, y) , and thus $\mathcal{B}(R)$ is a polyhedral function (see [Borwein and Lewis, 2010, Prop 5.1.8]).
3. From $R \geq \tilde{R}$, we deduce that $J(R) \geq J(\tilde{R})$, and thus $\mathcal{B}(R) \geq \mathcal{B}(\tilde{R})$.

The proof is complete. \square

Remark 7. Assume that function R is proper and polyhedral. Then, under relatively complete recourse (see Definition 4), $\mathcal{B}(R)$ is a proper polyhedral function. Furthermore, if $\mathcal{B}(R)(x)$ is finite, computing it also generates a supporting hyperplane of $\mathcal{B}(R)$, that is, a pair $(\lambda, \beta) \in \mathbb{R}^{n_x} \times \mathbb{R}$ such that

$$\begin{cases} \langle \lambda, \cdot \rangle + \beta \leq \mathcal{B}(R)(\cdot) \\ \langle \lambda, x \rangle + \beta = \mathcal{B}(R)(x) . \end{cases}$$

Such hyperplanes, or cuts, are of paramount importance for the SDDP algorithm. \diamond

Proposition 8. Let R be a proper polyhedral function of $F(\mathbb{R}^{n_x})$ and let \mathcal{B} be a LBO. Assume that (\mathcal{B}, R) satisfy the RCR assumption . If R is Lipschitz (for the L_1 -norm) with constant L_R , then $\mathcal{B}(R)$ is also Lipschitz on its domain (which is $\text{dom}(\mathcal{G})$) with constant $\phi(L_R) = (\|\mathbf{C}\|_\infty + L_R)\kappa_W\|T\|_\infty$, where κ_W is a constant associated to the linear operator $(\mathcal{W}_u, \mathcal{W}_y)$.

Proof. Consider x_1 (resp. x_2) an element in $\text{dom}(\mathcal{B}(R))$, and denote by Z_1 (resp. Z_2) the polyhedron of optimal solutions of Problem (2). These polyhedrons are non-empty thanks to the RCR assumption. Let $(\mathbf{U}_1, \mathbf{Y}_1) \in Z_1$ be fixed, from [Shapiro et al., 2009, Theorem 7.12], there exists a positive constant κ_W such that

$$\inf_{(\mathbf{U}_2, \mathbf{Y}_2) \in Z_2} \|(\mathbf{U}_1, \mathbf{Y}_1) - (\mathbf{U}_2, \mathbf{Y}_2)\|_1 \leq \kappa_W \|T(x_1 - x_2)\|_1 \leq \kappa_W \|T\|_\infty \|x_1 - x_2\|_1 .$$

Let $(\mathbf{U}_2^\sharp, \mathbf{Y}_2^\sharp) \in Z_2$ be an optimal solution of the above minimization problem. Then we have that

$$\begin{aligned} \mathcal{B}(R)(x_1) &= \mathbb{E} \left[\mathbf{C}^\top \mathbf{U}_1 + R(\mathbf{Y}_1) \right] \\ &\leq \mathbb{E} \left[\mathbf{C}^\top \mathbf{U}_2^\sharp + \mathbf{C}^\top (\mathbf{U}_1 - \mathbf{U}_2^\sharp) + R(\mathbf{Y}_2^\sharp) + L_R |\mathbf{Y}_2^\sharp - \mathbf{Y}_1| \right] \\ &\leq \mathcal{B}(R)(x_2) + \|\mathbf{C}\|_\infty \|\mathbf{U}_1 - \mathbf{U}_2^\sharp\|_1 + L_R \|\mathbf{Y}_1 - \mathbf{Y}_2^\sharp\|_1 \\ &\leq \mathcal{B}(R)(x_2) + (\|\mathbf{C}\|_\infty + L_R)\kappa_W \|T\|_\infty \|x_1 - x_2\|_1 . \end{aligned}$$

Exchanging x_1 and x_2 in the previous majoration leads to the reverse inequality. \square

2.2 Fenchel transform of a LBO

We now define the *dual LBO* \mathcal{B}^\dagger of a linear Bellman operator \mathcal{B} .

Definition 9. Let \mathcal{B} be a LBO (see Definition (3)). We denote by \mathcal{B}^\dagger the dual LBO of \mathcal{B} , defined, for a given function $Q \in F(\mathbb{R}^{n_x})$ and for any $\lambda \in \mathbb{R}^{n_x}$, by

$$\mathcal{B}^\dagger(Q)(\lambda) = \inf_{\boldsymbol{\mu} \in \mathcal{L}^0(\mathbb{R}^{n_x}), \boldsymbol{\nu} \in \mathcal{L}^0(\mathbb{R}^{n_c})} \mathbb{E} \left[-\boldsymbol{\mu}^\top \mathbf{H} + Q(\boldsymbol{\nu}) \right] \quad (7a)$$

$$\text{s.t. } T^\top \mathbb{E}[\boldsymbol{\mu}] + \lambda = 0 \quad (7b)$$

$$\mathcal{W}_u^\dagger(\boldsymbol{\mu}) = \mathbf{C} \quad (7c)$$

$$\mathcal{W}_y^\dagger(\boldsymbol{\mu}) = \boldsymbol{\nu} \quad (7d)$$

$$\boldsymbol{\mu} \leq 0 , \quad (7e)$$

where \mathcal{W}_u^\dagger (resp. \mathcal{W}_y^\dagger) is the adjoint operator of \mathcal{W}_u (resp. \mathcal{W}_y). We define the dual constraint set valued mapping

$$\mathcal{G}^\ddagger(\lambda) = \{(\boldsymbol{\mu}, \boldsymbol{\nu}) \in \mathbb{R}^{n_x+n_c} \mid T^\top \mathbb{E}[\boldsymbol{\mu}] + \lambda = 0, \mathcal{W}_u^\dagger(\boldsymbol{\mu}) = \mathbf{C}, \mathcal{W}_y^\dagger(\boldsymbol{\mu}) = \boldsymbol{\nu}, \boldsymbol{\mu} \leq 0\}. \quad (8)$$

Note that straightforward computations show that $(\mathcal{B}^\ddagger)^\ddagger = \mathcal{B}$.

The next proposition exhibit the relationship between \mathcal{B}^\ddagger and the Fenchel transform of \mathcal{B} .

Proposition 10. *Let R be a proper polyhedral function, \mathcal{B} be a compact LBO (see Definition 3), such that the pair (\mathcal{B}, R) satisfy the RCR assumption. Then $\mathcal{B}(R)$ is a proper polyhedral function and we have that*

$$[\mathcal{B}(R)]^*(\lambda) = \mathcal{B}^\ddagger([\mathcal{B}^\ddagger]^*(\lambda)) \quad . \quad (9)$$

Proof. First note that as \mathcal{B} is compact, \mathcal{G} has non-empty compact domain, thus by (RCR) $\mathcal{B}(R)$ is finite at some point.

During the proof we denote $\langle \mathbf{X}, \mathbf{Y} \rangle = \mathbb{E}(\mathbf{X}^\top \mathbf{Y})$, $\mathcal{R}(\mathbf{Y}) = \mathbb{E}(R(\mathbf{Y}))$, and

$$\begin{aligned} \mathcal{K}(x, \mathbf{Y}) &= \min_{\mathbf{U}} \langle \mathbf{C}, \mathbf{U} \rangle \\ \text{s.t. } & T x + \mathcal{W}_u(\mathbf{U}) + \mathcal{W}_y(\mathbf{Y}) \leq \mathbf{H} . \end{aligned}$$

By definition, we have

$$\mathcal{B}(R)(x) = \inf_{\mathbf{Y}} \mathcal{K}(x, \mathbf{Y}) + \mathcal{R}(\mathbf{Y}).$$

Thus, for any $x^* \in \mathbb{R}^{n_x}$, we have

$$\begin{aligned} [\mathcal{B}(R)]^*(x^*) &= \sup_{x \in \mathbb{R}^{n_x}} \left\{ x^\top x^* - \inf_{\mathbf{Y}} \{ \mathcal{K}(x, \mathbf{Y}) + \mathcal{R}(\mathbf{Y}) \} \right\} \\ &= - \inf_{\mathbf{Y}} \left\{ \mathcal{R}(\mathbf{Y}) - \underbrace{\sup_{x \in \mathbb{R}^{n_x}} x^\top x^* - \mathcal{K}(x, \mathbf{Y})}_{:= \Phi(\mathbf{Y})} \right\} \end{aligned}$$

As R is polyhedral proper, so is \mathcal{R} . By construction \mathcal{K} is polyhedral. By compacity of \mathcal{B} , the minimization in \mathbf{U} is over a compact, thus $\mathcal{K} > -\infty$. Further, as $\text{dom}(\mathcal{B})(R) = \text{dom}(\mathcal{G}) \neq \emptyset$, \mathcal{K} is proper. Note that for $x \notin \text{dom}(\mathcal{G})$ we have $\mathcal{K}(x, \mathbf{Y}) = +\infty$, thus $\Phi(\mathbf{Y}) = \sup_{x \in \text{dom}(\mathcal{G})} x^\top x^* - \mathcal{K}(x, \mathbf{Y})$. Consequently, $-\Phi$ is polyhedral proper as $\text{dom}(\mathcal{G})$ is a non empty compact set. Finally, the RCR assumption ensures that $\text{dom}(-\Phi) \cap \text{dom}(\mathcal{R}) \neq \emptyset$. Now, using Fenchel-Duality (in the polyhedral case) we have that

$$[\mathcal{B}(R)]^*(x^*) = - \sup_{\mathbf{Y}^*} \Phi_*(\mathbf{Y}^*) - \mathcal{R}^*(\mathbf{Y}^*) = \inf_{\mathbf{Y}^*} \mathcal{R}^*(\mathbf{Y}^*) - \Phi_*(\mathbf{Y}^*)$$

where $\mathcal{R}^*(\mathbf{Y}^*) = \mathbb{E}(R^*(\mathbf{Y}^*))$ and

$$\begin{aligned} \Phi_*(\mathbf{Y}^*) &= \inf_{\mathbf{Y}} \langle \mathbf{Y}^*, \mathbf{Y} \rangle - \Phi(\mathbf{Y}) \\ &= \inf_{\mathbf{Y}} \langle \mathbf{Y}^*, \mathbf{Y} \rangle - \sup_x \{ x^\top x^* - \mathcal{K}(x, \mathbf{Y}) \} \\ &= \inf_{x, \mathbf{Y}} \langle \mathbf{Y}^*, \mathbf{Y} \rangle - x^\top x^* + \mathcal{K}(x, \mathbf{Y}) \\ &= \inf_{x, \mathbf{Y}, \mathbf{U}} \langle \mathbf{Y}^*, \mathbf{Y} \rangle - x^\top x^* + \langle \mathbf{C}, \mathbf{U} \rangle \\ \text{s.t. } & T x + \mathcal{W}_u(\mathbf{U}) + \mathcal{W}_y(\mathbf{Y}) \leq \mathbf{H} \end{aligned}$$

As $\text{dom}(\mathcal{G}) \neq \emptyset$, there exists a primal feasible solution to the above linear program, and by duality we have

$$\begin{aligned}
\Phi_*(\mathbf{Y}^*) &= \sup_{\boldsymbol{\lambda} \geq 0} -\langle \boldsymbol{\lambda}, \mathbf{H} \rangle + \inf_x \{ -x^\top x^* + \langle T^\top \boldsymbol{\lambda}, x \rangle \} + \inf_{\mathbf{Y}} \{ \langle \mathbf{Y}^*, \mathbf{Y} \rangle + \langle \mathcal{W}_y^\dagger(\boldsymbol{\lambda}), \mathbf{Y} \rangle \} \\
&\quad + \inf_{\mathbf{U}} \{ \langle \mathbf{C}, \mathbf{U} \rangle + \langle \mathcal{W}_u^\dagger(\boldsymbol{\lambda}), \mathbf{U} \rangle \} \\
&= \sup_{\boldsymbol{\lambda} \geq 0} -\langle \boldsymbol{\lambda}, \mathbf{H} \rangle \\
&\quad \text{s.t. } \mathcal{W}_y^\dagger(\boldsymbol{\lambda}) = -\mathbf{Y}^* \\
&\quad \quad \mathcal{W}_u^\dagger(\boldsymbol{\lambda}) = -\mathbf{C} \\
&\quad \quad T^\top \mathbb{E}(\boldsymbol{\lambda}) = x^*
\end{aligned}$$

