Journal of Convex Analysis
Volume 30 (2023), No. 3, 951–999

Trajectory Following Dynamic Programming Algorithms without Finite Support Assumptions

Maël Forcier, Vincent Leclère
CERMICS, Ecole des Ponts, Marne-la-Vallée, France
vincent.leclere@enpc.fr

Dedicated to Roger J-B Wets on the occasion of his 85th birthday.

Received: June 1, 2022
Accepted: April 21, 2023

We introduce a class of algorithms, called Trajectory Following Dynamic Programming (TFDP) algorithms, that iteratively refines approximations of cost-to-go functions of multistage stochastic problems with independent random variables. This framework encompasses most variants of the Stochastic Dual Dynamic Programming algorithm. Leveraging a Lipschitz assumption on the expected cost-to-go functions, we provide a new convergence and complexity proof that allows random variables with non-finitely supported distributions. In particular, this leads to new complexity results for numerous known algorithms. Further, we detail how TFDP algorithms can be implemented without the finite support assumption, either through approximations or exact computations.

Keywords: Multistage stochastic programming, SDDP, duality.

2010 Mathematics Subject Classification: 90C15, 90C25, 90C39, 90C06, 49M37, 93A15.

1. Introduction

Multistage stochastic programming problems (MSP) are optimization under uncertainty problems where decisions are taken sequentially during *stages*. Between stages, some part of the uncertainty is revealed. These problems have numerous applications, in, for example, finance, energy and supply chain (see e.g. [19, 26, 68] and references therein). Unfortunately, MSP problems are known to be $\sharp P$-hard ([33, 56, 60]) and numerically challenging especially when the number of stages grows.

More precisely, considering a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, we define a sequence of random variables, called *noises*, $(\boldsymbol{\xi}_t)_{t\in[T]}$, where $[T]$ stands for $\{1, \dots, T\}$ and T is the *horizon* of the problem. Assuming that each $\boldsymbol{\xi}_t$ has finite support of size n_ξ, an MSP problem admits an equivalent deterministic formulation with $O(n_\xi^T)$ variables. There are multiple algorithms (for a recent introduction to the topic we recommend [55]), each with various extensions and a rich literature, that exploit the special structure of the equivalent deterministic formulation, among which L-Shaped method [38, 66], and its extension to MSP i.e. nested Benders decomposition [10, 41], or progressive hedging algorithm [53].

However, each of these algorithms is numerically limited to a small horizon T. For a larger horizon, we need some additional assumptions on the noises. If they have

ISSN 0944-6532 / $ 2.50

limited memory (i.e. that $(\boldsymbol{\xi}_t, \boldsymbol{\xi}_{t+1}, \dots, \boldsymbol{\xi}_{t+\tau})$ is a Markov chain – for adequate indices) this opens the door to Dynamic Programming methods, among which the Stochastic Dual Dynamic Programming (SDDP) algorithm [46], and its variants (e.g. [1, 7, 50, 71]). All these algorithms compute a state trajectory and then follow it to update approximations of cost-to-go functions. We call them Trajectory Following Dynamic Programming (TFDP) algorithms.

1.1. Problem setting

We present here the general setting of multistage stochastic problems (MSP) we are considering in the paper. We also introduce three assumptions that are assumed to hold true throughout the paper.

All random variables (noises $\boldsymbol{\xi}_t$ or states $\boldsymbol{x}_t$) are assumed to be valued, for some adequate integer n_t, in $\mathbb{R}^{n_t}$ endowed with its Borel σ-algebra. To model the constraint of our stochastic problem we consider, for $t \in [T]$, the following Borel-measurable set-valued applications $\mathcal{X}_t : \mathbb{R}^{n_{t-1}} \times \Xi_t \rightrightarrows \mathbb{R}^{n_t}$ where $\Xi_t := \operatorname{supp}(\boldsymbol{\xi}_t) \subseteq \mathbb{R}^n$. We further assume, for simplicity, that the first noise is deterministic, that is $\Xi_1 = \{\xi_1\}$. For notational consistency we introduce $x_0 \in \mathbb{R}^{n_0}$ as a parameter, and $\boldsymbol{x}_0$ as the random variable with support $\{x_0\}$. We define recursively a sequence of *reachable sets* $X_t^r \subset \mathbb{R}^{n_t}$, for $t \in \{0, \dots, T\}$, by

$$X_0^r = \{x_0\}, \quad X_t^r = \bigcup_{x_{t-1} \in X_{t-1}^r} \bigcup_{\xi \in \Xi_t} \mathcal{X}_t(x_{t-1}, \xi) \quad \forall t \in [T]. \tag{1}$$

Finally, a sequence of *loss functions* $(\ell_t)_{t\in[T]}$, where $\ell_t : \mathbb{R}^{n_t} \times \Xi_t \to \mathbb{R} \cup \{+\infty\}$, is considered.

Assumption 1. (Compatibility of constraints) *We make the following assumptions, for all $t \in [T]$,*

(i) *ℓ_t is a proper normal integrand;*

(ii) *for all $x_t \in X_t^r$, the random variable $\ell_t(x_t, \boldsymbol{\xi}_t)$ is integrable (in particular $\ell_t(x_t, \boldsymbol{\xi}_t) < +\infty$ $\mathbb{P}$-almost surely);*

(iii) *for all $x_{t-1} \in X_{t-1}^r$ and almost all $\xi_t \in \Xi_{t-1}$, $\mathcal{X}_t(x_{t-1}, \xi_t)$ is a non-empty compact subset of $\mathbb{R}^{n_t}$.*

Finally, we say that $(\boldsymbol{x}_t)_{t\in[1:T]}$ is an *admissible policy* if it is a sequence of random variables such that, for all $t \in [T]$, $\boldsymbol{x}_t \in \mathcal{X}_t(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_t)$ $\mathbb{P}$-almost surely, and $\boldsymbol{x}_t$ is measurable with respect to $\sigma(\{\boldsymbol{\xi}_\tau\}_{\tau\in[t]})$. We denote X^{ad} the set of admissible policies. Then, the multistage stochastic problem (MSP) consists in minimizing over the set of admissible policies the sum of losses, that is

$$\text{(MSP)} \qquad \min_{\boldsymbol{x} \in X^{ad}} \quad \mathbb{E}\Big[\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t, \boldsymbol{\xi}_t)\Big]$$

Assumption 1 ensures that (MSP) is well-posed and admits an optimal solution. It also guarantees that we are in a *relatively complete recourse* setting in the sense that any sequence of variables $(\boldsymbol{x}_\tau)_{\tau\le t}$ satisfying $\boldsymbol{x}_\tau \in X_\tau(\boldsymbol{x}_{\tau-1}, \boldsymbol{\xi}_\tau)$, for $\tau \le t$ can be completed into an admissible policy $(\boldsymbol{x}_\tau)_{\tau\le T}$ such that $\mathbb{E}\big[\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t, \boldsymbol{\xi}_t)\big] < +\infty$.

As we are considering Dynamic Programming methods, the following *stagewise independence*, as well as exogenous noises, assumption is assumed to hold true.

Assumption 2. (Stagewise independence) *$(\boldsymbol{\xi}_t)_{t\in[T]}$ is a sequence of independent exogenous random variables. By exogenous we mean that the law of $\boldsymbol{\xi}_t$ is independent of all decision variables.*

Leveraging Assumption 2, we can rewrite Problem (MSP) in the following equivalent nested form (see e.g., [58, Chap. 3])

$$\min_{x_1\in\mathcal{X}_1(x_0,\xi_1)} \ell_1(x_1,\xi_1)+\mathbb{E}\Big[\min_{x_2\in\mathcal{X}_2(x_1,\boldsymbol{\xi}_2)} \ell_2(x_2,\boldsymbol{\xi}_2)+\mathbb{E}\big[\cdots+\mathbb{E}\big[\min_{x_T\in\mathcal{X}_T(x_{T-1},\boldsymbol{\xi}_T)} \ell_T(x_T,\boldsymbol{\xi}_T)\big]\big]\Big], \quad (2)$$

which can be tackled by Dynamic Programming. To this end, we introduce the following (backward) *Bellman operators*.

For a measurable proper l.s.c function $\tilde{V} : \mathbb{R}^{n_t} \to \mathbb{R}\cup+\infty$, we denote the Bellman operator of Problem (MSP) applied to $\tilde{V}$ by

$$\hat{\mathcal{B}}_t(\tilde{V}) = \begin{cases} \mathbb{R}^{n_t}\times\Xi_{t+1} & \to \mathbb{R}\cup\{+\infty\} \\ (x_t,\xi_{t+1}) & \mapsto \min\limits_{x_{t+1}\in\mathcal{X}_{t+1}(x_t,\xi_{t+1})} \ell_{t+1}(x_{t+1},\xi_{t+1})+\tilde{V}(x_{t+1}) \end{cases}. \quad (3a)$$

Further, note that for $\tilde{V}$ l.s.c. and finite valued on X_t^r, $\hat{\mathcal{B}}_t(\tilde{V})$ is also a normal integrand. We then define,

$$\mathcal{B}_t(\tilde{V}) : x_t \mapsto \mathbb{E}\big[\hat{\mathcal{B}}_t(\tilde{V})(x_t,\boldsymbol{\xi}_{t+1})\big]. \quad (3b)$$

With this notation we further define by induction the *expected cost-to-go* functions

$$V_t : \mathbb{R}^{n_{t-1}} \to \mathbb{R}, \quad V_T :\equiv 0, \quad V_t := \mathcal{B}_t(V_{t+1}) \quad \forall t\in\{0,...,T-1\}. \quad (4)$$

Finally, as the law of $\boldsymbol{\xi}_1$ is a Dirac distribution on ξ_1, the value of Problem (MSP) is simply $V_0(x_0)$, and any admissible policy $(\boldsymbol{x}_t)_{t\in[T]}$ minimizing the Bellman equation (4) for all $t\in[T]$, that is such that $V_{t-1}(\boldsymbol{x}_t) = \ell_t(\boldsymbol{x}_t,\boldsymbol{\xi}_t)+V_t(\boldsymbol{x}_t)$, is optimal.

Remark 1.1. (Stepwise control) For notational simplicity, we chose to consider loss function ℓ_t that only depends on the next state[1] x_t. However, it is worth keeping in mind that these loss functions are often defined as the partial minimum of another normal integrand, i.e.

$$\ell_t(x_t,\xi) = \inf_{y\in\mathbb{R}^m} \tilde{\ell}(x_t,y,\xi).$$

In theory, the same problem can be tackled by extending the state vector x to also contain the decisions y. However, this is misleading: the theoretical complexity is exponential in the dimension of x, which is in line with the curse of dimensionality of Dynamic Programming. Thus extending the state to include y falsely seems to imply an increase in the number of iterations required by trajectory following algorithms to converge. For example, in long term electricity management problems it is standard to have decisions y of dimension a few thousand (thermal generation, transmission on lines...) while the actual state x (hydroelectric storage) is of dimension a few dozen at most.

[1] Cost dependence on x_{t-1} is not considered here simply for notational convenience.

We end the presentation of our setting with a non-trivial assumption.

Assumption 3. (Lipschitz) *For $t \in [T]$, we assume that*

(i) *X_t^r has a diameter smaller than $D_t < +\infty$;*

(ii) *the expected cost-to-go function V_t is L_t-Lipschitz.*

Both parts of Assumption 3 are strong requirements, needed for the convergence results, while still being natural in most settings. Part (i) is satisfied for example if Assumption 1 holds, $\mathcal{X}_t(x_{t-1}, \cdot)$ is Lipschitz for all $x_{t-1} \in X_{t-1}^r$ and all Ξ_t are bounded. Part (ii) is satisfied under Assumption 1 in the linear case, or through an *extended relatively complete recourse* assumption (see [28]) which requires that states x_t that are slightly outside of X_t^r are still admissible. Note that we do not require the actual knowledge of D_t and L_t to implement the algorithm.

1.2. The SDDP algorithm

As a warm-up for the forthcoming framework and results we recall the well-known Stochastic Dual Dynamic Programming algorithm (SDDP). It was originally designed by Pereira and Pinto [46], based on the nested Benders decomposition method [10], for multistage stochastic linear problems (MSLP) with finitely supported noises. Note that, in this setting, the expected cost-to-go functions V_t, defined in (4), are polyhedral and in particular convex. As such, they can be approximated as maxima of k affine functions.