Finally,

$$\begin{aligned}
[\mathcal{B}(R)]^*(x^*) &= \inf_{\mathbf{Y}^*, \boldsymbol{\lambda} \geq 0} \mathbb{E} \left(\mathbf{H}^\top \boldsymbol{\lambda} + R^*(\mathbf{Y}^*) \right) \\
&\quad \text{s.t. } T^\top \mathbb{E}(\boldsymbol{\lambda}) = x^* \\
&\quad \quad \mathcal{W}_y^\dagger(\boldsymbol{\lambda}) = -\mathbf{Y}^* \\
&\quad \quad \mathcal{W}_u^\dagger(\boldsymbol{\lambda}) = -\mathbf{C},
\end{aligned}$$

which ends the proof with $\lambda \rightarrow -\mu$, $x^* \rightarrow \lambda$ and $\mathbf{Y}^* \rightarrow \boldsymbol{\nu}$. \square

2.3 An abstract SDDP algorithm

We consider a sequence of functions $\{R_t\}_{t \in \llbracket 0, T \rrbracket}$ that follows

$$\begin{cases} R_T = K \\ R_t = \mathcal{B}_t(R_{t+1}) \quad \forall t \in \llbracket 0, T-1 \rrbracket, \end{cases} \quad (10)$$

where K is a proper polyhedral function and \mathcal{B}_t a LBO given by

$$\begin{aligned}
\mathcal{B}_t(R)(x) &= \inf_{\mathbf{U}, \mathbf{Y}} \mathbb{E} \left[\mathbf{C}_t^\top \mathbf{U} + R(\mathbf{Y}) \right] \\
&\quad \text{s.t. } T_t x + \mathcal{W}_t^u(\mathbf{U}) + \mathcal{W}_t^y(\mathbf{Y}) \leq \mathbf{H}_t,
\end{aligned}$$

with associated solution operator $\{\mathcal{S}_t\}_{t \in \llbracket 0, T-1 \rrbracket}$:

$$\begin{aligned}
\mathcal{S}_t(R)(x) &= \arg \min_{\mathbf{Y}} \inf_{\mathbf{U}} \mathbb{E} \left[\mathbf{C}_t^\top \mathbf{U} + R(\mathbf{Y}) \right] \\
&\quad \text{s.t. } T_t x + \mathcal{W}_t^u(\mathbf{U}) + \mathcal{W}_t^y(\mathbf{Y}) \leq \mathbf{H}_t.
\end{aligned}$$

Definition 11. We say that a sequence of LBOs $(\mathcal{B}_t)_{t \in \llbracket 0, T-1 \rrbracket}$ is compatible (with final cost K) if, at any time $t \in \llbracket 0, T-1 \rrbracket$ every future state \mathbf{Y}_t admissible for \mathcal{B}_t is (almost surely) an admissible current state for \mathcal{B}_{t+1} . More precisely, for $t \in \llbracket 0, T-1 \rrbracket$, for all $x \in \text{dom}(\mathcal{G}_t)$, for all $(\mathbf{U}_t, \mathbf{Y}_t) \in \mathcal{G}_t(x)$ we have almost surely $\mathbf{Y}_t \in \text{dom}(\mathcal{G}_{t+1})$, where, by convention we set $\text{dom}(\mathcal{G}_T) := \text{dom}(K)$.

SDDP is an algorithm that iteratively construct finite lower polyhedral approximation of R_t . Starting from an initial point $x_0 \in \mathbb{R}^{n_x}$, the algorithm determines in a forward pass a sequence of state $(x_t^k)_{t \in \llbracket 0, T \rrbracket}$ at which the approximation will be refined in the backward pass. Algorithm 1 details more precisely the pseudo-code of the abstract SDDP algorithm.

```

Data: Initial point  $x_0$ 
Set  $\underline{R}_t^{(0)} \equiv -\infty$ 
for  $k = 0, 1, \dots$  do
    // Forward Pass : compute a set of trial points  $\{x_t^k\}_{t \in \llbracket 0, T \rrbracket}$ 
    Randomly select  $\omega^k \in \Omega$ ;
    Set  $x_0^k = x_0$ ;
    for  $t : 0 \rightarrow T - 1$  do
        select  $\mathbf{X}_{t+1}^k \in \mathcal{S}_t(\underline{R}_{t+1}^k)(x_t^k)$ ; // see Definition 3
        set  $x_{t+1}^k = \mathbf{X}_{t+1}^k(\omega^k)$ ;
    end
    // Backard Pass : refine the lower-approximations at the trial points
    Set  $\underline{R}_T^{k+1} = K$ ;
    for  $t : T - 1 \rightarrow 0$  do
         $\beta_t^{k+1} = \mathcal{B}_t(\underline{R}_{t+1}^{k+1})(x_t^k)$ ; // cut coefficients (see Remark 7)
         $\lambda_t^{k+1} \in \partial \mathcal{B}_t(\underline{R}_{t+1}^{k+1})(x_t^k)$ ;
         $\beta_t^{k+1} := \theta_t^{k+1} - \langle \lambda_t^{k+1}, \bar{x}_t^k \rangle$ ;
        set  $\mathcal{C}_t^{k+1} : x \mapsto \langle \lambda_t^{k+1}, x \rangle + \beta_t^{k+1}$ ; // new cut
         $\underline{R}_t^{k+1} := \max \{ \underline{R}_t^k, \mathcal{C}_t^{k+1} \}$ ; // update lower approximation
    end
    If some stopping test is satisfied STOP ;
end

```

Algorithm 1: Abstract SDDP algorithm

Lemma 12. *Assume that $R_0(x_0)$ is finite and that $(\mathcal{B}_t)_{t \in \llbracket 0, T-1 \rrbracket}$ is a compatible sequence of LBO. Then, the SDDP algorithm 1 is well defined and there exists a sequence $(L_t)_{t \in \llbracket 0, T-1 \rrbracket}$ such that R_t is L_t -Lipschitz on its domain, and $\|\lambda_t^{(k)}\|_\infty \leq L_t$.*

Proof. We prove by induction that $x_t^{(k)}$ is well defined during the forward passes of SDDP. Let $t = 0$. By hypothesis, $x_0^k \in \text{dom}(\mathcal{G}_0)$. So $x_1^k = \mathbf{X}_1^{(k)}(\omega^k)$ exists as solution of a finite valued LP. Let $t \geq 1$. By induction hypothesis, we suppose that x_t^k is well defined and belongs to $\text{dom}(\mathcal{G}_t)$. We set $x_{t+1}^k = \mathbf{X}_{t+1}^{(k)}(\omega^k)$, which is well defined as solution of a finite value LP. By assumption the sequence $\{\underline{R}_t\}_{t \in \llbracket 0, T \rrbracket}$ is compatible, hence $x_{t+1}^k \in \text{dom}(\mathcal{G}_{t+1})$, thus proving the hypothesis at time $t + 1$.

We now prove by backward induction that $\lambda_t^{(k)}$ is well defined during the backward passes of SDDP, and that it exists L_t such that $\|\lambda_t^{(k)}\|_\infty \leq L_t$. As K is given and L_T Lipschitz-continuous, the property is direct for $t = T$. Let $t \leq T - 1$. We suppose that induction hypothesis holds for $t + 1$. Then, by Proposition 8, we know that $\mathcal{B}_t(\underline{R}_{t+1}^{k+1})$ is L_t -Lipschitz. We set $\lambda_t^{k+1} \in \partial \mathcal{B}_t(\underline{R}_{t+1}^{k+1})(x_t^k)$, which is well defined as subgradient of a polyhedral function. As $\mathcal{B}_t(\underline{R}_{t+1}^{k+1})$ is L_t -Lipschitz on its domain, we are able to choose λ_t^{k+1} such that $\|\lambda_t^{(k+1)}\|_\infty \leq L_t$ \square

Proposition 13. *Assume that $R_0(x_0)$ is finite and that $(\mathcal{B}_t)_{t \in \llbracket 0, T-1 \rrbracket}$ is a compatible sequence of LBO. Further assume that, for all $t \in \llbracket 0, T \rrbracket$ there exists compact sets X_t such that, for all k ,*

$x_t^k \in X_t$. In particular this is the case if \mathcal{B}_t is compact for all t .

Then, the SDDP algorithm generates a non-decreasing sequence $(\underline{R}_t^{(k)})_{k \in \mathbb{N}}$ of lower approximation of R_t , and $\lim_k \underline{R}_0^{(k)}(x_0) = R_0(x_0)$.

Proof. Lemma 12 gives the boundedness of $\lambda_t^{(k)}$, from which we can easily adapt the proof of Girardeau et al. [2014]. \square

This algorithm is abstract in the sense that it only requires a sequence of LBOs. In the following section we show how it can be applied to approximate the Bellman value functions $\{V_t\}_{t \in \llbracket 0, T \rrbracket}$, or to approximate the Fenchel transform of these functions.

3 Primal and dual SDDP

In this section we recall the usual SDDP algorithm applied to Problem (1). Next, leveraging the results of Sect. 2, we introduce a dual SDDP algorithm, which is the abstract SDDP algorithm applied to the dual value functions. This eventually gives an exact upper-bound over the value of Problem (1).

In this section, we denote by V_t the primal value functions, and by $\mathcal{D}_t = [V_t]^*$ the dual value functions. As noted in Sect. 1.4, outer (resp. inner) approximations are denoted by \underline{V} (resp. \underline{D}).

3.1 Primal SDDP

3.1.1 Primal Dynamic Programming equations

We consider Problem (1) under the discrete white noise Assumption 1.

Denote by $U_{t+1}(x, \xi)$ the set of admissible controls at time t with given state x and uncertainty ξ

$$U_{t+1}(x, \xi) = \{ \underline{u}_{t+1} \leq u \leq \bar{u}_{t+1} \mid D_t x + E_{t+1} u + G_{t+1} \xi \leq 0 \}, \quad (11)$$

and by $X_t \subset \mathbb{R}^{n_x}$ be the set of admissible state at time t , more precisely

$$X_t = \{ x_t \leq x \leq \bar{x}_t \mid U(x, \xi) \neq \emptyset \quad \mathbb{P} - \text{a.s.} \}, \quad (12)$$

with $X_T = \text{dom}(K)$.

Assumption 14. For all $t \in \llbracket 0, T - 1 \rrbracket$, we make the following assumptions.

1. For all $t \in \llbracket 0, T - 1 \rrbracket$, for all $x \in X_t$ and for all $\xi \in \text{supp}(\xi_{t+1})$, the set of admissible controls $U_{t+1}(x, \xi)$ is a polytope (non empty since $x \in X_t$) of \mathbb{R}^{n_u} .
2. For all $t \in \llbracket 0, T \rrbracket$, the set of admissible states X_t is a polytope of \mathbb{R}^n .
3. We are in a relatively complete recourse framework : for all $t \in \llbracket 0, T - 1 \rrbracket$,

$$\forall x \in X_t, \quad \forall \xi \in \text{supp}(\xi_t), \quad \exists u \in U_t(x, \xi), \quad A_t x + B_{t+1} u + C_{t+1} \xi \in X_{t+1}. \quad (13)$$

4. Problem (1) is finite valued.

Under this assumptions we can solve Problem (1) by Dynamic Programming, computing backward the value functions $\{V_t\}_{t \in \llbracket 0, T \rrbracket}$ given by

$$\begin{cases} V_T = K, \\ V_t = \mathcal{T}_t(V_{t+1}), \end{cases} \quad (14)$$

where the primal Bellman operator $\mathcal{T}_t : F(\mathbb{R}^n) \rightarrow F(\mathbb{R}^n)$ is defined as follows:

$$\mathcal{T}_t(R) : x \mapsto \inf_{\mathbf{U}_{t+1}, \mathbf{X}_{t+1}} \mathbb{E} \left[a_t^\top x + b_{t+1}^\top \mathbf{U}_{t+1} + R(\mathbf{X}_{t+1}) \right], \quad (15a)$$

$$\text{s.t.} \quad \mathbf{X}_{t+1} = A_t x + B_{t+1} \mathbf{U}_{t+1} + C_{t+1} \boldsymbol{\xi}_{t+1}, \quad (15b)$$

$$D_t x + E_{t+1} \mathbf{U}_{t+1} + G_{t+1} \boldsymbol{\xi}_{t+1} \leq 0, \quad (15c)$$

$$x \in X_t, \quad \underline{u}_{t+1} \leq \mathbf{U}_{t+1} \leq \bar{u}_{t+1}, \quad \mathbf{X}_{t+1} \in X_{t+1}. \quad (15d)$$

For notational simplicity we assume from now on that constraint (15d) is included in the previous ones.

Lemma 15. *Under Assumption 14, we have that $\text{dom}(V_t) = \text{dom}(G_t) = X_t$. Further, $(\mathcal{T}_t)_{t \in [0, T-1]}$ is a compatible sequence of compact LBOs, and*

$$\mathcal{T}_t(R) : x \mapsto \mathbb{E}(\widehat{\mathcal{T}}_t(R)(x, \boldsymbol{\xi}_t)) \quad (16)$$

where

$$\widehat{\mathcal{T}}_t(R) : (x, \xi) \mapsto \inf_{u_{t+1}, x_{t+1}} a_t^\top x + b_{t+1}^\top u_{t+1} + R(x_{t+1}), \quad (17a)$$

$$\text{s.t.} \quad x_{t+1} = A_t x + B_{t+1} u_{t+1} + C_{t+1} \xi, \quad (17b)$$

$$D_t x + E_{t+1} u_{t+1} + G_{t+1} \xi \leq 0. \quad (17c)$$

Proof. We can cast Bellman operator (15) into the definition of abstract Bellman operators (2), by noting that

$$\begin{aligned} \mathcal{T}_t(R) : x \mapsto a_t^\top x + \inf_{\mathbf{U}_{t+1}, \mathbf{X}_{t+1}} \mathbb{E} \left[b_{t+1}^\top \mathbf{U}_{t+1} + R(\mathbf{X}_{t+1}) \right], \\ \text{s.t.} \quad \begin{pmatrix} A_t \\ -A_t \\ D_t \end{pmatrix} x + \begin{pmatrix} B_{t+1} \\ -B_{t+1} \\ E_{t+1} \end{pmatrix} \mathbf{U}_{t+1} + \begin{pmatrix} I \\ -I \\ 0 \end{pmatrix} \mathbf{X}_{t+1} \leq \begin{pmatrix} -C_{t+1} \\ C_{t+1} \\ -G_{t+1} \end{pmatrix} \boldsymbol{\xi}_{t+1}. \end{aligned}$$

Then, the reformulation as Equations (16) and (17) is well known using the measurability assumption $\mathbf{U}_{t+1} \preceq \mathcal{F}_{t+1}$. Compactness comes from constraints (1d) and (1e), and compatibility from the relatively complete recourse assumption. \square

To recover the optimal state and control trajectories from Bellman functions, we introduce the set valued mappings:

$$\widehat{\mathcal{S}}_t(R) : (x, \xi) \mapsto \arg \min_{x_{t+1}} \min_u a_t^\top x + b_{t+1}^\top u + R(x_{t+1}), \quad (19a)$$

$$\text{s.t.} \quad x_{t+1} = A_t x + B_{t+1} u + C_{t+1} \xi \quad (19b)$$

$$D_t x + E_{t+1} u + G_{t+1} \xi \leq 0. \quad (19c)$$

3.1.2 Primal SDDP

We now apply the abstract SDDP algorithm presented in §2.3 to the primal Bellman operator given by Equations (15). We recall that the final cost K is polyhedral. We denote, for all

$t \in \llbracket 1, T \rrbracket$, and $\pi_t^\xi := \mathbb{P}(\boldsymbol{\xi}_t = \xi)$ for all $\xi \in \text{supp}(\boldsymbol{\xi})$.

Data: Initial point x_0 , initial lower bounds \underline{V}_t^0 on V_t
for $k \in \mathbb{N}$ **do**
 Draw a noise scenario $\{\boldsymbol{\xi}_t^k\}_{t \in \llbracket 0, T \rrbracket}$;
 // Forward Pass : compute a set of trial points $\{x_t^k\}_{t \in \llbracket 0, T \rrbracket}$
 Set $x_0^k = x_0$;
 for $t : 0 \rightarrow T - 1$ **do**
 | select $x_{t+1}^k \in \widehat{\mathcal{S}}_t(\underline{V}_{t+1}^k)(x_t^k, \boldsymbol{\xi}_{t+1}^k)$
 end
 // Backward Pass : refine the lower-approximations at the trial points
 Set $\underline{V}_T^{k+1} = K$;
 for $t : T - 1 \rightarrow 0$ **do**
 for $\xi \in \text{supp}(\boldsymbol{\xi}_{t+1})$ **do**
 | solve the linear program $\widehat{\mathcal{T}}_t(\underline{V}_{t+1}^{k+1})(x_t^k, \xi)$;
 | yielding $\underline{\theta}_t^{\xi, k+1} := \widehat{\mathcal{T}}_t(\underline{V}_{t+1}^{k+1})(x_t^k, \xi)$ and $\underline{\lambda}_t^{\xi, k+1} \in \partial \widehat{\mathcal{T}}_t(\underline{V}_{t+1}^{k+1})(x_t^k, \xi)$;
 end
 $\underline{\lambda}_t^{k+1} := \sum_{\xi \in \text{supp}(\boldsymbol{\xi}_{t+1})} \pi_{t+1}^\xi \underline{\lambda}_t^{\xi, k+1}$; // taking expectation
 $\underline{\beta}_t^{k+1} := \sum_{\xi \in \text{supp}(\boldsymbol{\xi}_{t+1})} \pi_{t+1}^\xi (\underline{\theta}_t^{\xi, k+1} - \langle \underline{\lambda}_t^{\xi, k+1}, x_t^k \rangle)$;
 $\underline{V}_t^{k+1} := \max \left\{ \underline{V}_t^k(\cdot), \langle \underline{\lambda}_t^k, \cdot \rangle + \underline{\beta}_t^{k+1} \right\}$; // update lower approximation
 end
 If some stopping test is satisfied STOP ;
end

Algorithm 2: Primal SDDP algorithm

Remark 16. Note that, the primal Bellman operator (15) is a specialized version of the abstract Bellman operator used in Equation (10), which only involves pointwise operator in the constraints. Hence, in the forward pass we just have to compute $\widehat{\mathcal{T}}_t(\underline{V}_{t+1}^k)(x_t^k, \boldsymbol{\xi}_{t+1}^k)$ and do not need to compute $\mathcal{T}_t(\underline{V}_{t+1}^k)(x_t^k)$ which would be a larger linear problem. Similarly, in the backward pass at time t we solve $|\text{supp}(\boldsymbol{\xi}_{t+1})|$ linear problem of the form $\widehat{\mathcal{T}}_t(\underline{V}_{t+1}^{k+1})(x_t^k, \boldsymbol{\xi}_{t+1}^s)$ instead of the larger $\mathcal{T}_t(\underline{V}_{t+1}^{k+1})(x_t^k)$ and then perform an expectation. We will show in the sequel that this is no more the case in the dual SDDP algorithm. \diamond

Proposition 17. *Under Assumptions 1, and 14, the primal SDDP algorithm yields a converging lower bound for the value of Problem (1). Further, the strategy induced by $\underline{V}_t^{(k)}$ is converging toward an optimal strategy.*

Proof. By Dynamic Programming, we know that the value functions V_t follow the recursion (14). By Lemma 15 we have the compatibility of the sequence of LBOs $(\mathcal{T}_t)_{t \in \llbracket 0, T-1 \rrbracket}$. The compacity of X_t ensures that the sequence $(x_t^k)_{k \in \mathbb{N}}$ remains in a compact. Hence, we can apply Proposition 13.

Proof of the convergence of the strategy obtained can be found in Girardeau et al. [2014]. \square

3.2 Dual SDDP

We present here a dual SDDP algorithm, which leverages the results of §2.2. We show that the Fenchel conjugate of the primal value functions V_t follows a recursive equation on which we apply the abstract SDDP algorithm of §2.3.

3.2.1 Dual Dynamic Programming equations

By Definition 9, the dual LBO of \mathcal{T}_t , is given by

$$\mathcal{T}_t^\ddagger(Q) : \lambda_t \mapsto \inf_{\mathbf{\Lambda}_{t+1}, \mathbf{N}_{t+1} \geq 0} \mathbb{E} \left[- (C_{t+1}^\top \mathbf{\Lambda}_{t+1} + G_{t+1}^\top \mathbf{N}_{t+1})^\top \boldsymbol{\xi}_{t+1} + Q(\mathbf{\Lambda}_{t+1}) \right] \quad (20a)$$

$$\text{s.t. } a_t + A_t^\top \mathbb{E}[\mathbf{\Lambda}_{t+1}] + D_t^\top \mathbb{E}[\mathbf{N}_{t+1}] - \lambda_t = 0 \quad (20b)$$

$$b_{t+1} + B_{t+1}^\top \mathbf{\Lambda}_{t+1} + E_{t+1}^\top \mathbf{N}_{t+1} = 0, \quad (20c)$$

where $\mathbf{\Lambda}_{t+1} : \Omega \rightarrow \mathbb{R}^{n_x}$ and $\mathbf{N}_{t+1} : \Omega \rightarrow \mathbb{R}^{n_c}$ are two $\boldsymbol{\xi}_{t+1}$ -measurable random variables.

In Equation (20), the function Q is a cost-to-go at time $t+1$ for the dual optimization problem, λ_t is a state variable, and $(\mathbf{\Lambda}_{t+1}, \mathbf{N}_{t+1})$ are control variables. Equations (20b) and (20c) define the admissible control set of the problem.

Theorem 18. *Denote, for any $t \in \llbracket 0, T \rrbracket$, $\mathcal{D}_t := V_t^*$, where V_t is the Bellman value function following Dynamic Programming equations (14). Let, for all $t \in \llbracket 0, T \rrbracket$, $L_t > 0$ be such that V_t is L_t -Lipschitz (for the L_1 -norm) on its domain. Then the sequence of dual value functions $\{\mathcal{D}_t\}_{t \in \llbracket 0, T \rrbracket}$ satisfy the following backward recursion:*

$$\mathcal{D}_T = K^*, \quad (21a)$$

$$\mathcal{D}_t = \mathcal{T}_{t, L_{t+1}}^\ddagger(\mathcal{D}_{t+1}) \quad \forall t \in \llbracket 0, T-1 \rrbracket, \quad (21b)$$

where $\mathcal{T}_{t, L_{t+1}}^\ddagger$ is defined by Equation (20), with the additional constraint $\|\mathbf{\Lambda}_{t+1}(\omega)\|_\infty \leq L_{t+1}$.

Remark 19. Lemma 12 shows that a sequence of Lipschitz constants $N_{L_{t \in \llbracket 0, T \rrbracket}}$ exists. But in some cases we can directly derive Lipschitz constant on the value functions, and plug it into Equation (21). \diamond

Proof. First recall that by assumption \mathcal{T}_t is a compact LBO. By Dynamic Programming (see Equation (14)) we have $V_T = K$, and for all $t \in \llbracket 0, T-1 \rrbracket$, $V_t = \mathcal{T}_t(V_{t+1})$. Let $t \in \llbracket 0, T-1 \rrbracket$. Consider the L_t -Lipschitz regularization of V_t (or Pasch-Hausdorff envelope, see Bauschke et al. [2017]) defined by $V_t^{L_t} := V_t \square (L_t \|\cdot\|_1)$, where $f \square g : x \mapsto \inf_y f(y) + g(x-y)$. As V_t is L_t -Lipschitz continuous on its domain, $V_t^{L_t}$ coincides with V_t on its domain, and is L_t -Lipschitz continuous everywhere (see [Bauschke et al., 2017, Corollary 12.19]). The compatibility assumption implies that only the restriction of V_{t+1} to X_{t+1} matters for all $t \in \llbracket 0, T-1 \rrbracket$, thus $V_t = \mathcal{T}_t(V_{t+1}^{L_{t+1}})$. Then Proposition 10 gives

$$[V_t]^* = \mathcal{T}_t^\ddagger \left(\left[V_{t+1}^{L_{t+1}} \right]^* \right).$$

As V_{t+1} and $L_{t+1} \|\cdot\|_1$ takes values in $(-\infty, +\infty]$, we have ([Bauschke et al., 2017, Corollary 13.24])

$$\left[V_{t+1}^{L_{t+1}} \right]^* = [V_{t+1}]^* + \chi_{B_\infty(0, L_{t+1})},$$

where $B_\infty(0, L_{t+1})$ is the L_∞ -ball of radius L_{t+1} centered in 0. Thus we have

$$\mathcal{D}_t = \mathcal{T}_t^\ddagger \left(\mathcal{D}_{t+1} + \chi_{B_\infty(0, L_{t+1})} \right),$$

which proves (21). \square

3.2.2 Dual SDDP

From now on we assume that $\{\mathcal{T}_{t,L_{t+1}}^\ddagger\}_{t \in \llbracket 0, T-1 \rrbracket}$ is compatible. This is ensured for example if all A_t in Problem (1) are square invertible matrix (see Appendix A.3).