The crux of the algorithm is the following. At step k of the algorithm, we assume that we have, for each $t \in [T]$, a collection of k affine function f_t^κ, called *cuts*, such that $f_t^\kappa \leq V_t^k$. Then, in a forward pass, we randomly draw a scenario (ξ_t^k), and compute a trajectory $(x_t^k)_{t\in[T]}$ that is the best along this scenario according to current cost-to-go estimation, that is such that

$$\begin{aligned} x_{t+1}^k \in \operatorname*{arg\,min}_{x_{t+1},\theta} \quad & \ell_{t+1}(x_{t+1}, \xi_{t+1}) + \theta \\ \text{s.t.} \quad & x_{t+1} \in \mathcal{X}_{t+1}(x_t^k, \xi_{t+1}^k), \quad f_{t+1}^\kappa(x_{t+1}) \leq \theta \quad \forall \kappa \in [k]. \end{aligned}$$

Then, in a backward pass, we use linear programming (or, more generally, convex) duality theory to compute new cuts f_t^{k+1}. More precisely, define $\underline{V}_t^k := \max_{\kappa\leq k} f_t^\kappa$. Then, we set $V_{T+1}^{k+1} \equiv 0$, and loop backward in time from $t = T$ to $t = 0$. At time t, for all $\xi_{t+1} \in \operatorname{supp} \boldsymbol{\xi}_{t+1}$, we solve

$$\begin{aligned} &\hat{\mathcal{B}}_t(\underline{V}_t^{k+1})(x_t^k, \xi_{t+1}) = \min_{x_t, x_{t+1}, \theta} \ell_{t+1}(x_{t+1}, \xi_{t+1}) + \theta \\ &\text{s.t.} \quad x_{t+1} \in \mathcal{X}_{t+1}(x_t^k, \xi_{t+1}^k), \ f_{t+1}^\kappa(x_{t+1}) \leq \theta \ \ \forall \kappa \in [k+1], \quad x_t = x_t^k \ \ [\hat{\alpha}(\xi_{t+1})] \end{aligned}$$

Let $\hat{\theta}(\xi_{t+1}) = \hat{\mathcal{B}}_t(\underline{V}_t^{k+1})(x_{t+1}^k, \xi_{t+1})$ be the optimal value of the above problem, and $\hat{\alpha}(\xi_{t+1})$ be a subgradient of $\hat{\mathcal{B}}_t(\underline{V}_t^{k+1})(x_{t+1}^k, \xi_{t+1})$. Then, for all $\xi_{t+1} \in \operatorname{supp}(\boldsymbol{\xi}_{t+1})$, $\hat{\theta}(\xi_{t+1}) + \langle \hat{\alpha}(\xi_{t+1}), \cdot - x_t^k \rangle \leq \hat{\mathcal{B}}_t(\underline{V}_{t+1}^{k+1})(\cdot, \xi_{t+1})$. Taking the expectation, we can define the new cut

$$f_t^{k+1} = \mathbb{E}\big[\hat{\theta}(\boldsymbol{\xi}_{t+1})\big] + \langle \mathbb{E}\big[\hat{\alpha}(\boldsymbol{\xi}_{t+1})\big], \cdot - x_t^k \rangle \leq \mathcal{B}_t(\underline{V}_t^{k+1}) \leq \mathcal{B}_t(\underline{V}_{t+1}) = V_t.$$

This algorithm has been successfully applied to various applications, especially for energy management problems (see e.g., [13, 29, 42]). It has also given birth to numerous variants, with different types of node selection methods and/or cuts, that we gather here under the name Trajectory Following Dynamic Programming (TFDP) algorithm. Our aim is to provide a generic convergence theory for all these algorithms.

1.3. Review of known convergence results

The main idea of SDDP and its various extensions consists in iteratively refining lower (and sometimes upper) approximations of the expected cost-to-go functions V_t. More precisely, at each iteration, they decide, in a forward phase, based on a node selection method, trial points at which the approximations should be refined. Then, in a backward phase, they construct *cuts*, which are functions that under approximate V_t. These cuts are as close as possible to the true expected cost-to-go functions around the trial points. The lower approximations are finally defined as the maximum of computed cuts. This is detailed in Section 2.

To our knowledge, by contrast to our work, almost all prior works make the following assumption or consider an approximated problem that satisfies this assumption.

Assumption 1.2. (FSN – Finitely Supported Noise) *The support of the random process* $(\boldsymbol{\xi}_t)_{t\in[T]}$ *is finite.*

The first proven convergence result of the SDDP algorithm is due to Philpott and Guan [49]. In this paper, the authors consider the linear setting. Using Assumption (FSN) they prove that the number of (affine) cuts that can be generated is finite. Then, leveraging the fact that each scenario is sampled an infinite number of times, they prove the almost-sure convergence in a finite number of iterations, without any bound on this number. Later convergence results by [28] (then reformulated and adapted to the risk-averse setting in [30]) showed convergence in a non-linear, convex setting. Again, the proof argues that each scenario is selected randomly an infinite number of times. A technical lemma coupled with Borel-Cantelli's yields almost-sure asymptotic convergence.

Instead of random node selection, some deterministic node selection methods have been proposed. The problem-child algorithm [7], which maintains both an upper and a lower approximation, proved convergence by showing that the gap between upper and lower bounds is non-increasing with the iteration. This algorithm has been extended to convex-concave framework, using saddle-cuts [8], e.g. allowing for stagewise-dependent objective uncertainty in [16], or risk-averse problem [27]. In other cases, deterministic samplings are considered as a first step for proving the convergence of the randomized version [37, 49, 70].

The above papers all rely on affine, often called Benders', cuts. Some variants of SDDP, handled by our framework, use other types of cuts and also have proven asymptotic convergence. Zou et. al. presented a version of SDDP for binary variables, which has an asymptotic convergence proven in [71] for the convex case, although the proof can be directly adapted to the Lipschitz case. In addition to traditional Benders' cuts it relies on integer, Lagrangian and strengthened Benders'

cut, recalled in Appendix A. Stochastic Lipschitz Dynamic Programming (SLDP) by Ahmed *et. al.* [1] uses concave L_1 cuts instead of affine cuts for any Lipschitz V_t. MIDAS by Philpott et al. [50] uses step-function cuts for quasi-monotonous V_t and also fall in this category.

Further work has been dedicated to improving the numerical efficiency of the algorithm. Some methods alleviate the computational burden of each iteration, like Guigues in [31] which considers inexact cuts, or [6] which presents cut selection strategies, deleting some cuts from the representation of V_t. Other methods, like regularization approaches [5, 9, 65], try to reduce the number of required iterations. To our knowledge, if they sometimes preserve asymptotic convergence, none of these approaches provably reduce the number of iterations required to reach an ε-solution. These extensions are either handled by our framework or discussed in Subsection 2.2.

It is worth noting that the convergence arguments that rely on each possible scenario being sampled an infinite number of times are mainly theoretical: due to the sheer number of scenarios, in most applications, the algorithms sample only a very small subset of scenarios (and probably never twice the same).

In two recent papers ([37, 69]) new approaches were developed, focusing on the state space akin to the complexity proof of Kelley's cutting plane algorithm. They independently obtained the first explicit bound on the number of iterations required to obtain an ε-solution. To this end, they fix the error ε and define some saturated points in the state space. These points are such that the gap between the approximated value and the true value is controlled. Then, leveraging Lipschitz continuity of the value function they control the error in a ball around the saturated points. As the reachable sets are compact, only a finite number of such balls exist. They then each provide a deterministic algorithm with proven convergence and use it as a proxy to bound the expected number of iterations. Interestingly, the complexity of the deterministic algorithm is polynomial in the horizon T while the sampled algorithm complexity is exponential in T as it requires a given event to happen at each stage simultaneously.

All the convergence proofs recalled here rely on reachable set compactness, relatively complete recourse and finitely supported noise assumption. They then fall into two categories: either they directly use the Lipschitz continuity [2] of V_t, or argue that there exists only a finite number of possible cuts (e.g., [49, 71]). Our framework covers all these variants, but the convergence proof presented in Section 4, which is built on [37], does not require finitely supported noise Assumption (FSN). It is instead replaced, for the randomized algorithms, by a dedicated nested Hoeffding lemma (see Appendix D.1). This is another step toward understanding the practical convergence of these TFDP algorithms.

Further, without finitely supported noise Assumption (FSN), the standard approach consists in first discretizing the noise and then solving the discretized problem. A common method consists in sampling the problem through the Sample Average Approximation (SAA) approaches. The statistical guarantees of this approach are discussed by Shapiro in [57]. Other sampling strategies are numerically discussed in

[2] Actually MIDAS has a slightly milder requirement (see [50, Eq. (17)]).

[34, 40]. While never used, to our knowledge, in the context of TFDP algorithms, there are ways of discretizing the noise distribution in order to guarantee that the value of the discretized model under (or over) estimates the value of the true problem, especially in the convex setting, see [11, 36, 43, 44].

An alternative approach consists in finding a finitely supported, stagewise independent distribution that minimizes the nested-distance [47], to provide a good representation of the true problem. Other approaches exist, like [12, 25], which uses convexity tools and information relaxation to construct bounds. These approaches seem more relevant for problems with non-independent noises.

Finally, SDDP has been extended to various problem settings to handle risk aversion (e.g. [61, 27, 20]), infinite horizon (e.g. [59]), and partially observable problems [18]. We briefly discuss extensions to risk-averse settings in the last section; other extensions are outside the scope of this paper.

1.4. Contributions and structure of the paper

Our main contributions are the following:

- we provide a flexible framework (including inexact or regularized computations) for TFDP algorithms for finite horizon, risk-neutral problems, that encompass at least 14 variants of SDDP summed up in Table 1;
- we provide geometric tools to extend those algorithms to non-finitely supported uncertainties, without sampling or approximations in the linear case;
- we give a convergence speed result with an upper bound on the (expected) number of iterations to reach an ε-solution for these algorithms (which is new for most of those variants) that does not require the finite support assumption;
- we explain how to adapt those results to the minimax case. Some risk-averse or robust cases are seen as special cases.

The remainder of the paper is as follows. Section 2 introduces the general TFDP framework for risk neutral settings. It details further variants, not explicit in Algorithm 1, but to which the theoretical analysis of Section 4 can be easily adapted. Finally, it recalls classical ways of obtaining exact or approximated cuts. Section 3 provides a new method to leverage polyhedral geometry to implement an exact SDDP algorithm for non-finitely supported cost for multistage stochastic linear problems. Section 4 provides the main convergence and complexity results. Finally, Section 5 briefly reviews extensions to some robust and risk-averse settings. Technical proofs and definitions can be found in the appendices.

1.5. Notation

For ease of reference, we recall here some notational conventions used throughout the paper. $[n]$ is the set of integers between 1 and n. Bold font (e.g. $\boldsymbol{x}_t$) denotes a random variable, normal font of the same character (e.g. x_t) an element of its support. k and κ are indices of iteration, t and τ are indices of time-step. L represents a Lipschitz-constant, γ errors allowed in solving the subproblems, ε characterize the quality of the solution obtained. x denote a state, X a set of states (e.g. X_t^r is the set of reachable states) and $\mathcal{X}$ a set-valued application representing set of states

(e.g. $\mathcal{X}_t(x,\xi)$ is the set of admissible next states, $\mathcal{X}^\sharp_{\gamma,t}(\tilde{V})(x,\xi)$ is the set of γ-optimal next state...), ξ a noise, ℓ_t is the loss function at time t, V a cost-to-go function, f a cut. Overline is used for upper approximation, underline for lower approximation, hat (as in $\hat{\mathcal{B}}$) represents functions parametrized by ξ, and their counterpart without hat (as in $\mathcal{B}$) being their expectation. $\mathcal{B}$ (resp. $\mathcal{F}$) represent backward (resp. forward) Bellman operators. $\mathrm{ri}(X)$ is the relative interior of X.