The dual value functions $\{\mathcal{D}_t\}_{t \in \llbracket 0, T \rrbracket}$ are solutions of a backward iterative scheme (Theorem 18) involving linear Bellman Operators $\{\mathcal{T}_{t,L_{t+1}}^\ddagger\}_{t \in \llbracket 0, T-1 \rrbracket}$, thus opening the door to the computation of outer approximations $\{\underline{\mathcal{D}}_t\}_{t \in \llbracket 0, T \rrbracket}$ by SDDP.

```

Data: Initial primal point  $x_0$ , Lipschitz bounds  $\{L_t\}_{t \in \llbracket 0, T \rrbracket}$ 
for  $k = 0, 1, \dots$  do
    // Forward Pass : compute a set of trial points  $\{\lambda_t^{(k)}\}_{t \in \llbracket 0, T \rrbracket}$ 
    Compute  $\lambda_0^k \in \arg \max_{\|\lambda_0\|_\infty \leq L_0} \{x_0^\top \lambda_0 - \underline{\mathcal{D}}_0^k(\lambda_0)\}$  ;
    for  $t : 0 \rightarrow T - 1$  do
        select  $\lambda_{t+1}^k \in \arg \min \mathcal{T}_t^\ddagger(\underline{\mathcal{D}}_{t+1}^k)(\lambda_t^k)$  ;
        and draw a realization  $\lambda_{t+1}^k$  of  $\lambda_{t+1}^k$  ;
    end
    // Backard Pass : refine the lower-approximations at the trial points
    Set  $\underline{\mathcal{D}}_T^k = K^*$  . ;
    for  $t : T - 1 \rightarrow 0$  do
         $\bar{\theta}_t^{k+1} := \mathcal{T}_{t,L_{t+1}}^\ddagger(\underline{\mathcal{D}}_{t+1}^{k+1})(\lambda_t^k)$  ; // computing cut coefficients
         $\bar{x}_t^{k+1} \in \partial \mathcal{T}_{t,L_{t+1}}^\ddagger(\underline{\mathcal{D}}_{t+1}^{k+1})(\lambda_t^k)$  ;
         $\bar{\beta}_t^{k+1} := \bar{\theta}_t^{k+1} - \langle \lambda_t^k, \bar{x}_t^{k+1} \rangle$  ;
         $\mathcal{C}_t^{k+1} : \lambda \mapsto \langle \bar{x}_t^{k+1}, \lambda \rangle + \bar{\beta}_t^{k+1}$  ;
         $\underline{\mathcal{D}}_t^{k+1} = \max(\mathcal{D}_t^k, \mathcal{C}_t^{k+1})$  ; // update lower approximation
    end
    If some stopping test is satisfied STOP ;
end

```

Algorithm 3: Dual SDDP algorithm

Remark 20. In the forward phase of Algorithm 3, we need to solve $\mathcal{T}_t^\ddagger(\underline{\mathcal{D}}_{t+1}^k)(\lambda_t^k)$, which in extended form reads

$$\{\lambda_{t+1}^{k,\xi}\}_{\xi \in \text{supp}(\xi_{t+1})} \in \arg \min_{\{\lambda_{t+1}^\xi\}_{\xi \in \text{supp}(\xi_{t+1})}} \min_{\nu_{t+1} \geq 0} \sum_{\xi \in \text{supp}(\xi_{t+1})} \pi_{t+1}^\xi \left[- (C_{t+1}^\top \lambda_{t+1}^\xi + G_{t+1}^\top \nu_{t+1}^\xi)^\top \xi_{t+1}^\xi + \underline{\mathcal{D}}_{t+1}^k(\lambda_{t+1}^\xi) \right] \quad (22a)$$

$$\text{s.t.} \quad \sum_{\xi \in \text{supp}(\xi_{t+1})} \pi_{t+1}^\xi (A_t^\top \lambda_{t+1}^\xi + D_{t+1}^\top \nu_{t+1}^\xi) = \lambda_t^k \quad (22b)$$

$$c_{t+1} + B_{t+1}^\top \lambda_{t+1}^\xi + E_{t+1}^\top \nu_{t+1}^\xi = 0 \quad \forall \xi, \quad (22c)$$

$$\|\lambda_{t+1}^\xi\|_\infty \leq L_{t+1}. \quad (22d)$$

Then drawing a random realization of λ_{t+1}^k consists in drawing ξ with respect to the law of ξ_{t+1}

and selecting $\lambda_{t+1}^{k,\xi}$.

In contrast with primal SDDP algorithm (Remark 16), here we need to solve a coupled problem in both the forward and backward pass. In particular it means that we can also compute cuts during the forward pass, thus rendering the backward pass optional. \diamond

The convergence of this algorithm is analyzed in the next paragraph.

3.3 Upper-bound

To estimate an upper-bound of the optimal value of Problem (1), the seminal method consists in computing the expected cost of SDDP's strategy with a Monte-Carlo approach (see discussion in §1.2). This approach has two weaknesses: it requires a large number M of forward pass (simulation), and the bound obtained is only an upperbound with (asymptotic) probability α , where increasing α increases the bound as well. In contrast to the Monte Carlo method, we can obtain with dual SDDP a decreasing sequence of exact upper-bounds for Problem (1).

Lemma 21. *For all $t \in \llbracket 0, T-1 \rrbracket$, $(\underline{\mathcal{D}}_t^k)^*$ is a decreasing sequence of upper-bounds of the primal value function V_t : $(\underline{\mathcal{D}}_t^k)^* \geq V_t$.*

Proof. The sequence of functions $\underline{\mathcal{D}}_t^k$ is obtained by applying SDDP to the dual problem and we know by Proposition 13 that it gives an increasing sequence of lower-bounds of the function \mathcal{D}_t . By conjugacy property, we obtain a decreasing sequence of functions $(\underline{\mathcal{D}}_t^k)^*$ which are upper-bounds of the function $\mathcal{D}_t^* = V_t$. \square

We obtain the following theorem.

Theorem 22. *For any $k \in \mathbb{N}$, $(\underline{\mathcal{D}}_0^k)^*(x_0)$ is an upper-bound to the value $V(x_0)$ of Problem (1). Furthermore, $\lim_k (\underline{\mathcal{D}}_0^k)^*(x_0) = V_0(x_0)$*

Proof. To take care of the fact that the dual initial point is not constant, we add a fictive time step $t = -1$, and define $\mathcal{T}_{-1, L_0}^\ddagger$ as follows

$$\mathcal{T}_{-1, L_0}^\ddagger(R) := \min_{\lambda_0: \|\lambda_0\|_\infty \leq L_0} -x_0^\top \lambda_0 + R(\lambda_0).$$

then the algorithm presented in Sect. 3 is the abstract SDDP algorithm of §2.3 applied to the recursion $\mathcal{D}_t = \mathcal{T}_{t, L_{t+1}}^\ddagger(\mathcal{D}_{t+1})$ for $t \in \llbracket -1, T-1 \rrbracket$ and $\mathcal{D}_T = K^*$. The initial point is arbitrarily set to a fixed value 0 as \mathcal{D}_{-1} is constant.

We check that, by definition of $\mathcal{T}_{t, L_t}^\ddagger$, $\|\lambda_t^k\|_\infty \leq L_t$. Further, as V_0 is L_0 Lipschitz for the L_1 -norm, $\max_\lambda x_0^\top \lambda - V_0^*(\lambda)$ is attained for λ_0 such that $\|\lambda_0\|_\infty \leq L_0$, thus we have $\mathcal{D}_{-1}(0) = -[V]^{**}(x_0) = -V(x_0) \in \mathbb{R}$. Finally, note that $\{\mathcal{T}_{t, L_{t+1}}^\ddagger\}_{t \in \llbracket -1, T \rrbracket}$ is a compatible sequence of LBO.

Lemma 21 shows that, for any $k \in \mathbb{N}$, $-\underline{\mathcal{D}}_{-1}^k(0) = [\underline{\mathcal{D}}_0(x_0)]^*$ is an upper bound of $V_0(x_0)$. Finally, the convergence of the abstract SDDP algorithm and the lower semicontinuity of V_0 at x_0 yields the convergence of the upper bound. \square

4 Inner-approximation strategy

In Sect. 3, we detailed how to use the SDDP algorithm to get an outer approximation $\{\underline{\mathcal{D}}_t\}_{t \in \llbracket 0, T \rrbracket}$ of the dual value functions $\{\mathcal{D}_t\}_{t \in \llbracket 0, T \rrbracket}$. We now explain how to build an inner approximation of the primal value functions $\{V_t\}_{t \in \llbracket 0, T \rrbracket}$ using this dual outer approximation. Assume that, $(L_t)_{t \in \llbracket 0, T-1 \rrbracket}$ is given by Lemma 12.

4.1 Inner approximation of value functions

Let $\{\underline{\mathcal{D}}_t^k\}_{t \in \llbracket 0, T \rrbracket}$ be the outer approximation of the dual value functions $\{\mathcal{D}_t\}_{t \in \llbracket 0, T \rrbracket}$ obtained at iteration k of the dual SDDP algorithm. We denote by $\{(\bar{x}_t^\kappa, \bar{\beta}_t^\kappa)\}_{\kappa \in \llbracket 1, k \rrbracket}$ the cuts coefficients computed by the dual SDDP algorithm:

$$\underline{\mathcal{D}}_t^k(\lambda) = \min_{\theta} \theta, \quad (23a)$$

$$\text{s.t. } \theta \geq \langle \bar{x}_t^\kappa, \lambda \rangle + \bar{\beta}_t^\kappa \quad \forall \kappa \in \llbracket 1, k \rrbracket. \quad (23b)$$

We define the linear inner approximation of the primal value functions $\{V_t\}_{t \in \llbracket 0, T \rrbracket}$ as the Lipschitz regularization of the Fenchel conjugate of the dual outer approximation.

Definition 23. We define \bar{V}_t^k by

$$\bar{V}_t^k = \left[\underline{\mathcal{D}}_t^k \right]^* \square(L_t \|\cdot\|_1) \quad \forall t \in \llbracket 0, T \rrbracket. \quad (24)$$

The following proposition gives properties of \bar{V}_t^k .

Proposition 24. For all $t \in \llbracket 0, T \rrbracket$ we have

i) $\bar{V}_t^k \geq V_t$ on X_t .

ii) We have

$$\bar{V}_t^k(x) = \min_{y \in \mathbb{R}^{n_x}, \sigma \in \Delta} L_t \|x - y\|_1 - \sum_{\kappa=1}^k \sigma_\kappa \bar{\beta}_t^\kappa \quad (25a)$$

$$\text{s.t. } \sum_{\kappa=1}^k \sigma_\kappa \bar{x}_t^\kappa = y, \quad (25b)$$

where $\Delta = \{\sigma \in \mathbb{R}^k \mid \sigma \geq 0, \sum_{\kappa=1}^k \sigma_\kappa = 1\}$ is the simplex of \mathbb{R}^k .

iii) The inner approximation can be computed by solving

$$\bar{V}_t^{k+1}(x) = \sup_{\lambda, \theta} x^\top \lambda - \theta \quad (26a)$$

$$\text{s.t. } \theta \geq \langle \bar{x}_t^\kappa, \lambda \rangle + \bar{\beta}_t^\kappa \quad \forall \kappa \in \llbracket 1, k \rrbracket \quad (26b)$$

$$\|\lambda\|_\infty \leq L_t. \quad (26c)$$

iv) The Fenchel transform of the inner approximation is given by $\left[\bar{V}_t^k \right]^* = \underline{\mathcal{D}}_t^k + \chi_{B_\infty(0, L_t)}$.

Proof. i) Lemma 21 proves that $\left[\underline{\mathcal{D}}_t^k \right]^* \geq V_t$ for all $t \in \llbracket 0, T \rrbracket$. Thus $\bar{V}_t^k \geq V_t \square(L_t \|\cdot\|_1)$, which is equal to V_t on X_t as V_t is L_t -Lipschitz on its domain.

ii) Further, the Fenchel conjugate $\left[\underline{\mathcal{D}}_t^k \right]^*$ reads

$$\left[\underline{\mathcal{D}}_t^k \right]^*(x) = \sup_{\lambda, \theta} x^\top \lambda - \theta$$

$$\text{s.t. } \theta \geq \langle \bar{x}_t^\kappa, \lambda \rangle + \bar{\beta}_t^\kappa \quad \forall \kappa \in \llbracket 1, k \rrbracket,$$

which is a linear program whose dual reads

$$\begin{aligned} \min_{\sigma \in \Delta} & - \sum_{\kappa=1}^k \sigma_{\kappa} \bar{\beta}_t^{\kappa} \\ \text{s.t.} & \sum_{\kappa=1}^k \sigma_{\kappa} \bar{x}_t^{\kappa} = x . \end{aligned}$$

Taking the inf-convolution with $L_t \|\cdot\|$ yields Problem (25)

iii) The right hand side of Equation (26) is simply $[\mathcal{D}_t^k + \chi_{B_{\infty}(0, L_t)}]^*(x)$, which is equal to $[\underline{\mathcal{D}}_t^k]^* \square [\chi_{B_{\infty}(0, L_t)}]^*(x)$ by finite polyhedrality, hence the results.

iv) Finally,

$$[\bar{V}_t^k]^*(\lambda) = [\underline{\mathcal{D}}_t^k]^{**}(\lambda) + \chi_{B_{\infty}(0, L_t)}(\lambda) .$$

□

Figure 4.1 illustrates how to interpret the dual outer approximation as a primal inner approximation of the original value function (in black). The slopes x_1, x_2, x_3 computed for the dual outer approximation (blue curve, right) are breakpoints for the primal problem and we can consider the value of the dual value function at these points to build a primal inner approximation (blue curve, left).

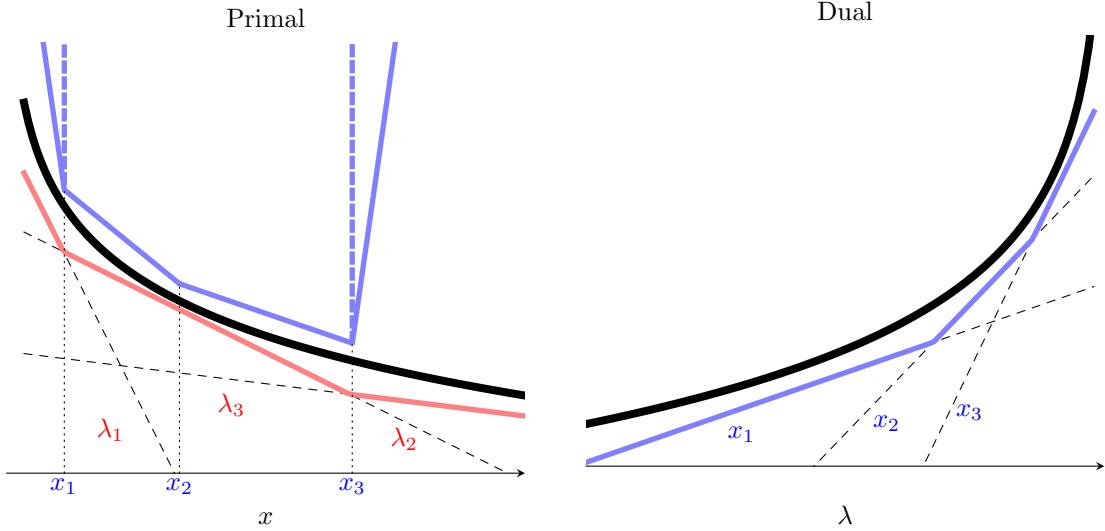


Figure 1: Primal SDDP computes an outer approximation (in red) of the original value function (in black). Dual SDDP computes an outer approximation in the dual, whose Fenchel-transform (in blue) yields an inner approximation of the primal problem.

4.2 A bound on the inner-approximation strategy value

Hence, we have obtained inner approximations of the primal value functions. Such approximations can be used to define an admissible strategy for the initial problem. We now study the properties of such a strategy.

Lemma 25. *We have,*

$$\mathcal{T}_{t,L_{t+1}}^\ddagger(\underline{\mathcal{D}}_{t+1}^k) \geq \underline{\mathcal{D}}_t^k \quad \forall t \in \llbracket 0, T \rrbracket. \quad (27)$$

Proof. Equation (27) is satisfied for $k = 0$. Assume that Equation (27) holds at iteration k . By definition of \mathcal{C}^{k+1} in Algorithm 3, we have $\mathcal{C}^{k+1} \leq \mathcal{T}_{t,L_{t+1}}^\ddagger(\underline{\mathcal{D}}_{t+1}^{k+1})$. On the other hand, by monotonicity of $\mathcal{T}_{t,L_{t+1}}^\ddagger$, since $\underline{\mathcal{D}}_{t+1}^{k+1} \geq \underline{\mathcal{D}}_{t+1}^k$, we have $\mathcal{T}_{t,L_{t+1}}^\ddagger(\underline{\mathcal{D}}_{t+1}^{k+1}) \geq \mathcal{T}_{t,L_{t+1}}^\ddagger(\underline{\mathcal{D}}_{t+1}^k)$ which is greater than $\underline{\mathcal{D}}_t^k$ by induction hypothesis. Thus, $\mathcal{T}_{t,L_{t+1}}^\ddagger(\underline{\mathcal{D}}_{t+1}^{k+1}) \geq \max\{\underline{\mathcal{D}}_t^k, \mathcal{C}^{k+1}\} = \underline{\mathcal{D}}_t^{k+1}$. \square

From the convergence proof of SDDP, we recall the following technical Lemma.

Lemma 26. *Let \bar{V}_t^k be the inner approximation of the value function V_t generated at iteration k of the dual SDDP algorithm. Then,*

$$\mathcal{T}_t(\bar{V}_{t+1}^k)(x) \leq \bar{V}_t^k(x) \quad \forall t \in \llbracket 0, T \rrbracket. \quad (28)$$

Proof. We have

$$\begin{aligned} [\mathcal{T}_t(\bar{V}_{t+1}^k)]^\star &= \mathcal{T}_t^\ddagger([\bar{V}_{t+1}^k]^\star) && \text{by Proposition 10} \\ &= \mathcal{T}_t^\ddagger(\underline{\mathcal{D}}_{t+1}^k + \chi_{B_\infty(0, L_{t+1})}) && \text{by Proposition 24} \\ &= \mathcal{T}_{t,L_{t+1}}^\ddagger(\underline{\mathcal{D}}_{t+1}^k) \\ &\geq \underline{\mathcal{D}}_t^k && \text{by Lemma 25} \end{aligned}$$

Furthermore, as $\mathcal{T}_t(\bar{V}_{t+1}^k)$ is polyhedral, we have

$$\mathcal{T}_t(\bar{V}_{t+1}^k) = [\mathcal{T}_t(\bar{V}_{t+1}^k)]^{\star\star} \leq [\underline{\mathcal{D}}_t^k]^\star,$$

and as \bar{V}_{t+1}^k is L_{t+1} -Lipschitz, then $\mathcal{T}_t(\bar{V}_{t+1}^k)$ is L_t -Lipschitz, thus $\mathcal{T}_t(\bar{V}_{t+1}^k) \leq [\underline{\mathcal{D}}_t^k]^\star \square(L_t \|\cdot\|)$ which ends the proof. \square

We are now able to state the main result of this section.

Theorem 27. *Let $\{\mathbf{X}_t^{IA}, \mathbf{U}_t^{IA}\}_{t \in \llbracket 0, T-1 \rrbracket}$ be the state and control processes obtained by applying the strategy induced by the inner approximation $\{\bar{V}_t^k\}_{t \in \llbracket 0, T \rrbracket}$, that is, $(\mathbf{X}_{t+1}^{IA}, \mathbf{U}_{t+1}^{IA}) \in \mathcal{S}(\bar{V}_{t+1}^k)(\mathbf{X}_t^{IA})$. Consider the expected cost of this strategy when starting from state x at time t :*

$$C_t^{IA}(x) = \mathbb{E} \left(\sum_{\tau=t}^{T-1} a_\tau^\top \mathbf{X}_\tau^{IA} + b_{\tau+1}^\top \mathbf{U}_{\tau+1}^{IA} + K(\mathbf{X}_T^{IA}) \mid \mathbf{X}_t^{IA} = x \right).$$

Then,

$$C_t^{IA}(x) \leq \bar{V}_t^k(x). \quad (29)$$

Proof. We proceed by backward induction on time t . The property holds for $t = T$.

Assume that $C_{t+1}^{IA} \leq \bar{V}_{t+1}^k$. We have

$$\begin{aligned}
C_t^{IA}(x) &= \mathbb{E}[a_t^\top x + b_{t+1}^\top \mathbf{U}_{t+1}^{IA} + C_{t+1}^{IA}(\mathbf{X}_{t+1}^{IA})] \\
&\leq \mathbb{E}[a_t^\top x + b_{t+1}^\top \mathbf{U}_{t+1}^{IA} + \bar{V}_{t+1}^k(\mathbf{X}_{t+1}^{IA})] && \text{by induction} \\
&= \mathcal{T}_t(\bar{V}_{t+1}^k)(x) && \text{by definition of } \mathbf{U}_{t+1}^{IA} \\
&\leq \bar{V}_t^k(x) && \text{by Lemma 26}
\end{aligned}$$

hence the result. \square

Corollary 28. *The cost $C_0^{IA,k}(x_0)$ of the IA strategy converges to $V_0(x_0)$.*

$$\lim_{k \rightarrow +\infty} C_0^{IA,k}(x_0) = V_0(x_0) . \quad (30)$$

Proof. Let k be the number of iterations performed by SDDP, and $C_0^{IA,k}(x_0)$ the expected cost of the inner strategy at iteration k . By definition of $V_0(x_0)$, we have $C_0^{IA,k}(x_0) \geq V_0(x_0)$. Furthermore, using Theorem 27, we obtain the following inequalities

$$V_0(x_0) \leq C_0^{IA,k}(x_0) \leq \bar{V}_0^k(x_0) . \quad (31)$$

Using Theorem 22, we know that $\lim_k(\bar{V}_0^k)(x_0) = V_0(x_0)$. Hence the result. \square

Remark 29. A similar result on the performance of an inner approximation is given in Philpott et al. [2013]. As already explained in §1.2, the authors construct polyhedral inner approximations \bar{V}_t of the Bellman functions V_t . They then prove that the expected cost of the policy based on the functions \bar{V}_t is always less than or equal to the deterministic upper bound given by the inner approximation algorithm. \diamond

We sum up the available inequalities for the values obtained when implementing the primal and dual SDDP algorithms.

$$\underline{V}_0(x_0) \leq V_0(x_0) \leq \bar{V}_0(x_0) , \quad (32a)$$

$$\underline{V}_0(x_0) \leq C_0^{IA}(x_0) \leq \bar{V}_0(x_0) , \quad (32b)$$

$$\underline{V}_0(x_0) \leq C_0^{OA}(x_0) . \quad (32c)$$

Equation (32a) corresponds to the deterministic bounds of the optimal value of Problem (1), whereas Equations (32b) and (32c) are of statistical nature.

5 Numerical results

In this section, we present some numerical results applying dual SDDP and inner strategy evaluation to a stochastic operation planning problem inspired by Électricité de France (EDF, main European electricity producer). The problem is about the energy production planning on a multiperiod horizon including a network of production zones, like in the European Market for electricity. It results in a large-scale stochastic multi-stage optimization problem, for which we need to determine strategies for the management of the European water dams. Such strategies cannot be computed via Dynamic Programming because of the state variable size, so that SDDP is the reference method to compute the optimal Bellman functions.

5.1 Description of the problem

We consider an operation planning problem at the European scale. Different countries are connected together via a network, and exchange energy with their neighbors. We formulate the problem on a graph, where each country is modeled as a node and each interconnection line between two countries as an edge (see Figure 2). Every country uses a reservoir to store energy,

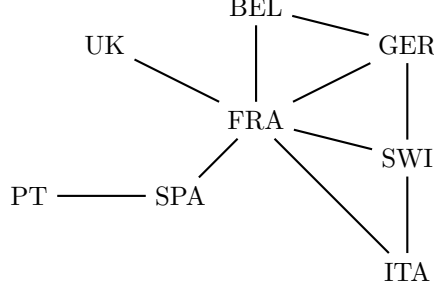


Figure 2: Schematic description of the European network

and must fulfill its own energy demand. To do so, it can produce energy from its reservoir, with its local thermal power plant, or it can import energy from the other countries. A very similar problem has already been studied by Mahey et al. [2017]. Its formulation is close to the one given in Shapiro et al. [2012] concerning the Brazilian interconnected power system.