2. Trajectory Following Dynamic Programming framework

Various extensions of the Stochastic Dual Dynamic Programming (SDDP) algorithm, recalled in Subsection 1.2, have been developed for different sets of assumptions. In this section, we first present, in Subsection 2.1, a generic algorithmic framework for TFDP algorithms (see Algorithm 1) for risk-neutral multistage stochastic programs that encompass multiple known algorithms (see Table 1). These algorithms consider under and upper approximation of the expected cost-to-go functions. The under approximations are defined as the maximum of basic functions called *cuts* (some classical cuts are recalled in Appendix A). The upper approximations are more diverse and not always computed. While Algorithm 1 is quite generic, other extensions have been proposed, and are discussed in Section 2.2. Finally, Subsection 2.3 present methods to compute cuts in simple settings: exact with finite support assumption and inexact with convexity assumption. Exact cut computation for the linear setting, without finite support, are discussed in the next Section 3. The remains of the section detail how to obtain cuts, in particular, Subsection 2.3.1 presents the finitely supported case, while Subsection 2.3.2 builds approximated cuts in the convex case.

2.1. Algorithm

The flexible framework of Algorithm 1 defines improving lower approximations $\underline{V}_t^k$ (resp. upper approximations[3] $\overline{V}_t^k$) of the expected cost-to-go functions V_t. Each iteration k of the algorithm consists in a forward phase to determine where to refine the approximations, followed by a backward phase to actually refine the approximations.

During the forward phase, we generate a trajectory $x_1^k, \cdots, x_{T-1}^k$. Each x_t^k is chosen as an (almost) optimal decision at time t starting from state x_{t-1}^k, knowing that the random variable $\boldsymbol{\xi}_t$ takes the value ξ_t^k, and considering that the cost-to-go is given by the lower approximation $\underline{V}_t^{k-1}$. This is encapsulated in the forthcoming notion of *forward Bellman operator*. We denote $\gamma\text{-}\arg\min_{x\in X} f(x)$ the set of $x \in X$ such that $f(x) \leq \inf_{x\in X} f(x) + \gamma$. We now define, for any Lipschitz (on X_t^r) function $\tilde{V}$, the set $\mathcal{X}^\sharp_{\gamma,t}(\tilde{V})$ of γ-optimal solutions of the parametrized stage t problem with cost-to-go function $\tilde{V}$, that is,

$$\mathcal{X}^\sharp_{\gamma,t}(\tilde{V}) : (x,\xi) \mapsto \gamma\text{-}\operatorname*{arg\,min}_{y\in\mathcal{X}_t(x,\xi)} \ell_t(y,\xi) + \tilde{V}(y).$$

Since $\ell_t + \tilde{V}$ is a normal integrand, [54, Corollary 14.33] guarantees measurability of $\mathcal{X}^\sharp_{\gamma,t}(\tilde{V})$. Further, as $\mathcal{X}^\sharp_{\gamma,t}(\tilde{V})$ is compact by Assumption 1, there exists ([54, Corollary 14.6]) a measurable selection of $\xi \mapsto \mathcal{X}^\sharp_{\gamma,t}(\tilde{V})(x,\xi)$ for all $x \in X_t^r$.

[3] In some common cases, the upper approximations are chosen as $\overline{V}_t^k = V_t$ but never evaluated.

Name of algorithm	Paper	Node selection: Choice $\boldsymbol{\xi}_t^k$	$\mathcal{F}_t$	$\underline{V}_t^k$	$\overline{V}_t^k$	Hypothesis	Complexity known
SDDP	[46]	Random sampling	Exact	Benders cuts	V_t	Convex	✓
EDDP	[37]	Explorative	Exact	Benders cuts	V_t	Convex	✓
APSDDP	[62]	Random sampling	Exact	Adaptive partition	V_t	Linear	×
SDDiP	[71]	Random sampling	Exact	Lagrangian or integer cuts	V_t	Mixed Integer Linear	×
MIDAS	[50]	Random sampling	Exact	Step cuts	V_t	Monotonic Mixed Integer	×
SLDP	[1]	Random sampling	Exact	Reverse norm cuts	V_t	Non-Convex	×
	[7]	Problem child	Exact	Benders cuts	Epigraph as convex hull	Convex	×
	[8]	Problem child	Exact	Benders × Epigraph	Hypograph × Benders	Convex-Concave	×
RDDP	[27]	Deterministic	Exact	Benders cuts	Epigraph as convex hull	Robust	×
ISDDP	[31]	Random sampling	Inexact	Inexact Lagrangian cuts	V_t	Convex	×
TDP	[2]	Problem child	Exact	Benders cuts	Min of quadratic	Convex	×
	[69]	Random or Problem	Regularized	Generalized conjugacy cuts	Norm cuts	Mixed Integer Convex	✓
NDDP	[70]	Random or Problem	Regularized	Benders cuts	Norm cuts	Distributionally Robust	✓
DSDDP	[39]	Random sampling	Exact	Benders cuts	Fenchel transform	Linear	×

Table 1: Synthesis of algorithms following the same framework

The following definition mathematically formalizes the selection choice [4].

Definition 2.1. (Forward operator) We say that $\mathcal{F}_t$ is a γ_{t+1}^F-*forward operator* if, for all functions $\tilde{V} : \mathbb{R}^{n_{t+1}} \to \mathbb{R} \cup \{+\infty\}$, Lipschitz on X_{t+1}^r and $x \in X_t^r$, $\mathcal{F}_t(\tilde{V})(x, \cdot)$ is a measurable selection of $\mathcal{X}^\sharp_{\gamma_{t+1}^F, t+1}(\tilde{V})(x, \cdot)$.

During the backward phase, we refine the approximations $\underline{V}_t^k$ and $\overline{V}_t^k$, they are both assumed to be Lipschitz on X_t^r. Further, the lower approximations $\underline{V}_t^k$ are defined as the maximum of a finite number of *cuts*: $\underline{V}_t^k = \max_{\kappa \leq k} f_t^k$.

For Algorithm 1, described on the next page, to converge we make the following assumption on the approximations computed.

Assumption 4. (Admissible approximations)
The computed cuts f_t^k of $\mathcal{B}_t(\underline{V}_{t+1}^k)$ at x_t^k satisfy:

(i) *f_t^k is $\underline{\gamma}_t$-tight, i.e. $f_t^k(x_t^k) \geq \mathcal{B}_t(\underline{V}_{t+1}^k)(x_t^k) - \underline{\gamma}_t$,*
(ii) *f_t^k is valid, i.e. $f_t^k \leq \mathcal{B}_t(\underline{V}_{t+1}^k)$,*
(iii) *$\underline{V}_t^k$ is $\underline{L}_t$-Lipschitz.*

On the other hand, the upper approximation $\overline{V}_t^k$, not necessarily computed, shall satisfy the following properties:

(iv) $\overline{V}_t^k(x_t^k) \leq \mathcal{B}_t(\overline{V}_{t+1}^k)(x_t^k) + \bar{\gamma}_t$, *(tightness)*
(v) $\overline{V}_t^k \geq \mathcal{B}_t(\overline{V}_{t+1}^k)$, *(validity)*
(vi) $\overline{V}_t^k \leq \overline{V}_t^{k-1}$, *(monotonicity)*
(vii) *$\overline{V}_t^k$ is $\bar{L}_t$-Lipschitz.*

For the algorithm to be well-defined we need to guarantee the existence of cuts and upper approximation satisfying the previous assumption, as formally assumed now:

Assumption 5. *For every $t \in [T]$ and $k \in \mathbb{N}^\star$, there exists at least one cut f_t^k of $\mathcal{B}_t(\underline{V}_{t+1}^k)$ satisfying Assumption 4.*

This assumption is for example ensured through relatively complete recourse in the linear setting [46], through extended relatively complete recourse in the convex setting [28], through relatively complete continuous recourse in the binary setting [71], and relatively complete recourse and Lipschitz assumption in the Lipschitz setting of [1].

Remark 2.2. (Asymmetry of upper and lower approximations) The framework is not symmetrical in its treatment of the upper and lower cost-to-go approximations. Indeed, line 6 should not be done with the upper approximations [5] as it would restrict the exploration of the state space. For example, assume that $\overline{V}_t$ are (slightly Lipschitz-regularized) indicator functions of a single point, then the forward phase

[4] This choice is comparable to selecting a stage solver which always returns the same solution among the set of optimal solutions.

[5] The upper approximations $(\overline{V}_t^k)_{t \in [T]}$ still provide an admissible policy through the forward Bellman operators which have interesting properties, see [39].

would always produce the same trajectory, and the upper bound would not be updated.

Further, multiple TFDP algorithms do not actually compute $\overline{V}_t$, simply setting it to the true expected cost-to-go V_t (for iterations bounds). □

Data: Random variables $\boldsymbol{\xi}_t$, cost function at each step ℓ_t, constraints set-valued function X_t, initial state x_0, γ_t^F-forward operators $\mathcal{F}_t$.

1 $\underline{V}_t^0 \equiv -\infty$ and $\overline{V}_t^0 \equiv +\infty$ for $t \in [T]$;
2 **for** $k \in \mathbb{N}$ **do**
  /* `Forward phase`
3 Set $x_0^k = x_0$;
4 **for** $t = 1 : T-1$ **do**
5 Choose $\xi_t^k \in \text{supp}(\boldsymbol{\xi}_t)$
6 Let $x_t^k = \mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \xi_t^k)$
7 **end**
  /* `Backward phase`
8 Set $\underline{V}_T^k \equiv \overline{V}_T^k \equiv 0$;
9 **for** $t = T-1 : -1 : 1$ **do**
10 Find a $\underline{L}_t$-Lipschitz on X_t^r, valid and $\underline{\gamma}$-tight cut f_t^k of $\mathcal{B}_t(\underline{V}_{t+1}^k)$ at x_t^k, i.e. such that $f_t^k(x_t^k) \geq \mathcal{B}_t(\underline{V}_{t+1}^k)(x_t^k) - \underline{\gamma}_t$ and $f_t^k \leq \mathcal{B}_t(\underline{V}_{t+1}^k)$
11 Set $\underline{V}_t^k = \max(\underline{V}_t^{k-1}, f_t^k)$;
12 Define $\overline{V}_t^k$ satisfying Assumption 4 (iv) and (vii)
13 **end**
14 **end**

Algorithm 1: A general framework for TFDP algorithms

We have a first monotonicity result.

Lemma 2.3. *Under Assumptions* 1 – 5, *for all* $k \in \mathbb{N}$, $t \in [T-1]$ *and* $x \in \mathbb{R}^{n_t}$ *we have*

$$\underline{V}_t^{k-1}(x) \leq \underline{V}_t^k(x) \leq \mathcal{B}_t(\underline{V}_{t+1}^k)(x) \leq V_t(x), \tag{5a}$$

$$V_t(x) \leq \mathcal{B}_t(\overline{V}_{t+1}^k)(x) \leq \overline{V}_t^k(x) \leq \overline{V}_t^{k-1}(x). \tag{5b}$$

In particular, the gap can only decrease

$$0 \leq \overline{V}_t^k(x) - \underline{V}_t^k(x) \leq \overline{V}_t^{k-1}(x) - \underline{V}_t^{k-1}(x). \tag{6}$$

Proof. Direct by double induction on t and k and monotonicity of the Bellman operator. □

Remark 2.4. (The standard SDDP algorithm) The most common TFDP algorithm is the stochastic dual dynamic programming (SDDP), recalled in Subsection 1.2.

In SDDP, the value of the noise ξ_t^k, chosen in line 5, is drawn randomly on $\text{supp}(\boldsymbol{\xi}_t)$ which is assumed to be finite. The lower approximations are defined as a maximum of affine cuts.

Under relatively complete recourse assumption, the cuts can be assumed to be $\underline{L}_t$-Lipschitz. Further, in this simple setting, all errors are null: $\underline{\gamma}_t = \bar{\gamma}_t = \gamma_t^F = 0$. Finally, no upper bounds are computed and the complexity results of Section 4 are obtained by taking $\overline{V}_t^k = V_t^k$. □

Algorithm 1 is a flexible framework, and some lines remain to be detailed, which we now discuss.

Node selection choice in line 5. Most TFDP algorithms choose ξ_t^k by drawing it randomly according to the law of the random variable $\boldsymbol{\xi}_t^k$. The forward phase can then be seen as a Monte Carlo method for finding a trajectory $x_1^k, \cdots, x_T^k$. Then, it is also possible to choose ξ_t^k thanks to quasi-Monte Carlo methods.

Another way of choosing ξ_t^k consists in picking the $\xi \in \text{supp}(\boldsymbol{\xi}_t)$ that maximizes a certain criterion. In [7], Baucke, Downward and Zakeri suggested to chose ξ_t^k such that x_t^k maximizes the gap between the upper and lower approximations, i.e., $\overline{V}_t^k(x_t^k) - \underline{V}_t^k(x_t^k)$. They called this choice of ξ_t^k, the *problem child* node selection. In [37], Lan presented the Explorative Dual Dynamic Programming algorithm, where ξ_t^k is chosen so that x_t^k is the most distinguishable point, i.e. such that x_t^k is far from the previous computed points, see Eq. (45b), we speak of *explorative* node selection.

The proofs of convergence are harder to derive when ξ_t^k is chosen randomly, and the best upper bound known on the number of iterations of these algorithms are exponential in the horizon T. In comparison, when ξ_t^k is chosen deterministically as the problem child or as the most distinguishable point, the number of iterations is bounded by a polynomial in T. However, random sampling is often more efficient in practice (and easier to implement). We discuss the complexity results in Section 4.

Forward operator choice in line 6. In most algorithms, we assume that $\gamma_t^F = 0$ for all $t \in [T-1]$, thus $\mathcal{F}_{t-1}(\tilde{V})(x, \cdot)$ is a measurable selection of

$$\arg\min_{y \in \mathcal{X}_t(x,\cdot)} \ell_t(y, \cdot) + \tilde{V}(y).$$

The use of inexact cuts has also been proposed in [31] to alleviate the computational burden of each iteration.

Further, there have been various propositions to regularize the SDDP algorithm, see [5, 32, 65]. They mostly boil down to choosing a different forward operator, e.g., by adding a regularization term, which can be seen as γ_t^F-forward operator with $\gamma_t^F > 0$. For example, one can choose $\mathcal{F}_{t-1}(\tilde{V})(x, \xi)$ as a proximity operator $\text{prox}_{\ell_t(\cdot,\xi)+\tilde{V}(\cdot),\alpha}(\check{y}) := \arg\min_{y \in \mathcal{X}_t(x,\xi)} \ell_t(y, \xi) + \tilde{V}(y) + \alpha\|y - \check{y}\|_2^2$. In that case, if $X_t(x, \xi)$ has a finite diameter D, for $y = \mathcal{F}_{t-1}(\tilde{V})(x, \xi)$, we have

$$\ell_t(y, \xi) + \tilde{V}(y) \leq \min_{y' \in \mathcal{X}_t(x,\xi)} \ell_t(y', \xi) + \tilde{V}(y') + \alpha D.$$

Then, $\mathcal{F}_t$ is an αD-forward operator.

Finally, it is important that the algorithm use a single γ_t^F-forward operator. Indeed, if the set of γ_t^F-optimal solutions $\mathcal{X}^\sharp_{\gamma_t^F,t}(\tilde{V})(x, \xi)$ is not reduced to a single point, the convergence results only hold for the points selected by the forward operator. This remark is not only theoretical but has implications in practice: to be safe one should use the same solver (and parameters) during the training phase and

exploitation phase of the algorithm. For example, consider a problem with two equivalent storage and that only one of them is required to provide an optimal solution. Consider two forward operators, the first one, $\mathcal{F}^1_{t-1}$, prefers using the first storage while the second, $\mathcal{F}^2_{t-1}$ prefers using the second storage. Now assume that the algorithm ran until convergence with $\mathcal{F}^1_{t-1}$ yielding the approximations $\underline{V}^\infty_t$. Then, $\underline{V}^\infty_t$ correctly evaluates the value of the first storage, but has no information on the second. Consequently, a trajectory given by $\mathcal{F}^2_{t-1}(\underline{V}^\infty_t)$ might be far from optimal. A discussion of this fact, and practical answers, can be found in [17, §2.7].

Cuts f^k_t choice in line 10. We need to compute cuts f^k_t to approximate $\mathcal{B}_t(\underline{V}^k_{t+1})$ in the neighborhood of x^k_{t-1}. Recall that in Eq. (3b), $\mathcal{B}_t$ is defined as an expectation of parametric Bellman operators $\mathcal{B}_t(\underline{V}^k_{t+1}) = \mathbb{E}\big[\hat{\mathcal{B}}_t(\underline{V}^k_{t+1})(\cdot, \boldsymbol{\xi}_t)\big]$ Eq. (3a). Then, we can compute the average cut f^k_t thanks to parametric cuts $\hat{f}^k_{t,\xi}$.

In the finitely supported case in Subsection 2.3.1, we show that we can compute the average cut f^k_t directly by taking $f^k_t = \mathbb{E}\big[\hat{f}^k_{t,\boldsymbol{\xi}}\big]$ whereas in the convex, non-finitely-supported case, we present in Subsection 2.3.2 methods to approximate $\mathbb{E}\big[f^k_{t,\boldsymbol{\xi}}\big]$.

Finally, exact methods for linear problems are developed in Section 3. Furthermore, depending on the problem structure, there exist several types of parametrized cuts $\hat{f}^k_{t,\xi}$ in the literature. We recall them in Appendix A.

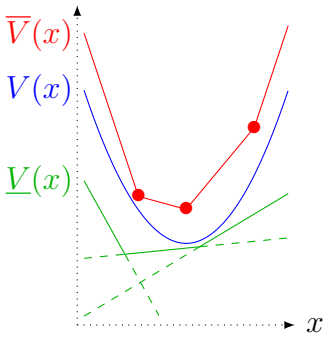


Figure 2.1: An example of upper and lower approximations

Upper approximations $\overline{V}^k_t$ choice in line 12. In most TFDP algorithms, no upper bound function is computed. In that case, we just set $\overline{V}^k_t \equiv V_t$ in the convergence proof. However, some algorithms rely on the computation of these upper bounds, for example for computing a problem-child node selection. In the convex case, assume that we have, for $t \in [T]$, some points $(x^\kappa_t, \bar{v}^\kappa_t)_{\kappa\in[k]}$ that are in the epigraph of V_t. Now define $\overline{V}^k_t$ such that

$$\operatorname{epi}(\overline{V}^k_t) = \operatorname{conv}\big((x^\kappa_t, \bar{v}^\kappa_t)_{\kappa\in[k]}\big) + \{(x,z) \in \mathbb{R}^{n_t} \times \mathbb{R} \mid \bar{L}_t\|x\|_1 \le z\} \subseteq \operatorname{epi}(V_t).$$

Then, $\overline{V}^k_t$ is an upper approximation $\overline{V}^k_t$ of V_t on X^r_t. Computing points $(x^\kappa_t, \bar{v}^\kappa_t)_{\kappa\in[k]}$ in the epigraph of $\hat{V}_t$ can be done either throughout the algorithm as in the problem-child approach [7], or in batch backward in time for a given set of trajectories as suggested by [48]. Upper approximation functions can also be obtained through duality see [14, 39].

2.2. Extensions of the TFDP framework

Although we tried to present a general framework, for the sake of simplicity, Algorithm 1 does not integrate every variant of SDDP. We show here how some variants could be integrated into the TFDP framework, and if the results and proofs of Section 4 remain valid.

Multiple forward phases. In practice, SDDP is often implemented with multiple forward phases, i.e., at iteration k we compute N forward phases $(x_t^{k,i})_{t\in[T-1],i\in[N]}$, in parallel. Consequently, in the backward phase we compute, for each time step $t \in [T-1]$, N tight and valid cuts $(f_t^{k,i})_{i\in[N]}$. This variation is included in the framework of Algorithm 1 by considering that the cut f_t^k is the maximum over $i \in [N]$ of all cuts $f_t^{k,i}$. The complexity results follow directly (in iteration number).

Multicut. In the finitely supported case, instead of computing an average cut f_t^k of the expected cost-to-go function V_t, it is possible to store for each $\xi \in \operatorname{supp}(\boldsymbol{\xi}_t)$ a cut $\hat{f}_{t,\xi}$ of the cost-to-go function $\hat{V}_t(\cdot,\xi)$. Unlike the single-cut case where we have $\underline{V}_t^k(\cdot) = \max_{\kappa\le k} f_t^\kappa(\cdot)$, in the multicut case, we compute the approximation function as $\underline{V}_t^k(\cdot) = \sum_{\xi\in\operatorname{supp}(\boldsymbol{\xi}_t)} \mathbb{P}\big[\xi\big] \max_{\kappa\le k} \hat{f}_{t,\xi}^\kappa(\cdot)$. Up to a slight reinterpretation, by considering a global cut $f_t(\cdot) = \sum_{\xi\in\operatorname{supp}(\boldsymbol{\xi}_t)} \mathbb{P}\big[\xi\big] \max_{\kappa\le k} \hat{f}_{t,\xi}(\cdot)$, this variation is covered by our framework.
However, with continuous random variables, the notion of multiple cuts is not well-defined.

Cut computation in forward. Another variation of SDDP consists in computing the cuts during the forward phase (and no backward phase). In this variant, the cuts do not approximate $\mathcal{B}_t(\underline{V}_{t+1}^k)$ and $\mathcal{B}_t(\overline{V}_{t+1}^k)$ in the neighborhood of x_t^k, but approximate $\mathcal{B}_t(\underline{V}_{t+1}^{k-1})$ and $\mathcal{B}_t(\overline{V}_{t+1}^{k-1})$ in the same neighborhood. Although this variant is not handled by the framework, all proofs can be adapted straightforwardly. More precisely, we only need to adapt the forthcoming proof of Lemma C.1. In particular, in the proof of Lemma C.1, we obtain directly Eq. (86c) and Eq. (87c), without using the monotonicity, because we approximate $\mathcal{B}_t(\underline{V}_{t+1}^{k-1})$ and $\mathcal{B}_t(\overline{V}_{t+1}^{k-1})$.

Cut selection. After many iterations, the number of cuts can slow down the new iterations. To speed up SDDP iterations, another idea is to delete some cuts. For example, we can decide to delete only the dominated cuts, i.e., the cuts that do not affect the values of the approximations $\underline{V}_t^k$. The monotonicity property and the complexity results are still valid in this setting. Unfortunately, finding which cut is dominated is time-consuming which can make this method numerically inefficient. Instead, we often use some heuristics to delete cuts that are probably dominated. However, these heuristics do not guarantee that we have the monotonicity property of approximations. Then, the complexity and convergence results seem harder to obtain. See [6] for an asymptotic convergence result on SDDP with cut selection.

Adaptive partition-based methods. In [63], Song and Luedtke presented the adaptive partition-based methods (APM) to solve 2-stage linear problems by partitioning the set of scenarios. It was then adapted to the multistage case in [62] where Siddig and Song proposed an adaptive partition-based SDDP, in both cases under

the finitely supported noise Assumption (FSN). The idea of APM is to replace the expected cost-go-function $V = \mathbb{E}\big[\hat{V}(\cdot, \boldsymbol{\xi})\big]$ by a partitioned expected cost-to-go function $V_{\mathcal{P}} = \sum_{P\in\mathcal{P}} \mathbb{P}\big[\boldsymbol{\xi} \in P\big]\hat{V}(\cdot, \mathbb{E}\big[\boldsymbol{\xi} \,|\, \boldsymbol{\xi} \in P\big])$ where $\mathcal{P}$ is a partition of the sample space Ξ. A partition $\mathcal{P}$ is said to be *tight at* $\check{x}$, if $V_{\mathcal{P}}(\check{x}) = V(\check{x})$, *valid* if $V_{\mathcal{P}}(x) \leq V(x)$ for all $x \in \mathbb{R}^{n_t}$ and *adapted to* $\check{x}$ if it is valid and tight at $\check{x}$. Then, when $\mathcal{P}$ is a partition adapted to $\check{x}$, we can see the partitioned expected cost-to-go function $V_{\mathcal{P}}$ as a valid and tight cut of V at $\check{x}$. Such cuts represent the tangent cone of $\mathrm{epi}(\mathcal{B}_t(\underline{V}^k_{t+1}))$ at x where Benders' cut represents a single tangent plane (see [24, §3.2]). APM methods were extended to general distribution in [51]. In [24], the authors provided a necessary and sufficient condition for a partition to be adapted (without Assumption (FSN)) as well as a geometric method to obtain a valid and adapted partition. In particular, APM SDDP algorithm of [62] is a TFDP algorithm falling in the framework of Algorithm 1. It can be adapted to the non-finitely supported case through the discussion in Section 3.