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ the graph modeling the European network. The number of nodes in \mathcal{N} is denoted by n and the number of edges in \mathcal{E} by ℓ . For each node $i \in \llbracket 1, n \rrbracket$, we denote by v_t^i the energy stored in the reservoir at time t . The reservoir's dynamics is given by

$$v_{t+1}^i = v_t^i + a_{t+1}^i - q_{t+1}^i - s_{t+1}^i, \quad (33)$$

where a_{t+1}^i is the (random) water inflow in the reservoir and q_{t+1}^i is the water turbinated between time t and $t + 1$ in order to produce electricity. We add a spillage s_{t+1}^i as recourse variable to avoid to overthrow the reservoir. Still at node i , the *load balance* equation at stage t writes

$$q_t^i + g_t^i + \sum_{j \in N_i} f_t^{ji} + r_t^i = d_t^i, \quad (34)$$

where g_t^i is the thermal production, $N_i \subset \mathcal{N}$ is the set of nodes connected to node i and f_t^{ji} is the energy exchanged between nodes $j \in N_i$ and node i , d_t^i is the (random) demand of the node and $r_t^i \geq 0$ is a recourse variable added to ensure that the load balance is always satisfied. The thermal production g_t^i and the exchanges f_t^{ji} between node i and nodes $j \in N_i$ induce linear costs, and the cost of the recourse variable r_t^i is taken into account through a linear penalization. Hence the total cost attached to node i at time t writes

$$c_t^i g_t^i + \delta_t^i r_t^i + \sum_{j \in N_i} p_t^{ji} f_t^{ji}, \quad (35)$$

where c_t^i is the thermal price, δ_t^i is the recourse price and p_t^{ji} is the transportation price between nodes j and i . To avoid empty stocks at the end of the time horizon, we penalize the final stock at each node i if it is beyond a threshold v_0^i using a piecewise linear function:

$$K^i(v_T^i) = \kappa_T^i \max(0, v_0^i - v_T^i). \quad (36)$$

Stocks and controls are bounded:

- $0 \leq v_t^i \leq \bar{v}^i$, reservoir volume upper bound,
- $0 \leq q_t^i \leq \bar{q}^i$, reservoir generation upper bound,
- $0 \leq g_t^i \leq \bar{g}^i$, thermal generation upper bound,
- $0 \leq r_t^i$, recourse control lower bound,
- $\underline{f}^{ji} \leq f_t^{ji} \leq \bar{f}^{ji}$, energy flow lower and upper bound.

This problem is formulated as a stochastic optimal control problem, where for all t ,

- the state is $v_t = (v_t^1, \dots, v_t^n)$ (denoted x_t in §3),
- the control is $u_t = (q_t, s_t, g_t, r_t, f_t)$, with $q_t = (q_t^i)_{i \in [1, n]}$ and likewise for s_t, g_t, r_t and f_t ,
- the uncertainty is $\xi_t = (a_t^i, d_t^i)_{i \in [1, n]}$.

The state has dimension n , the control u_t dimension $4n + \ell$ and the uncertainty ξ_t dimension $2n$. We assume that the random variables ξ_t have a discrete finite support. For a given realization (a_{t+1}, d_{t+1}) of the uncertainty, the primal Bellman operator $\widehat{\mathcal{T}}_t$, defined by (17), writes

$$\widehat{\mathcal{T}}_t(V_{t+1})(v_t, (a_{t+1}, d_{t+1})) = \min_{v_{t+1}, q_{t+1}, s_{t+1}, g_{t+1}, r_{t+1}, f_{t+1}} \sum_{i=1}^n (c_{t+1}^i g_{t+1}^i + \delta_{t+1}^i r_{t+1}^i + \sum_{j \in N_i} p_{t+1}^{ji} f_{t+1}^{ji}) + V_{t+1}(v_{t+1}), \quad (37a)$$

$$\text{s.t. } v_{t+1}^i = v_t^i - q_{t+1}^i - s_{t+1}^i + a_{t+1}^i, \quad (37b)$$

$$q_{t+1}^i + g_{t+1}^i + \sum_{j \in N_i} f_{t+1}^{ji} + r_{t+1}^i = d_{t+1}^i, \quad (37c)$$

$$0 \leq v_{t+1}^i \leq \bar{v}^i, \quad 0 \leq q_{t+1}^i \leq \bar{q}^i, \quad 0 \leq r_{t+1}^i, \quad (37d)$$

$$0 \leq g_{t+1}^i \leq \bar{g}^i, \quad \underline{f}^{ji} \leq f_{t+1}^{ji} \leq \bar{f}^{ji}. \quad (37e)$$

We rewrite Problem (37) in the standard form (1) with matrices (A, B, C) for the dynamics and (D, E, G) for the constraints. We note that A is the identity matrix I_n . The expressions of these matrices are given in §A.1. Then the expression (20) of the dual Bellman operator \mathcal{T}_t^\ddagger is obtained in a straightforward manner, namely

$$\mathcal{T}_{t, L_{t+1}}^\ddagger(\mathcal{D}_{t+1})(\lambda_t) = \inf_{\lambda_{t+1}^\xi, \nu_{t+1}^\xi \geq 0} \sum_{\xi \in \text{supp}(\xi_{t+1})} \pi_\xi [- (\xi_{t+1}^\xi)^\top C^\top \lambda_{t+1}^\xi - g_{t+1}^\top \nu_{t+1}^\xi + \mathcal{D}_{t+1}(\lambda_{t+1}^\xi)] \quad (38a)$$

$$\text{s.t. } \sum_{\xi \in \text{supp}(\xi_{t+1})} \pi_\xi (\lambda_{t+1}^\xi + D^\top \nu_{t+1}^\xi) = \lambda_t \quad (38b)$$

$$\bar{c}_{t+1} + B^\top \lambda_{t+1}^\xi + E^\top \nu_{t+1}^\xi = 0 \quad \forall \xi, \quad (38c)$$

$$\| \lambda_{t+1}^\xi \|_\infty \leq L_{t+1}, \quad (38d)$$

where L_t a Lipschitz constant of V_{t+1} .

Remark 30. Problem (38) cannot be decomposed into S independent optimization subproblems because of the coupling constraint (38b), so that the number of constraints in the problem is proportional to the support size S of the uncertainty ξ_{t+1} . Indeed, the number of constraint defined by Equation (38c) is equal to $S \times m$. Furthermore, if the value function \mathcal{D}_{t+1} is approximated by k cuts $(\bar{x}_{t+1}^l, \bar{\beta}_{t+1}^l)_{l \in \llbracket 1, k \rrbracket}$ (as the SDDP algorithm does), then we have to add k constraints of the form $\theta \geq \langle \bar{x}_{t+1}^l, \lambda_{t+1}^\xi \rangle + \bar{\beta}_{t+1}^l$ for each term $\mathcal{D}_{t+1}(\lambda_{t+1}^\xi)$ present in the cost function (38a), thus adding $S \times k$ constraints in Problem (38). The consequence for the numerical resolution is that the bigger the quantization size S , the slower the resolution by dual SDDP. \diamond

5.2 Numerical implementation

The forward and backward passes of dual SDDP are *independent* from the forward and backward passes of primal SDDP. Accordingly, a first “natural” implementation of the whole algorithm run primal and dual SDDP in two independent processes, thus able to compute primal and dual value functions in parallel.

However, each backward pass of the primal SDDP algorithm computes a set of cuts whose slopes are $\{\lambda_t\}_{t \in \llbracket 0, T \rrbracket}$. As explained in Figure 4.1, these slopes can be considered as trajectories for the dual problem. If primal SDDP has converged, they are even the optimal co-state of the problem, because of the Fenchel-Young equality. Thereby, it may prove useful to view these sequences of slopes as trajectories for the dual problem, along which we run afterward a backward pass producing cuts for the dual problem. In this implementation, the algorithm becomes for each iteration:

1. Run a forward pass of primal SDDP Algorithm 2 and get trajectories $\{x_t\}_{t \in \llbracket 0, T \rrbracket}$.
2. Run a backward pass of primal SDDP Algorithm 2 along $\{x_t\}_{t \in \llbracket 0, T \rrbracket}$ and obtain new slopes $\{\lambda_t\}_{t \in \llbracket 0, T \rrbracket}$.
3. Run a backward pass of dual SDDP Algorithm 3 along $\{\lambda_t\}_{t \in \llbracket 0, T \rrbracket}$, thus updating the sets of cuts for the dual problem.
4. Run a forward pass of dual SDDP Algorithm 3 and update the cuts along the obtained trajectories.

The complete algorithm is given in §A.2.

The last step of this iteration ensures that we recover the convergence hypotheses of SDDP, as given in Girardeau et al. [2014], by having one set of cuts computed at point sampled along uncertainty drawn independently from the past. This algorithm has the same number of forward and backward passes as the original one (one forward pass and one backward pass in both the primal and the dual space). However, this scheme proves to be numerically more efficient, both in term of convergence and computation time. That is why we use this implementation in all the numerical experiments.

From the computational point of view, we implement primal and dual SDDP in Julia 0.6, with the `StochDynamicProgramming.jl` package built on top of the JuMP modeler of Dunning et al. [2017]. We use Gurobi 7.02 to solve the LP subproblems. All experiments are run on a Intel Core i7-5500 CPU @2.4GHz, 64bit computer.

5.3 Results

We consider the problem described at §5.1, which exhibits a 8-dimensional state. We aim to compute the value functions $\{V_t\}_{t \in \llbracket 0, T \rrbracket}$ with monthly time steps, and we consider different time horizons T , depending on the desired goal (illustration of convergence, comparison of bounds. . .). The uncertainties in the model are the inflows a_t in the reservoir and the demands d_t in every considered countries. Inflows and demands trajectories are simulated using a software provided by EDF, so that these data are realistic enough. From these simulated samples, we use quantization methods to obtain the marginal laws of the uncertainty ξ_t at each $t \in \llbracket 0, T \rrbracket$. The support of the quantized probability laws is limited to 10 possible values for ξ_t at each timestep t .

To solve the problem, we run primal and dual SDDP on 1,000 iterations, with a single forward pass in the primal and in the dual.

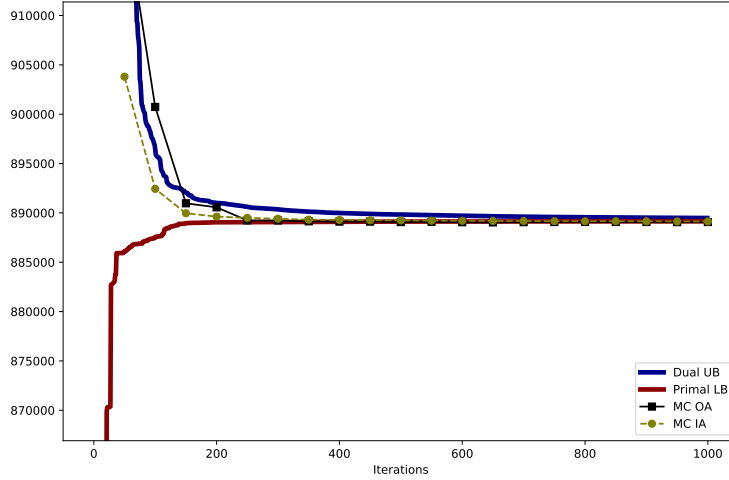
5.3.1 Assessing convergence

To ease the description of the results, we denote by **Primal LB** the primal lower bound $\underline{V}_0(x_0)$ obtained by primal SDDP, and by **Dual UB** the upper bound $[\underline{\mathcal{D}}_0]^*(x_0)$ given by dual SDDP. The Monte Carlo cost evaluation obtained by simulating the outer (resp. inner) strategy uses the procedure described in §3.1.2, and is denoted by **MC OA** (resp. **MC IA**). Confidence intervals (with a confidence level $\alpha = 97.5\%$) are associated to these Monte Carlo approximations, and we denote by **MC OA UB** and **MC IA UB** the associated upper bounds of these intervals. Whereas **Primal LB** and **Dual UB** are deterministic bounds, **MC OA**, **MC IA**, **MC OA UB** and **MC IA UB** are statistical quantities.

Solving the problem over a one-year time horizon. First, we run dual and primal SDDP on a twelve months problem, that is, with $T = 12$. Convergence of the optimal costs given by dual and primal SDDP is detailed in Figure 3. The two last columns in the table give the cumulative computation times needed to run both primal and dual SDDP algorithms. We observe that the upper bound **Dual UB** $[\underline{\mathcal{D}}_0]^*(x_0)$ given by dual SDDP converges towards the primal lower bound **Primal LB** $\underline{V}_0(x_0)$ given by primal SDDP, with a relative gap close to 0.03% after 1,000 iterations. For this specific (with few time steps) example, the convergence of dual SDDP proves to be effective. As noticed at Remark 30, running dual SDDP is much more time consuming as running primal SDDP.