2.3. Computing cuts

We now focus on finding a cut in line 10 of Algorithm 1. More precisely, we want to approximate $\mathcal{B}_t(\underline{V}^k_{t+1})$ in the neighborhood of x^k_{t-1}. Recall that $\mathcal{B}_t$ is defined as an expectation of parametric Bellman operators $\hat{\mathcal{B}}_t$ (see Eq. (3)). We present two known cases: exact cuts under Assumption (FSN) and approximated cuts if the dependence in the noise is convex.

2.3.1. Cuts with finitely supported distribution

When the distribution of $\boldsymbol{\xi}_t$ is finitely supported, computing a cut of $\hat{\mathcal{B}}_t(\tilde{V})(\cdot, \xi)$ for each element $\xi \in \mathrm{supp}(\boldsymbol{\xi}_t)$ automatically yields a cut for $\mathcal{B}_t(\tilde{V})$.

Proposition 2.5. *Assume that $\boldsymbol{\xi}_t$ is finitely supported with $p_\xi := \mathbb{P}\big[\boldsymbol{\xi}_t = \xi\big]$, for all $\xi \in \mathrm{supp}(\boldsymbol{\xi}_t)$, then*

$$\mathcal{B}_t(\tilde{V})(x) = \sum_{\xi\in\mathrm{supp}(\boldsymbol{\xi}_t)} p_\xi \hat{\mathcal{B}}_t(\tilde{V})(x, \xi). \tag{7a}$$

For every $\xi \in \mathrm{supp}(\boldsymbol{\xi}_t)$, assume that $\hat{f}_\xi$ is a valid and $\underline{\hat{\gamma}}_{t,\xi}$-tight cut of function $\hat{\mathcal{B}}_t(\tilde{V})(\cdot, \xi)$ at $\check{x}$, then we have

$$f := \sum_{\xi\in\mathrm{supp}(\boldsymbol{\xi}_t)} p_\xi \hat{f}_\xi \;\text{ is a valid and } \underline{\gamma}_t\text{-tight cut of } \mathcal{B}_t(\tilde{V}) \text{ at } \check{x} \text{ with } \underline{\gamma}_t := \sum_{\xi\in\mathrm{supp}(\boldsymbol{\xi}_t)} p_\xi \underline{\hat{\gamma}}_{t,\xi}. \tag{7b}$$

In this finitely-supported distribution setting, it remains to find cuts of the function $\hat{\mathcal{B}}_t(\tilde{V})(\cdot, \xi)$. There exist several tight and valid cuts depending on the structure of ℓ_t and X_t. We present classical cuts of the literature in Appendix A where we detail under which conditions these cuts are tight and valid and show how to compute them.

2.3.2. Approximated cuts in the convex case

In this section, we now turn to obtaining approximated cuts leveraging convexity. We present a method based on the inequalities of Jensen and Edmundson-Madansky, adapting the results of Birge and Wets [11] to our setting, see also [35, 4.7].

We start by recalling two well-known useful convex inequalities.

Proposition 2.6. (Jensen's and Edmunson-Madansky inequalities) *Let $g : \mathbb{R}^l \mapsto \mathbb{R}$ be a convex function and $\boldsymbol{\xi}$ be a random variable. Assume that there exists a polytope*[6] *$\Xi \subset \mathbb{R}^l$ containing* $\operatorname{supp}(\boldsymbol{\xi})$.

For any $\xi \in \Xi$ we denote $S_\Xi(\xi)$ the set of barycentric coordinates of ξ, that is the set of coefficients $(\tilde{\sigma}_{\Xi,v})_{v\in\operatorname{Vert}\Xi}(\xi) \in [0,1]^{|\operatorname{Vert}\Xi|}$ such that $\xi = \sum_{v\in\operatorname{Vert}(\Xi)} \tilde{\sigma}_{\Xi,v}(\xi)v$ and $\sum_{v\in\operatorname{Vert}(\Xi)} \tilde{\sigma}_{\Xi,v}(\xi) = 1$. Let $\xi \mapsto (\sigma_{\Xi,v}(\xi))_{v\in\operatorname{Vert}\Xi}$ be any measurable selection[7] *of $S_\Xi(\xi)$. We have the following inequality:*

$$g\big(\mathbb{E}\big[\boldsymbol{\xi}\big]\big) \le \mathbb{E}\big[g(\boldsymbol{\xi})\big] \le \sum_{v\in\operatorname{Vert}(\Xi)} \mathbb{E}\big[\sigma_{\Xi,v}(\boldsymbol{\xi})\big]g(v). \tag{8}$$

Moreover, if g is Lipschitz with constant L and Ξ has diameter D, the gap is at most LD:

$$\sum_{v\in\operatorname{Vert}(\Xi)} \mathbb{E}\big[\sigma_{\Xi,v}(\boldsymbol{\xi})\big]g(v) \le g\big(\mathbb{E}\big[\boldsymbol{\xi}\big]\big) + LD. \tag{9}$$

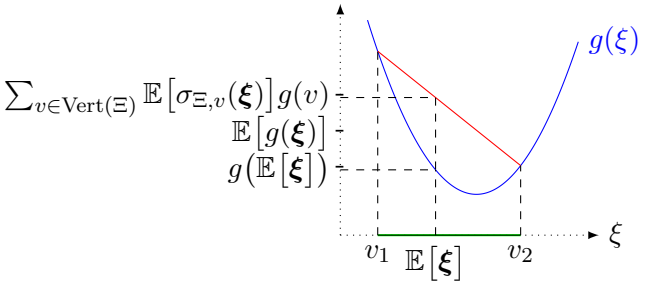


Figure 2.2: An illustration of Jensen and Edmundson-Madanski inequalities

Proof. The left-hand side of Eq. (8) is the classical Jensen inequality. Let $\xi \in \Xi$, as $(\sigma_{\Xi,v})_{v\in\operatorname{Vert}\Xi}(\xi)$ are barycentric coordinates, we have, due to the convexity of g, $g(\xi) \le \sum_{v\in\operatorname{Vert}(\Xi)} \sigma_{\Xi,v}(\xi)g(v)$. Taking the expectation leads to the right-hand side of Eq. (8) called Edmundson-Madanski inequality.

Assume now that Ξ has diameter D. Since Ξ is convex, $\mathbb{E}\big[\boldsymbol{\xi}\big] \in \Xi$, we conclude that for all $v \in \operatorname{Vert}\Xi$ we have $\|\mathbb{E}\big[\boldsymbol{\xi}\big] - v\| \le D$. Further, as g is Lipschitz, we obtain $\|g(v) - g(\mathbb{E}\big[\boldsymbol{\xi}\big])\| \le LD$. Taking the convex combination yields Eq. (9). □

These inequalities can be refined. Let $\mathcal{P}$ be a finite collection of almost surely disjoint polyhedra covering $\operatorname{supp}(\boldsymbol{\xi})$, i.e. $\operatorname{supp}(\boldsymbol{\xi}) \subset \cup_{P\in\mathcal{P}} P$ and $\mathbb{P}\big[P \cap P'\big] = 0$ if $P \ne P' \subset \mathcal{P}$.

Then, by the law of total expectation, $\mathbb{E}\big[g(\boldsymbol{\xi})\big] = \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big]\mathbb{E}\big[g(\boldsymbol{\xi})|P\big]$. Applying Jensen and Edmundson inequalities to each term of this sum, we get

$$\sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big]g(\mathbb{E}\big[\boldsymbol{\xi}|P\big]) \le \mathbb{E}\big[g(\boldsymbol{\xi})\big] \le \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big] \sum_{v\in\operatorname{Vert}(P)} \mathbb{E}\big[\sigma_{P,v}(\boldsymbol{\xi})\big]g(v). \tag{10}$$

[6] The results can be extended to the case where Ξ is an unbounded polyhedron. We must then consider a set $\operatorname{Ray}(\Xi)$ of extreme rays of the recession cone of Ξ (see [21])

[7] Such a selection always exists. Indeed, if Ξ is a simplex, barycentric coordinates are uniquely defined through a linear application. Then, any triangulation of Ξ defines a measurable selection as piecewise linear applications.

In particular, if all polyhedra $P \in \mathcal{P}$ has a diameter smaller than d, the gap can be bounded by Ld.

We now get back to the problem of line 10 of Algorithm 1 where we want to approximate $\mathcal{B}_t(\underline{V}^k_{t+1})$ in the neighborhood of x^k_{t-1}. Recall that $\mathcal{B}_t$ is defined as an expectation of parametric Bellman operators $\hat{\mathcal{B}}_t$ (see Eq. (3)). Unlike in Subsection 2.3.1 where the random variable where finitely supported, we cannot write the expected cut as a finite sum of parametric cuts. However, the Jensen and Edmundson-Madansky inequalities allow us to derive approximate cuts and upper-bound functions.

Proposition 2.7. *Assume that ℓ_t is a jointly convex function with Lipschitz constant L. Let $\mathcal{P}$ be a finite collection of almost surely disjoint polyhedra covering* $\mathrm{supp}(\boldsymbol{\xi})$*, such that any $P \in \mathcal{P}$ has a diameter smaller than $d \in \mathbb{R}_+$. Denote for each $P \in \mathcal{P}$, $p_P := \mathbb{P}\big[P\big]$ and $\xi_P := \mathbb{E}\big[\boldsymbol{\xi} | P\big]$.*
For every $P \in \mathcal{P}$, assume that $\underline{f}_P$ is a valid and $\underline{\gamma}_{t,P}$-tight cut of the parametric function $\hat{\mathcal{B}}_t(\tilde{V})(\cdot, \xi_P)$ at $\check{x}$, then by defining $\underline{\gamma}_t := Ld + \sum_{P\in\mathcal{P}} p_P \underline{\gamma}_{t,P}$, we have

$$\underline{f} := \sum_{P\in\mathcal{P}} p_P \underline{f}_P \text{ is a valid and } \underline{\gamma}_t\text{-tight cut of } \mathcal{B}_t(\tilde{V}) \text{ at } \check{x}. \tag{11}$$

For every $P \in \mathcal{P}$ and $v \in \operatorname{Vert} P$, assume that $\bar{f}_{v,P}$ satisfies $\bar{f} \geq \hat{\mathcal{B}}_t(\tilde{V})(\cdot, v)$ and $\bar{f}(\check{x}) \leq \hat{\mathcal{B}}_t(\tilde{V})(\check{x}, v) + \bar{\gamma}_{t,P}$. Then by defining

$$\bar{\gamma}_t := Ld + \sum_{P\in\mathcal{P}} p_P \sum_{v\in\mathrm{Vert}(P)} \mathbb{E}\big[\sigma_{P,v}(\boldsymbol{\xi})\big] \bar{\gamma}_{t,P},$$

we obtain that

$$\bar{f} := \sum_{P\in\mathcal{P}} p_P \sum_{v\in\mathrm{Vert}(P)} \mathbb{E}\big[\sigma_{P,v}(\boldsymbol{\xi})\big] \bar{f}_{P,v}$$

satisfies

$$\bar{f} \geq \mathcal{B}_t(\tilde{V}) \quad \text{and} \quad \bar{f}(\check{x}) \leq \mathcal{B}_t(\tilde{V})(\check{x}) + \bar{\gamma}_t. \tag{12}$$

This result can also be adapted to "saddle" cost functions, i.e. functions that are convex in some coordinates of ξ and concave in the other coordinates of ξ, by using both inequalities according to the sign of convexity, see e.g. [36, §4].

3. Exact SDDP in the linear case with generic distributions

Note that Algorithm 1 does not require the finite support assumption (FSN) as it directly requires the computation of cuts for the expected cost-to-go functions. These cuts are often given as expectation of easier-to-construct pointwise cuts, as done in Subsection 2.3. In this section, we present new geometrical tools to compute exact cuts without the finite support assumption for the cost, in the linear case. More precisely, we consider the particular case of multistage linear stochastic programming i.e., Problem (MSP) where, for all $t \in [T]$, $\boldsymbol{\xi}_t = (\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t, \boldsymbol{c}_t)$, $\ell_t(x_t, \xi_t) = c_t^\top x_t$ is linear and $\mathcal{X}_t(x_{t-1}, \xi_t) = \{x_t \,|\, A_t x_t + B_t x_{t-1} = b_t,\ x_t \geq 0\}$ is a polyhedron. In particular, we show that we can discretize our problem without getting an approximation error. To do so we leverage the polyhedral geometry for linear stochastic problem tools developed in [22, 23, 24]. For the sake of completeness, an introduction to those tools can be found in Appendix B.1 (see [15] for a more complete reference).

We start by reformulating the stage problem (3a) as a standard two-stage linear program in Subsection 3.1. Subsection 3.2 formally defines the notion of adapted partition that allows computing cuts. Subsection 3.3 leverages the geometric tools to construct an explicit adapted partition for the multistage linear setting.

3.1. Expected cost-to-go function in standard form

We make the following assumptions:

Assumption LS. (Linear setting) *For $t \in [T]$ we have $\ell_t(x_t, \xi_t) = c_t^\top x_t$ and $\mathcal{X}_t(x_{t-1}, \xi_t) = \{x_t \in \mathbb{R}^{n_t} \mid A_t x_t + B_t x_{t-1} = b_t,\ x_t \geq 0\}$. Further, the random variable $\boldsymbol{\xi}_t = (\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t, \boldsymbol{c}_t)$ and the approximated expected cost-to-go functions $\underline{V}_t^k$ satisfy*

(i) *$\boldsymbol{A}_t$ has a finitely supported distribution;*

(ii) *$\boldsymbol{c}_t$ and $(\boldsymbol{B}_t, \boldsymbol{b}_t)$ are independent*[8]*;*

(iii) *the lower expected cost-to-go function $\underline{V}_t^k$ are defined as the maximum of affine cuts, i.e., we have $(\alpha_t^l)_{l \leq k}$ and $(\beta_t^l)_{l \leq k}$ such that*

$$\underline{V}_t^k(x_t) = \max_{l \leq k} \ \alpha_t^{l\top} x_t + \beta_t^l. \tag{13}$$

Under Assumption (LS), the Bellman operator defined in (3a) applied to $\underline{V}_t^k$ reads

$$\hat{\mathcal{B}}_{t-1}(\underline{V}_t^k)(x_{t-1}, \xi_t) \tag{14a}$$

$$= \min_{x_t, z} \ c_t^\top x_t + z \qquad = \min_{x_t, z^+, z^-, r} \ c_t^\top x_t + z^+ - z^- \tag{14b}$$

$$\text{s.t.} \quad A_t x_t + B_t x_{t-1} = b_t, \qquad \text{s.t.} \quad A_t x_t + B_t x_{t-1} = b_t, \tag{14c}$$

$$\alpha_t^{\kappa\top} x_t + \beta_t^\kappa \leq z, \ \forall \kappa \leq k \qquad \alpha_t^{\kappa\top} x_t + \beta_t^l + r = z^+ - z^-, \ \forall \kappa \leq k$$

$$x_t \geq 0 \qquad x_t, z^+, z^-, r \geq 0. \tag{14d}$$

To simplify notation, we define

$$Q(x, W, q, T, h) := \min_y \left\{ q^\top y \mid Tx + Wy = h, y \geq 0 \right\}. \tag{15}$$

Then, for any $t \in [T]$ and $k \in \mathbb{N}$, setting $x := x_t$, $y := (x_t, z^+, z^-, r)$,

$$\boldsymbol{W} := \begin{pmatrix} \boldsymbol{A}_t & 0 & 0 & 0 \\ \alpha_t^1 & -1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_t^k & -1 & 1 & 1 \end{pmatrix}, \ \boldsymbol{q} := \begin{pmatrix} \boldsymbol{c}_t \\ 1 \\ -1 \\ 0 \end{pmatrix}, \ \boldsymbol{T} := \begin{pmatrix} \boldsymbol{B}_t \\ 0 \\ \vdots \\ 0 \end{pmatrix} \text{ and } \boldsymbol{h} := \begin{pmatrix} \boldsymbol{b}_t \\ -\beta_t^1 \\ \vdots \\ -\beta_t^k \end{pmatrix}, \tag{16}$$

$$\text{we obtain} \qquad \mathbb{E}\big[Q(x, \boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h})\big] = \mathbb{E}\big[\hat{\mathcal{B}}_{t-1}(\underline{V}_t^k)(x_{t-1}, \boldsymbol{\xi}_t)\big]. \tag{17}$$

Under Assumption (LS) we have that (i) $\boldsymbol{W}$ is finitely supported, and (ii) $\boldsymbol{q}$ and $(\boldsymbol{T}, \boldsymbol{h})$ are independent.

[8] Independence can be replaced by finite support assumption on one of the random variables. More generally, we can consider a finitely supported random variable $\boldsymbol{M}_t$ such that $\boldsymbol{c}_t$ and $(\boldsymbol{B}_t, \boldsymbol{b}_t)$ are independent *conditionally* to $\boldsymbol{M}_t$.

3.2. Partition and cuts

Let $n, m, p \in \mathbb{N}^\star$ be integers and $\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}$ be integrable random variables taking values respectively in $\mathbb{R}^{p\times m}$, $\mathbb{R}^m$, $\mathbb{R}^{p\times n}$, $\mathbb{R}^p$. We define the uncertainty set Ξ as the support of $(\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h})$. We make the following assumption:

Assumption FSW. (Finitely supported W)
The random variables $(\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h})$ satisfy:

(i) $\boldsymbol{W}$ *is finitely supported,*

(ii) $\boldsymbol{q}$ *and* $(\boldsymbol{T}, \boldsymbol{h})$ *are independent.*

Note that if the random variables are defined thanks to (16) then Assumption (LS) implies Assumption (FSW).

To alleviate notation, for a measurable set $P \subset \Xi$, we shorten $\mathbb{P}\big[(\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \in P\big]$ as $\mathbb{P}\big[P\big]$, and similarly for conditional expectation.

Definition 3.1. We define the value function $V : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$

$$V(x) := \mathbb{E}\big[Q(x, \boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h})\big]. \tag{18}$$

Let $\mathcal{P}$ be a partition of Ξ. Define the partitioned value function $V_{\mathcal{P}} : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$

$$\text{as} \qquad V_{\mathcal{P}}(x) := \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big] Q\big(x, \mathbb{E}\big[\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h} | P\big]\big), \tag{19}$$

where Q is defined in Eq. (15), and we made a slight abuse of notation with,

$$Q\big(x, \mathbb{E}\big[\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h} | P\big]\big) = Q\big(x, \mathbb{E}\big[\boldsymbol{W} | P\big], \mathbb{E}\big[\boldsymbol{q} | P\big], \mathbb{E}\big[\boldsymbol{T} | P\big], \mathbb{E}\big[\boldsymbol{h} | P\big]\big). \tag{20}$$

We say that

$$\mathcal{P} \text{ is valid if} \qquad V_{\mathcal{P}}(x) \le V(x) \qquad \forall x \in \mathbb{R}^n \tag{21a}$$

$$\mathcal{P} \text{ is tight at } \check{x} \text{ if} \qquad V_{\mathcal{P}}(\check{x}) = V(\check{x}) \tag{21b}$$

$$\mathcal{P} \text{ is adapted to } \check{x} \text{ if} \qquad \mathcal{P} \text{ is valid and tight at } \check{x} \tag{21c}$$

For any partition $\mathcal{P}$, if we know how to compute probabilities and conditional expectations on each element of the partition, we know by Subsection 2.3.1 how to compute a tight and valid cut for $V_{\mathcal{P}}$. If $\mathcal{P}$ is a valid and adapted partition, then the cut is still valid for the true value function V.

Proposition 3.2. *Let $\mathcal{P}$ be an adapted partition to $\check{x}$. For all $P \in \mathcal{P}$, let $\hat{f}_P$ be a valid and tight cut of $Q\big(\cdot, \mathbb{E}\big[\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h} | P\big]\big)$ at $\check{x}$. Then $f := \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big] \hat{f}_P$ is a valid and tight cut of V at $\check{x}$.*

Proof. We have

$$f(x) = \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big] \hat{f}_P(x), \tag{22a}$$

$$\le \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big] Q\big(x, \mathbb{E}\big[\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h} | P\big]\big), \qquad \text{since } \hat{f}_P \text{ is valid,} \tag{22b}$$

$$= V_{\mathcal{P}}(x) \le V(x), \qquad \text{since } \mathcal{P} \text{ is valid.} \tag{22c}$$

Thus, f is valid.

$$\text{Moreover,} \qquad f(\check{x}) = \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big]\, \hat{f}_P(\check{x}), \tag{23a}$$

$$= \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big]\, Q\big(\check{x}, \mathbb{E}\big[\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h} | P\big]\big), \qquad \text{since } \hat{f}_P \text{ is tight,} \tag{23b}$$

$$= V_{\mathcal{P}}(\check{x}) = V(\check{x}), \qquad \text{since } \mathcal{P} \text{ is tight.} \tag{23c}$$

Thus, f is tight. □

Thus, an adapted partition provides a valid and tight cut. We now detail how to obtain such a partition.

3.3. Explicit valid and adapted partition

In this section, we provide an explicit adapted partition for linear stochastic problems in standard form.

By Assumption 1, the feasible primal set $\{y \in \mathbb{R}^m \,|\, Tx + Wy = h, y \geq 0\}$ is nonempty and compact. Then, by strong duality, we can rewrite Q defined in Eq. (15)

$$\text{as} \qquad Q(x, W, q, T, h) = \max_{\lambda\in\mathbb{R}^p} \big\{(h - Tx)^\top \lambda \quad | \quad W^\top \lambda \leq q\big\}. \tag{24}$$

We now define the dual feasible set $D_{W,q}$ as

$$D_{W,q} := \{\lambda \in \mathbb{R}^p \mid W^\top \lambda \leq q\}. \tag{25}$$

We recall that the normal fan $\mathcal{N}(E)$ of a set E is the collection of all normal cones $N_E(x)$:

$$\mathcal{N}(E) := \{N_E(x) \mid x \in E\}, \tag{26a}$$

$$\text{where} \qquad N_E(x) := \{\phi \mid \forall x' \in E, \phi^\top(x' - x) \leq 0\}. \tag{26b}$$

When W and q are fixed, the value of $Q(\check{x}, W, q, T, h)$ depends on which normal cone $h - T\check{x}$ belongs to. Thus, we finally define

$$E_{N,\check{x}} := \{(T, h) \mid h - T\check{x} \in \mathrm{ri}(N)\} \tag{27a}$$

$$\mathcal{R}_{\check{x},W,q}, := \big\{E_{N,\check{x}} \mid N \in \mathcal{N}(D_{W,q})\big\}. \tag{27b}$$

We begin with the finitely supported $\boldsymbol{q}$ case as a warm-up.

Remark 3.3. (Finitely supported $\boldsymbol{q}$) We can show that, when W and q are fixed, $\mathcal{R}_{\check{x},W,q}$ is an adapted partition to $\check{x}$ (see [24, Thm. 3]). If $\mathrm{supp}(\boldsymbol{W})$ and $\mathrm{supp}(\boldsymbol{q})$ are finite, we can extend this result to show that

$$\big\{\{W\} \times \{q\} \times \mathcal{R}_{\check{x},W,q} \,|\, W \in \mathrm{supp}(\boldsymbol{W}),\, q \in \mathrm{supp}(\boldsymbol{q})\big\}$$

is an adapted partition to $\check{x}$:

$$\mathbb{E}\big[Q(\check{x}, \boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h})\big] \tag{28a}$$

$$= \sum_{(W,q)\in\mathrm{supp}(\boldsymbol{W},\boldsymbol{q})} \mathbb{E}\Big[Q(\check{x}, W, q, \boldsymbol{T}, \boldsymbol{h}) \;\Big|\; \boldsymbol{W} = W, \boldsymbol{q} = q\Big]\mathbb{P}\big[\boldsymbol{W} = W, \boldsymbol{q} = q\big], \tag{28b}$$

$$= \sum_{(W,q)\in\mathrm{supp}(\boldsymbol{W},\boldsymbol{q})} \sum_{R\in\mathcal{R}_{\check{x},W,q}} Q\Big(\check{x}, \mathbb{E}\Big[\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h} \;\Big|\; (\boldsymbol{T}, \boldsymbol{h}) \in R, \boldsymbol{W} = W, \boldsymbol{q} = q\Big]\Big) \cdot \mathbb{P}\big[(\boldsymbol{T}, \boldsymbol{h}) \in R, \boldsymbol{W} = W, \boldsymbol{q} = q\big]. \tag{28c}$$

We now extend this result to the case where $\boldsymbol{q}$ has a non-finitely supported distribution. This extension relies on a partition $\mathcal{S}_W$ of $\mathbb{R}^{p\times m}$ such that $q \mapsto \mathcal{R}_{\check{x},W,q}$ is constant on each $S \in \mathcal{S}_W$. Actually, this partition $\mathcal{S}_W$ is the collection of relative interiors of secondary cones (see [15, Definition 5.2.1]) which are classical objects from polyhedral geometry. We also give a more elementary, but equivalent, construction of $\mathcal{S}_W$ in Appendix B.1.