The outer and inner strategies are evaluated by Monte Carlo. An evaluation is performed every 50 iterations with an once for all given set of 10,000 scenarios, that is, a “large” sample. Both evaluations **MC OA** and **MC IA** converge to the optimal value. We notice that **MC IA** is below **Dual UB**, thus illustrating the result stated by Theorem 27.

Solving the problem over a three years time horizon. We now consider the same problem, but over a three years horizon, that is, with $T = 36$. The convergence of primal and dual SDDP is shown in Figure 4. Compared to Figure 3, we have materialized the confidence intervals (here very thin) of the inner and outer strategies Monte Carlo simulations, both estimated every 50 iterations with an once for all given set of 10,000 scenarios. A first observation is that both dual and primal SDDP exhibit a slower convergence than in the first example: after 1,000 iterations, the gap between the primal lower bound **Primal LB** and the dual upper bound **Dual UB** is equal to 0.13%. This well-known behavior of SDDP arises from the increasing number of time-steps (36 instead of 12). Moreover, **Dual UB** is still significantly decreasing after iteration 500, and it seems that it converges more slowly than **Primal LB**.



Iter.	Primal LB ($\times 10^5$)	Dual UB ($\times 10^5$)	Gap (%)	Time LB (s)	Time UB (s)
50	8.861	9.577	8.08	2.	8.
100	8.874	8.969	1.06	3.	22.
200	8.890	8.910	0.21	8.	72.
300	8.891	8.904	0.14	13.	153.
400	8.891	8.900	0.09	20.	275.
500	8.891	8.898	0.08	29.	443.
600	8.891	8.897	0.06	38.	651.
700	8.891	8.896	0.05	49.	888.
800	8.891	8.896	0.04	61.	1191.
900	8.891	8.895	0.04	74.	1534.
1000	8.891	8.895	0.03	89.	1928.

Figure 3: Convergence of primal and dual SDDP for $T = 12$. Time corresponds to cumulated time along iterations.

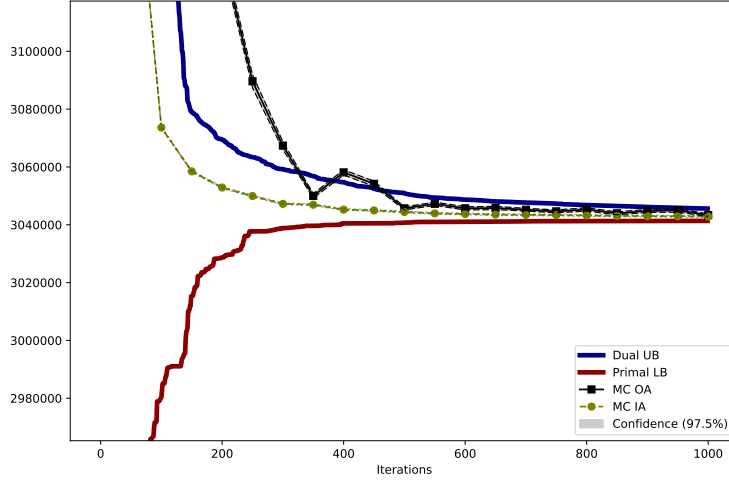
A second observation is that the dual upper bound `Dual UB` is better than the statistical cost value `MC OA` up to iteration 500. After the first 500 iterations, `MC OA` is better than `Dual UB` and slightly fluctuates above the primal lower bound `Primal LB` (the remaining gap being around 0.1% after 1,000 iterations).

Finally, on this example, the value of `MC OA` is greater than the value of `MC IA` at every iteration. Surprisingly enough, the value of `MC IA` exhibits a more stable behavior than the one given by `MC OA`. It would be interesting to be able to assess such behaviors.

5.3.2 Using the dual upper bound in a stopping criteria

Consider the problem over a three years time horizon. The gap between the two deterministic bounds (primal lower bound `Primal LB` and dual upper bound `Dual UB`) against the number of iterations is given in Figure 4. To complete these results, we give the evolution of the statistical upper bound `MC OA UB` obtained by the outer strategy in Table 1. We aim at comparing two stopping tests.

Statistical stopping test: it is the stopping test proposed in Shapiro [2011] and which has been detailed at §1.2. We choose a confidence level $\alpha = .975$, and we estimate the statistical upper bound `MC OA UB` every 50 iterations with a given set of 10,000 scenarios.



Iter.	Primal LB ($\times 10^6$)	Dual UB ($\times 10^6$)	Gap (%)	Time LB (s)	Time UB (s)
50	2.837	3.917	38.1	5.	20.
100	2.980	3.151	5.7	11.	74.
200	3.029	3.070	1.4	27.	267.
300	3.039	3.059	0.67	46.	592.
400	3.040	3.055	0.46	75.	1113.
500	3.041	3.051	0.34	108.	1783.
600	3.041	3.049	0.25	144.	2601.
700	3.041	3.048	0.21	187.	3585.
800	3.041	3.047	0.18	235.	4751.
900	3.041	3.046	0.15	296.	6140.
1000	3.041	3.046	0.13	360.	7545.

Figure 4: Convergence of primal and dual SDDP for $T = 36$. Time corresponds to cumulated time along iterations.

Iter.	Primal LB ($\times 10^6$)	Gap Dual UB (%)	MC OA UB ($\times 10^6$)	Gap MC OA UB (%)
50	2.837	38.1	3.392	19.6
100	2.980	5.7	3.310	11.1
200	3.029	1.4	3.137	3.6
300	3.039	0.67	3.069	1.0
400	3.040	0.46	3.059	0.62
500	3.041	0.34	3.046	0.18
600	3.041	0.25	3.046	0.18
700	3.041	0.21	3.046	0.15
800	3.041	0.18	3.046	0.16
900	3.041	0.16	3.045	0.14
1000	3.041	0.13	3.044	0.08

Table 1: Statistical upper bound for $T = 36$.

Dual stopping test: this stopping test is just based on the gap between the available deterministic upper and lower bounds, namely Dual UB and Primal LB.

For different accuracy levels ε , as described by Shapiro [2011] we compare the CPU times taken by these two tests in order to stop the SDDP algorithm. Results are given in Table 2. The given times correspond to the total time required to run SDDP (including both the computation of cuts

ε (%)	Dual stopping test		Statistical stopping test	
	n it.	CPU time	n it.	CPU time
2.0	156	183s	250	618s
1.0	236	400s	300	787s
0.5	388	1116s	450	1429s
0.1	> 1000	.	1000	5519s

Table 2: Comparing dual and statistical stopping criteria for different accuracy levels ε .

and the computation of the stopping test). We notice that the dual stopping test gives better results than the statistical stopping test: for $\varepsilon \geq 0.45\%$, it stops SDDP earlier and require less computation time. Compared with the statistical test, the speed-up is between 3.3 for $\varepsilon = 2\%$ and 1.3 for $\varepsilon = 0.5\%$. However, the dual stopping test is penalized by the slow convergence of dual SDDP. Indeed it cannot achieve a gap lower than 0.1 %, thus penalizing the performance of the dual stopping test for high accuracy levels ε .

As a conclusion of these numerical experiments, the deterministic dual stopping test seems to be better than the statistical stopping test, especially if restrictions on the CPU time impose to perform a limited number of SDDP iterations (less than 500 in our case). Such a situation exists in the energy field, as shown by the description of the Brazilian interconnected power system in Shapiro et al. [2012].

Remark 31. We can also use the statistical upper bound MC IA UB obtained by evaluating the inner strategy for the statistical stopping test designed by Shapiro. Indeed, in our numerical experiments, this upper bound is always lower than the one given by the outer strategy. However, this would require much longer computational time, as this approach combines the computation of the dual cuts together with a Monte-Carlo estimation. \diamond

5.3.3 Strengths and weaknesses of dual SDDP

Dual SDDP allows us to obtain a deterministic stopping criterion, which proves to be effective compared to the standard statistical stopping test. Furthermore, dual SDDP computes cuts that can be used to design an inner approximation strategy, which appears to be better than the outer strategy whenever primal SDDP has not exactly converged.

However, we observe that the convergence of dual SDDP is penalized by different considerations.

- It is well-known from Shapiro [2011] that the convergence of SDDP is impacted by the number of stages in the problem. This issue impacts both primal and dual SDDP.
- Furthermore, we notice that dual SDDP exhibits a slower convergence than primal SDDP. In fact, primal SDDP computes its trajectories from a fixed initial point x_0 , whereas dual SDDP updates its initial point λ_0 at each iteration, with

$$\lambda_0^k \in \arg \max_{\|\lambda_0\| \leq L_0} \left\{ x_0^\top \lambda_0 - \underline{\mathcal{D}}_0^k(\lambda_0) \right\} \quad (39)$$

- One iteration of dual SDDP takes longer than one iteration of primal SDDP. Indeed, dual SDDP solves bigger LP problems than primal SDDP, as it has to consider explicitly a coupling constraint (38b) between all samples. Dual SDDP would greatly benefit from a cuts selection algorithm, which would limit the number of constraints added in the problem.

6 Conclusion

In this paper, we give a new method to compute exact upper-bounds for SDDP, which relies on applying SDDP to the Fenchel transform of Bellman's value functions. We have shown that under classical assumptions the upperbound converges.

We have taken advantage of the dual value functions $\{\mathcal{D}_t\}_{t \in [0, T]}$ to build a set of inner approximations $\{\bar{V}_t\}_{t \in [0, T]}$ of the primal value function $\{V_t\}_{t \in [0, T]}$. We proved that the policy induced by these inner approximations is converging to an optimal policy, with guaranteed performance of the expected cost associated. We tested dual SDDP with extensive numerical results on a stochastic production planning problem. These numerical results proved the effectiveness of dual SDDP and the underlying inner strategy. Furthermore, we showed on this particular problem that using a dual stopping test outperforms the classical statistical stopping test of SDDP, both in term of iterations number and in term of computational burden.

We plan to extend this study in several directions. First, an extension of dual SDDP risk averse or distributionally robust problems remains to be investigated. Second, the dual SDDP algorithm used in this paper does not decompose the subproblems at each time steps uncertainty by uncertainty, so as to obtain smaller subproblems with a more tractable size. This penalizes dual SDDP towards primal SDDP, leading to greater computation times. A proper way to decompose the dual subproblems effectively is under study. Third, we want to test dual SDDP on other problem, such as the Brazilian hydro power operation planning problem presented in Shapiro et al. [2012]. Finally we want to explore the interactions between primal and dual SDDP. For example, we think that the upper-bounds given by dual SDDP might be effective to regularize SDDP, for instance with the method introduced by Van Ackooij et al. [2017].