To any constraint matrix W, we associate its secondary fan Σ-fan(W) (which is the normal fan of the secondary polytope defined, e.g., in [15, Definition 5.1.7]), a well-studied collection of closed cones associated with W. Let $\mathcal{S}_W$ be the collection of relative interiors of the elements of Σ-fan(W):

$$\mathcal{S}_W := \big\{ \mathrm{ri}(S) \mid S \in \Sigma\text{-fan}(W) \big\}. \tag{29}$$

In particular; the elements of $\mathcal{S}_W$ are relatively open (convex) cones of $\mathbb{R}^m$. Further, note that [15] provides constructive representation of Σ-fan(W) and thus of $\mathcal{S}_W$, which paves the way toward explicit computation of $\mathcal{S}_W$.

Lemma 3.4. *Let $W \in \mathbb{R}^{p\times m}$ and $S \in \mathcal{S}_W$. For every $q, q' \in S$ we have $\mathcal{R}_{x,W,q} = \mathcal{R}_{x,W,q'}$. Consequently, instead of considering an infinite number of $\mathcal{R}_{x,W,q}$ parametrized by q, we can consider a finite number of $\mathcal{R}_{x,W,S}$ parametrized by $S \in \mathcal{S}_W$ where*

$$\mathcal{R}_{x,W,S} := \mathcal{R}_{x,W,q} \text{ for an arbitrary } q \in S. \tag{30}$$

Proof. The proof is given in Appendix B.1. □

We now leverage this reduction to a finite number of $\mathcal{R}_{x,W,S}$ to define an adapted partition. We start by showing in Theorem 3.5 that, for given x and W, the cost-to-go function $Q(x, W, q, T, h)$ is piecewise bilinear in q and (T, h). More precisely, it is an adaptation of the *basis decomposition theorem* of Walkup and Wets [67] (see also Sturmfels and Thomas [64] or [15, Theorem 1.2.2] for a more modern presentation).

Theorem 3.5. *Let $W \in \mathbb{R}^{l\times m}$ and $\check{x} \in \mathbb{R}^n$. Then, for every $S \in \mathcal{S}_W$ and $R \in \mathcal{R}_{\check{x},W,S}$, there exists a basis $B \subset [m]$ such that*

$$\forall q \in S, \quad \forall (T, h) \in R, \qquad Q(\check{x}, W, q, T, h) = q_B^\top W_B^{-1}(h - T\check{x}). \tag{31}$$

Proof. The proof is given in Appendix B.2. □

We deduce the following lemma which gives regions where we can interchange the function Q with the expectation. This lemma can be seen as an exact quantization result.

Lemma 3.6. *Let $W \in \mathbb{R}^{l\times m}$. Assume that $(\boldsymbol{T}, \boldsymbol{h})$ and $\boldsymbol{q}$ are* independent *random variables, then, for all $S \in \mathcal{S}_W$ and $R \in \mathcal{R}_{\check{x},W,S}$,*

$$\begin{aligned} &Q\Big(\check{x}, W, \mathbb{E}\big[\boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h} \mid \boldsymbol{q} \in S, (\boldsymbol{T}, \boldsymbol{h}) \in R\big]\Big) \\ &\qquad = \mathbb{E}\Big[Q(\check{x}, W, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \mid \boldsymbol{q} \in S, (\boldsymbol{T}, \boldsymbol{h}) \in R\Big]. \end{aligned} \tag{32}$$

Proof. Let B be the basis in Theorem 3.5. By the independence of $(\boldsymbol{T}, \boldsymbol{h})$ and $\boldsymbol{q}$, we have

$$\begin{aligned}
&\mathbb{E}\big[Q(\check{x}, W, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \mid \boldsymbol{q} \in S, (\boldsymbol{T}, \boldsymbol{h}) \in R\big] \\
&\qquad = \mathbb{E}\big[\boldsymbol{q}_B^\top W_B^{-1}(\boldsymbol{h} - \boldsymbol{T}\check{x}) \mid \boldsymbol{q} \in S, (\boldsymbol{T}, \boldsymbol{h}) \in R\big], && \text{(33a)} \\
&\qquad = \mathbb{E}\big[\boldsymbol{q}_B^\top \mid \boldsymbol{q} \in S\big] W_B^{-1} \mathbb{E}\big[\boldsymbol{h} - \boldsymbol{T}\check{x} \mid (\boldsymbol{T}, \boldsymbol{h}) \in R\big], && \text{(33b)} \\
&\qquad = Q\big(\check{x}, W, \mathbb{E}\big[\boldsymbol{q} \mid \boldsymbol{q} \in S\big], \mathbb{E}\big[\boldsymbol{T}, \boldsymbol{h} \mid (\boldsymbol{T}, \boldsymbol{h}) \in R\big]\big). && \text{(33c)}
\end{aligned}$$

Indeed, by the convexity of S (resp. R), we conclude that $\mathbb{E}\big[\boldsymbol{q} \mid \boldsymbol{q} \in S\big] \in S$ (resp. $\mathbb{E}\big[\boldsymbol{T}, \boldsymbol{h} \mid (\boldsymbol{T}, \boldsymbol{h}) \in R\big] \in R$). □

By summing over all $W \in \operatorname{supp}(\boldsymbol{W})$, all S in $\mathcal{S}_W$ and $R \in \mathcal{R}_{\check{x},W,S}$ and applying this lemma for every term of the sum, we can now deduce an explicit adapted partition.

Theorem 3.7. (Adapted partition for general second stage cost q) *Assume that $\boldsymbol{q}$ and $\boldsymbol{\xi}$ are independent conditionally to $\boldsymbol{W}$ and that* $\operatorname{supp}(\boldsymbol{W})$ *is finite. We define $\mathcal{P}_{\check{x}}$ the following partition*

$$\mathcal{P}_{\check{x}} = \big\{\{W\} \times S \times R \mid W \in \operatorname{supp}(\boldsymbol{W}),\, S \in \mathcal{S}_W,\, R \in \mathcal{R}_{\check{x},W,S}\big\} \tag{34}$$

then $\mathcal{P}_{\check{x}}$ is an adapted partition to $\check{x}$.

Proof. The proof, based on Lemma 3.6, can be found in Appendix B.2. □

Remark 3.8. As we saw in Subsection 3.2, this explicit adapted partition provides a new method to find tight and valid cuts in line 10 of Algorithm 1 without having an approximation error, i.e., $\underline{\gamma}_t = 0$, in the linear case (under Assumptions (LS) and (LS)). Moreover, this explicit adapted partition allows to extend the scope of APM methods. In [24], a method to find an explicit adapted partition for deterministic W and q is presented and the authors state, in Remark 12, that the partition $\mathcal{P}_{\check{x}}$ is adapted without proving it formally. This section provides a formal proof of this statement.

Finally, Siddig and Song [62] presented an algorithm combining ideas of APM methods and SDDP in the finitely supported case. This explicit adapted partition paves the way to an APM-SDDP algorithm with non-finitely supported random variables. □

4. Complexity results

In this section, we give convergence and complexity results for various instances of TFDP Algorithm 1. In Subsection 4.1, we first define the notion of *effective iteration* and deduce an upper bound on the number of effective iterations required by Algorithm 1 to get an ε-solution. We then distinguish between deterministic and randomized node selection processes for line 5 of the algorithm.

For deterministic selection processes, namely the problem-child and explorative node selections, we show in Subsection 4.2 that all iterations are effective. Finally, when the node selection is randomized, we show in Subsection 4.3 the existence of a positive probability for an iteration to be effective. We then deduce a complexity bound on the expected number of iterations.

4.1. Bounding the number of effective iterations

We first recall that the value of Problem (MSP) can be written in a more concise form, by using the nested problem in Eq. (2) and the definition of expected cost-to-go function in Eq. (4), and keeping in mind that ξ_1 is deterministic:

$$\operatorname{val}(\mathrm{MSP}) = \min_{x_1 \in \mathcal{X}_1(x_0,\xi_1)} \ell_1(x_1,\xi_1) + V_1(x_1) \tag{35}$$

Our aim is to show that, for some iteration k, the solution x_1^k is a ε-solution of Eq. (35), and the lower-bound $\underline{V}_0(x_0)$ is ε-tight. Unfortunately, the Assumptions 1 to 5 are not enough to ensure convergence of Algorithm 1: we need a further assumption on the node selection process.

Regardless of node selection, we define the notion of *effective iteration*. Recall that γ_t^F, $\underline{\gamma}_t$, $\bar{\gamma}_t$ are errors in forward Bellman operator and approximation update (see Algorithm 1) at time $t \in [T]$, and $\underline{L}_t$ (resp. $\overline{L}_t$) are Lipschitz bounds on the cuts (resp. upper approximation) at time t. In the remainder of the section we consider a sequence $(\overline{V}_t^k, \underline{V}_t^k, x_k^t)_{t\in[T],k\in\mathbb{N}}$ produced by Algorithm 1.

Definition 4.1. (Effective iteration) For every $t \in [T-1]$, let $\delta_t > 0$ and $\eta_t \geq 0$. By backward induction, we define

$$\varepsilon_{T-1} := \underline{\gamma}_{T-1} + \bar{\gamma}_{T-1}, \tag{36a}$$

$$\varepsilon_t := \varepsilon_{t+1} + (\bar{L}_{t+1} + \underline{L}_{t+1})(\delta_{t+1} + \eta_{t+1}) + \gamma_{t+1}^F + \underline{\gamma}_t + \bar{\gamma}_t, \quad \forall t \in [T-2], \tag{36b}$$

$$\varepsilon_0 := \varepsilon_1 + (\bar{L}_1 + \underline{L}_1)(\delta_1 + \eta_1) + \gamma_1^F. \tag{36c}$$

For $t \in [T-1]$ and $k \in \mathbb{N}$, we say that x_t^k is *ε_t-saturated*, if $\overline{V}_t^k(x_t^k) - \underline{V}_t^k(x_t^k) \leq \varepsilon_t$ and x_t^k is *δ_t-distinguishable* if $\|x_t^k - x_t^\kappa\| > \delta_t$ for all $\kappa < k$ such that x_t^κ is ε_t-saturated.

We say that an iteration $k \in \mathbb{N}$ is *effective* if it generates either a ε_1-saturated point, which is also a ε_0-solution to Problem (35), or a new ε_t-saturated and δ_t-distinguishable point for at least one $t \in [T]$, i.e.,

$$x_1^k \text{ is } \varepsilon_1\text{-saturated and } \ell_1(x_1^k,\xi_1) + V_1(x_1^k) - \operatorname{val}(\mathrm{MSP}) \leq \varepsilon_0, \tag{37a}$$

$$\text{or} \qquad \exists t \in [T-1],\ x_t^k \text{ is } \varepsilon_t\text{-saturated and } \delta_t\text{-distinguishable.} \tag{37b}$$

We now give an upper bound on the number of effective iterations of Algorithm 1 to find an ε_0- optimal solution.