A Appendix

A.1 Numerical settings

We describe the problem in §5 with dynamics and constraint matrix.

$$A = I_n, \quad B = -(I_n \ I_n \ 0_n \ 0_n \ 0_{qn}), \quad C = (I_n \ 0_n), \quad (40)$$

where I_n is the identity matrix and 0_n the square null matrix with size n . The costs vector becomes $a_t = 0$ and $b_t = (0 \ 0 \ c_t \ t_t \ p_t)^\top$. The constraints matrix write

$$D = (0_n \ 0_n \ 0_n \ 0_n \ 0_n \ 0_n \ 0_{qn} \ 0_{qn} \ A \ -A)^\top, \quad (41)$$

and

$$E = \begin{pmatrix} I_n & 0_n & 0_n & 0_n & 0_{qn} \\ -I_n & 0_n & 0_n & 0_n & 0_{qn} \\ 0_n & -I_n & 0_n & 0_n & 0_{qn} \\ 0_n & 0_n & I_n & 0_n & 0_{qn} \\ 0_n & 0_n & -I_n & 0_n & 0_{qn} \\ 0_n & 0_n & 0_n & -I_n & 0_{qn} \\ 0_{qn} & 0_{qn} & 0_{qn} & 0_{qn} & I_q \\ 0_{qn} & 0_{qn} & 0_{qn} & 0_{qn} & -I_q \\ I_n & 0_n & I_n & I_n & R \\ & B & & & \\ & -B & & & \end{pmatrix}, \quad (42)$$

and

$$g_{t+1} = (\bar{q} \ 0 \ 0 \ \bar{g} \ 0 \ 0 \ \bar{f} \ \underline{f} \ \mathbf{d}_{t+1} \ \mathbf{d}_{t+1} \ (\bar{v} - \mathbf{a}_{t+1}) \ \mathbf{a}_{t+1})^\top \quad (43)$$

A.2 Exhaustive dual SDDP algorithm

Data: Lipschitz bounds $NL_{tt \in \llbracket 0, T \rrbracket}$

for $k \in \mathbb{N}$ **do**

Draw a noise scenario $\{\xi_t^k\}_{t \in \llbracket 0, T \rrbracket}$;

begin

Primal Forward Pass : compute a set of trial points $\{x_t^k\}_{t \in \llbracket 0, T \rrbracket}$;

Primal Backward Pass: refine primal value functions \underline{V}_t^k along $\{x_t^k\}_{t \in \llbracket 0, T \rrbracket}$;

Fetch computed cuts $\{\lambda_t^{k+1}\}_{t \in \llbracket 0, T \rrbracket}$;

Dual Backward Pass: refine dual value functions $\{\underline{\mathcal{D}}_{t+1}^{k+\frac{1}{2}}\}_{t \in \llbracket 0, T \rrbracket}$
along $\{\lambda_t^{k+1}\}_{t \in \llbracket 0, T \rrbracket}$;

end

Draw a new noise scenario $\{\bar{\xi}_t^k\}_{t \in \llbracket 0, T \rrbracket}$;

begin

Compute $\lambda_0^k \in \arg \max_{\|\lambda_0\| \leq L_0} \{x_0^\top \lambda_0 - \underline{\mathcal{D}}_0^k(\lambda_0)\}$;

Dual Forward Pass: compute a set of trial points $\{\bar{\lambda}_t^k\}_{t \in \llbracket 0, T \rrbracket}$ and update directly
dual value functions $\{\underline{\mathcal{D}}_{t+1}^{k+1}\}_{t \in \llbracket 0, T \rrbracket}$;

end

end

Algorithm 4: Primal-Dual SDDP algorithm

A.3 Compatibility of the primal and dual Bellman operators

We consider Problem (1) and its associated recursive Bellman equation:

$$\begin{aligned}
 V_T(x) &= K(x) , \\
 V_t(x) &= \inf_{\mathbf{U}, \mathbf{Y}} \mathbb{E} \left[a_t^\top x + b_{t+1}^\top \mathbf{U} + V_{t+1}(\mathbf{Y}) \right] , \\
 \text{s.t. } \quad \mathbf{Y} &= A_t x + B_{t+1} \mathbf{U} + C_{t+1} \boldsymbol{\xi}_{t+1} , \\
 \mathbf{U} &\in [\underline{u}_{t+1}, \bar{u}_{t+1}] , \\
 \mathbf{Y} &\in [\underline{x}_{t+1}, \bar{x}_{t+1}] , \\
 D_t x + E_{t+1} \mathbf{U} + G_{t+1} \boldsymbol{\xi}_{t+1} &\leq 0 .
 \end{aligned}$$

Ignoring the constant term $a_t^\top x$, this equation rewrites as a linear Bellman operator

$$\begin{aligned}
 \mathcal{B}_t(V_{t+1})(x) &= \inf_{\mathbf{U}, \mathbf{Y}} \mathbb{E} \left[b_{t+1}^\top \mathbf{U} + V_{t+1}(\mathbf{Y}) \right] , \\
 \text{s.t. } \quad Tx + \mathcal{W}_u(\mathbf{U}) + \mathcal{W}_y(\mathbf{Y}) &\leq \mathbf{H} ,
 \end{aligned}$$

with the notation:

$$T = \begin{bmatrix} A_t \\ -A_t \\ 0 \\ 0 \\ 0 \\ 0 \\ D_t \end{bmatrix}, \quad \mathcal{W}_u = \begin{bmatrix} B_{t+1} \\ -B_{t+1} \\ I \\ -I \\ 0 \\ 0 \\ E_{t+1} \end{bmatrix}, \quad \mathcal{W}_y = \begin{bmatrix} -I \\ I \\ 0 \\ 0 \\ I \\ -I \\ 0 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} -C_{t+1}\boldsymbol{\xi}_{t+1} \\ C_{t+1}\boldsymbol{\xi}_{t+1} \\ \bar{u}_{t+1} \\ -\underline{u}_{t+1} \\ \bar{x}_{t+1} \\ -\underline{x}_{t+1} \\ -G_{t+1}\boldsymbol{\xi}_{t+1} \end{bmatrix}.$$

Denoting by $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_7)$ the multiplier associated to the constraint of this problem, the Fenchel conjugate of $\mathcal{B}_t(V_{t+1}^*)$

$$\begin{aligned} \mathcal{B}_t^\dagger(V_{t+1}^*)(\lambda) &= \inf_{\boldsymbol{\mu}, \boldsymbol{\nu}} \mathbb{E} \left[-\boldsymbol{\mu}^\top \mathbf{H} + V_{t+1}^*(\boldsymbol{\nu}) \right], \\ \text{s.t. } & T^\top \mathbb{E}[\boldsymbol{\mu}] + \lambda = 0, \\ & \mathcal{W}_u^\dagger(\boldsymbol{\mu}) = b_{t+1}, \\ & \mathcal{W}_y^\dagger(\boldsymbol{\mu}) = \boldsymbol{\nu}, \\ & \boldsymbol{\mu} \leq 0, \end{aligned}$$

writes:

$$\begin{aligned} \mathcal{B}_t^\dagger(V_{t+1}^*)(\lambda) &= \inf_{\boldsymbol{\mu}, \boldsymbol{\nu}} \mathbb{E} \left[(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top C_{t+1} \boldsymbol{\xi}_{t+1} - \boldsymbol{\mu}_3^\top \bar{u}_{t+1} + \boldsymbol{\mu}_4^\top \underline{u}_{t+1} - \boldsymbol{\mu}_5^\top \bar{x}_{t+1} \right. \\ & \quad \left. + \boldsymbol{\mu}_6^\top \underline{x}_{t+1} + \boldsymbol{\mu}_7^\top G_{t+1} \boldsymbol{\xi}_{t+1} + V_{t+1}^*(\boldsymbol{\nu}) \right], \\ \text{s.t. } & A_t^\top \mathbb{E}[\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2] + D_t^\top \mathbb{E}[\boldsymbol{\mu}_7] = -\lambda, \\ & B_{t+1}^\top (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + (\boldsymbol{\mu}_3 - \boldsymbol{\mu}_4) + E_{t+1}^\top \boldsymbol{\mu}_7 = b_{t+1}, \\ & -(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + (\boldsymbol{\mu}_5 - \boldsymbol{\mu}_6) = \boldsymbol{\nu}, \\ & \boldsymbol{\mu}_1 \leq 0, \boldsymbol{\mu}_2 \leq 0, \boldsymbol{\mu}_3 \leq 0, \boldsymbol{\mu}_4 \leq 0, \boldsymbol{\mu}_5 \leq 0, \boldsymbol{\mu}_6 \leq 0, \boldsymbol{\mu}_7 \leq 0. \end{aligned}$$

We make the following assumption.

Assumption 32.

$$\forall \lambda \in \mathbb{R}^n, \exists \boldsymbol{\mu}_a \in \mathbb{R}^n, \exists \boldsymbol{\mu}_b \in \mathbb{R}^p, \boldsymbol{\mu}_b \leq 0 \text{ such that } A_t^\top \boldsymbol{\mu}_a + D_t^\top \boldsymbol{\mu}_b + \lambda = 0.$$

Remark 33. Such an assumption is for example fulfilled if A_t is a (square) full rank matrix. \diamond

Under Assumption 32, and with any arbitrary non positive random variables $\boldsymbol{\mu}_5$ and $\boldsymbol{\mu}_6$, the pair of random vectors $(\boldsymbol{\mu}, \boldsymbol{\nu})$ defined by

- $\boldsymbol{\mu} = \left((\mu_a)^-, -(\mu_a)^+, (b_{t+1} - B_{t+1}^\top \mu_a - E_{t+1}^\top \mu_b)^-, -(b_{t+1} - B_{t+1}^\top \mu_a - E_{t+1}^\top \mu_b)^+, \boldsymbol{\mu}_5, \boldsymbol{\mu}_6, \mu_b \right)$,
- $\boldsymbol{\nu} = -\mu_a + (\boldsymbol{\mu}_5 - \boldsymbol{\mu}_6)$,

satisfies the constraints of the optimization problem associated to the computation of $\mathcal{B}_t^\dagger(V_{t+1}^*)(\lambda)$. Such a pair $(\boldsymbol{\mu}, \boldsymbol{\nu})$ exists for all possible values of λ . Moreover, $\boldsymbol{\nu}$ linearly depends on the difference $\boldsymbol{\mu}_5 - \boldsymbol{\mu}_6$, $\boldsymbol{\mu}_5$ and $\boldsymbol{\mu}_6$ being any arbitrary negative random variables, so that $\boldsymbol{\nu}$ can take any possible value in \mathbb{R}^n .

We thus deduce that the domain of the dual constraint set valued mapping \mathcal{G}_t^\dagger , with

$$\mathcal{G}_t^\dagger(\lambda) = \{(\boldsymbol{\mu}, \boldsymbol{\nu}) \in \mathbb{R}^{n_x+n_c} \mid T^\top \mathbb{E}[\boldsymbol{\mu}] + \lambda = 0, \mathcal{W}_u^\dagger(\boldsymbol{\mu}) = \mathbf{C}, \mathcal{W}_y^\dagger(\boldsymbol{\mu}) = \boldsymbol{\nu}, \boldsymbol{\mu} \leq 0\},$$

is equal to the whole space \mathbb{R}^n , so that the sequence of dual linear Bellman operators $(\mathcal{B}_t^\dagger)_{t \in \llbracket 0, T-1 \rrbracket}$ is compatible.

References

- Regan Baucke, Anthony Downward, and Golbon Zakeri. A deterministic algorithm for solving multistage stochastic programming problems. *Optimization Online*, 2017.
- Heinz H Bauschke, Patrick L Combettes, et al. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 2011. Springer, 2017.
- Richard Bellman. *Dynamic Programming*. Princeton University Press, New Jersey, 1957.
- Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1 & 2. Athena Scientific Belmont, MA, 3rd edition, 2005.
- Jonathan Borwein and Adrian S Lewis. *Convex analysis and nonlinear optimization: theory and examples*. Springer, 2010.
- Iain Dunning, Joey Huchette, and Miles Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.
- Pierre Girardeau, Vincent Leclere, and Andrew B Philpott. On the convergence of decomposition methods for multistage stochastic convex programs. *Mathematics of Operations Research*, 40(1):130–145, 2014.
- Vincent Guigues. Convergence analysis of sampling-based decomposition methods for risk-averse multistage stochastic convex programs. *SIAM Journal on Optimization*, 26(4):2468–2494, 2016.
- Vincent Guigues. Dual dynamic programming with cut selection: Convergence proof and numerical experiments. *European Journal of Operational Research*, 258(1):47–57, 2017.
- Tito Homem-de Mello, Vitor L De Matos, and Erlon C Finardi. Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling. *Energy Systems*, 2(1):1–31, 2011.
- Philippe Mahey, Jonas Koko, and Arnaud Lenoir. Decomposition methods for a spatial model for long-term energy pricing problem. *Mathematical Methods of Operations Research*, 85(1):137–153, 2017.
- Mario VF Pereira and Leontina MVG Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, 52(1-3):359–375, 1991.
- Andrew Philpott and Ziming Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36(4):450–455, 2008.
- Andrew Philpott, Vitor de Matos, and Erlon Finardi. On solving multistage stochastic programs with coherent risk measures. *Operations Research*, 61(4):957–970, 2013.

- R Tyrrell Rockafellar. *Convex analysis*. Princeton University Press, 1970.
- R Tyrrell Rockafellar and Roger JB Wets. Stochastic convex programming: relatively complete recourse and induced feasibility. *SIAM Journal on Control and Optimization*, 14(3):574–589, 1976.
- Alexander Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, 2011.
- Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.
- Alexander Shapiro, Wajdi Tekaya, Joari P da Costa, and Murilo P Soares. Final report for technical cooperation between georgia institute of technology and ons–operador nacional do sistema elétrico. *Georgia Tech ISyE Report*, 2012.
- Wim Van Ackooij, Welington de Oliveira, and Yongjia Song. On regularization with normal solutions in decomposition methods for multistage stochastic programming. *Optimization Online*, 2017.