Theorem 4.2. (Bound on effective iterations number) *Let Assumptions 1 – 5 be satisfied and $t \in [T-1]$, assume that $\delta_t \in [0, D_t]$ and $\eta_t \in \mathbb{R}_+$ are given and ε_t defined by* (36)*. Let*

$$\overline{K} := \sum_{t=1}^{T-1} \left(\frac{D_t}{\delta_t} + 1\right)^{n_t}. \tag{38}$$

After at most $\overline{K}+1$ effective *iterations we have a ε_1-lower bound of Problem* (MSP):

$$\underline{V}_0^k(x_0) = \ell_1(x_1^k,\xi_1) + \underline{V}_1^k(x_1^k) \geq \operatorname{val}(\mathrm{MSP}) - \varepsilon_1. \tag{39}$$

Further, there exists, among those $\bar{K}+1$ effective iterations, at least one such that x_1^k is an ε_0-solution to Problem (35)*:*

$$\ell_1(x_1^k,\xi_1) + V_1(x_1^k) \leq \operatorname{val}(\mathrm{MSP}) + \varepsilon_0. \tag{40}$$

Proof. For $t \in [T-1]$, there are at most $(\frac{D_t}{\delta_t}+1)^{n_t}$ disjoint balls [9] of diameter δ_t in a ball of diameter $D_t+\delta_t$ (see [69, A.3.2]). In particular, we cannot compute more than $(\frac{D_t}{\delta_t}+1)^{n_t}$, δ_t-distinguishable points at step t. Thus, after $\bar{K} = \sum_{t=1}^{T-1}(\frac{D_t}{\delta_t}+1)^{n_t}$ effective iterations, for all $t \in [T]$, it is impossible to compute a new δ_t-distinguishable point. Then, as the iteration k is effective and we cannot have (37b), we have (37a) and in particular x_1^k is an ε_0-solution. Moreover, x_1^k is ε_1-saturated. Then,

$$\ell_1(x_1^k, \xi_1) + \underline{V}_1^k(x_1^k) \geq \ell_1(x_1^k, \xi_1) + \overline{V}_1^k(x_1^k) - \varepsilon_1 \geq \ell_1(x_1^k, \xi_1) + V_1(x_1^k) - \varepsilon_1 \tag{41a}$$

$$\geq \min_{x_1 \in \mathcal{X}_1(x_0, \xi_1)} \ell_1(x_1, \xi_1) + V_1(x_1) - \varepsilon_1 = \mathrm{val}\,(\mathrm{MSP}) - \varepsilon_1. \ \square \tag{41b}$$

Remark 4.3. Finally, although the theorems of this section state that we find an ε_0-optimal solution at stage 1, we have no guarantee that the approximations $\underline{V}_t^k$ converge to V_t. We cannot hope that these approximations converge to the true expected cost-to-go functions far from the optimal and reachable trajectories.

Nevertheless, by considering the sets of points that are δ_t-close to every optimal and reachable trajectories, we could hope to have a convergence of strategies generated by $\mathcal{F}(\underline{V}_t^k)$ on those sets. If we add a finite diameter of the support of $\boldsymbol{\xi}_t$ and Lipschitz assumptions for $\boldsymbol{\xi}_t$, we are confident that the proof can be adapted. However, the general case looks harder and might require different ideas for proving complexity results for the convergence of strategies at every stage. $\square$

For a class of specific (deterministic) implementations of Algorithm 1 each iteration is effective, in which case we can directly bound the number of iterations required to obtain an ε_0-optimal solution.

4.2. Deterministic node selection

In this section, we present sufficient condition for an iteration to be effective. Consequently, for two algorithms with deterministic node selections (namely problem-child node selection [7] and explorative node selection [37]), we show that each iteration is effective, yielding a complexity result.

We first define the distance to the set of ε_t-saturated points.

Definition 4.4. Let $t \in [T-1]$ and $k \geq 1$.
We denote by $\boldsymbol{y}_t^k$ the random variable

$$\boldsymbol{y}_t^k := \mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \boldsymbol{\xi}_t) \tag{42}$$

and by d_t^k the distance function to the set of ε_t-saturated points until iteration k:

$$d_t^k(x) := \min_{\kappa < k | x_t^\kappa \text{ is } \varepsilon_t\text{-saturated}} \|x - x_t^\kappa\|. \tag{43}$$

In particular, x_t^k is δ_t-distinguishable if and only if $d_t^k(x_t^k) > \delta_t$.

The following technical lemma, whose proof can be found in Appendix C, shows that if the new state x_t^k (resulting from the choice of ξ_t^k) is either (i) far enough from the set of saturated points, or (ii) yielding a large enough gap, then iteration k is effective.

[9] We consider here balls for the euclidean norm $\|\cdot\|_2$, but the result is still valid with the p-norm $\|\cdot\|_p$ for every $p \in [1, +\infty]$.

Lemma 4.5. *Let Assumptions* 1 – 5 *be satisfied and assume that, for all* $t \in [T-1]$, $\delta_t \in [0, D_t]$, $\eta_t \in \mathbb{R}_+$ *are given and* ε_t *defined by* (36). *Let* $k \in \mathbb{N}^\star$. *If, for all* $t \in [T-1]$, *at least one of the following inequalities is satisfied*

$$\mathbb{E}\big[\overline{V}_t^{k-1}(\boldsymbol{y}_t^k) - \underline{V}_t^{k-1}(\boldsymbol{y}_t^k)\big] \leq \overline{V}_t^{k-1}(x_t^k) - \underline{V}_t^{k-1}(x_t^k) + (\bar{L}_t + \underline{L}_t)\eta_t, \tag{44a}$$

$$\mathbb{E}\big[d_t^k(\boldsymbol{y}_t^k)\big] \leq d_t^k(x_t^k) + \eta_t. \tag{44b}$$

then, iteration k *is effective.*

In [37], Lan suggested choosing x_t^k as the most distinguishable point in a new algorithm called Explorative Dual Dynamic Programming (EDDP). We then speak of *explorative* node selection. The following lemma shows that both these selections lead to effective iterations.

Lemma 4.6. *Let Assumptions* 1 – 5 *hold and assume that, for all* $t \in [T]$ $\delta_t \in [0, D_t]$, $\eta_t = 0$, *are given and* ε_t *is defined by* (36).

We say that we have a problem-child *node selection if for all* $k \in \mathbb{N}^\star$, *and* $t \in [T-1]$, ξ_t^k *is chosen such that it maximizes the current gap, i.e.,*

$$\xi_t^k \in \operatorname*{argmax}_{\xi \in \operatorname{supp}(\boldsymbol{\xi}_t)} \overline{V}_t^{k-1}\big(\mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \xi)\big) - \underline{V}_t^{k-1}\big(\mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \xi)\big). \tag{45a}$$

We say that we have an explorative *node selection if for all* $k \in \mathbb{N}^\star$, *and* $t \in [T-1]$, ξ_t^k *is chosen such that* x_t^k *maximizes the distance to previous* ε_t*-saturated points, i.e.,*

$$\xi_t^k \in \operatorname*{argmax}_{\xi \in \operatorname{supp}(\boldsymbol{\xi}_t)} d_t^k\big(\mathcal{F}_{t-1}(\underline{V}_{t+1}^{k-1})(x_{t-1}^k, \xi)\big). \tag{45b}$$

Then, with a problem-child *or an* explorative *node selection method, each iteration of Algorithm* 1 *is effective.*

Proof. This is a direct consequence of Lemma 4.5. Indeed, since we have $\eta_t = 0$, $x_t^k = \mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \xi_t^k)$ and $\boldsymbol{y}_t^k = \mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \boldsymbol{\xi}_t)$, and since the maximum is greater than the expected value, Eq. (44a) implies Eq. (45a) and Eq. (44b) implies Eq. (45b). □

Lemma 4.6 implies that every iteration of these deterministic node selection methods is effective. Coupled with Theorem 4.2 we easily obtain complexity bounds, for example as follows.

Corollary 4.7. *Let Assumptions* 1 – 5 *hold and assume that every iteration of Algorithm* 1 *is effective (e.g., problem-child or explorative node selection). Further, for simplicity, let the total error be* $\gamma_\Sigma := \sum_{t=1}^{T-1} \underline{\gamma}_t + \bar{\gamma}_t + \gamma_t^F$ *and choose* n, D, L *such that, for all* $t \in [T-1]$, $n_t \leq n$, $D_t = D$, $\overline{L}_t = \underline{L}_t = L$. *Then, for every* $\varepsilon > \gamma_\Sigma$, *sufficiently small (e.g. such that* $\varepsilon \leq 2DL + \gamma_\Sigma$*), Algorithm* 1 *finds an* ε*-first stage solution* x_1^k *within at most* $\bar{K}_\varepsilon$ *iterations where*

$$\bar{K}_\varepsilon := \left(\frac{2DL}{\varepsilon - \gamma_\Sigma}\right)^n (T-1)^{n+1}. \tag{46}$$

Proof. We set $\delta_t := \frac{\varepsilon-\gamma_\Sigma}{2L(T-1)}$ and $\eta_t := 0$ for all $t \in [T-1]$. Then, as $\varepsilon \leq 2DL + \gamma_\Sigma$ we have $\delta_t \leq D = D_t$. Moreover, ε_0, defined in Eq. (36), satisfies

$$\varepsilon_0 = \sum_{t=1}^{T-1} \left[(\bar{L}_t + \underline{L}_t)(\delta_t + \eta_t) + \underline{\gamma}_t + \bar{\gamma}_t + \gamma_t^F \right] = (T-1)2L\delta + \gamma_\Sigma = \varepsilon. \tag{47}$$

With this setting, we have that $\overline{K}$, as defined in (38), satisfies

$$\bar{K} \leq (T-1)\left(\frac{D}{\delta} + 1\right)^n = T\left(\frac{2DL\,(T-1)}{\varepsilon - \gamma_\Sigma} + 1\right)^n.$$

Now, as ε is assumed to be small enough to have $2DL/(\varepsilon - \gamma_\Sigma) \leq 1$ (i.e. we have $\varepsilon \leq 2DL + \gamma_\Sigma$) we get

$$\bar{K} \leq (T-1)\left(\frac{2DL(T-1)}{\varepsilon - \gamma_\Sigma}\right)^n = \bar{K}_\varepsilon.$$

By assumption all iterations are effective and Theorem 4.2 ends the proof. □

Remark 4.8. Note that the maximum in (45a) (resp. (45b)) is easily obtained under finite noise Assumption (FSN). Indeed, we can compute the gap (resp. the distance) for every ξ_t in the support of $\boldsymbol{\xi}_t$ and keep ξ_t^k maximizing the gap.

However, without finite noise Assumption (FSN), we just need to find a ξ_t^k leading to a gap worse than the expected gap (see Lemma 4.5), and not necessarily a maximizer. This paves the way for a deterministic node selection, with non-finitely supported random variables.

4.3. Randomized algorithms

When the choice of ξ_t^k is made randomly, there is no guarantee that the iteration will be effective. However, applying a technical nested Hoeffding lemma shown in Appendix D.1, we show that there is a positive probability p for an iteration to be effective. Then, by comparing the time to obtain an effective iteration to a geometric random variable of parameter p in Appendix D.2, we deduce a bound on the expected number of iterations required to get an ε-optimal solution.

Remark 4.9. (Notational difficulty of randomized algorithm on stochastic problem) We are now considering a stochastic algorithm for solving the MSP Problem (MSP). Thus, there are two sources of randomness: the intrinsic $(\boldsymbol{\xi}_t)_{t\in[T]}$ and the node selection $\xi_t^k = \tilde{\xi}_t^k$. To distinguish both, we denote in bold random variables that are $(\boldsymbol{\xi}_t)_{t\in[T]}$ measurable, with a tilde random variables that are $(\tilde{\xi}_t^k)_{t\in[T],k\in\mathbb{N}^*}$ measurable (and with both if they are neither).

For example, the trajectory determined during the forward phase $(\tilde{x}_t^k)_{t\in[T]}$ only depends on the past node selections, whereas the tentative points $\tilde{\boldsymbol{y}}_t^k$ depends both on the past node selections and the actual realization of $\boldsymbol{\xi}_t$.

Under Assumption (FSN), this discussion is usually avoided by representing the dependence on $(\boldsymbol{\xi}_t)_{t\in[T]}$ with a (finite) scenario tree, and indexing the variables by the tree nodes. □

Let $(\mathcal{A}^k)_{k\in\mathbb{N}^\star}$ be the filtration such that $\mathcal{A}^k := \sigma\big(\tilde{\xi}_t^\kappa\big)_{t\in[T-1],\kappa\in[k]}$, and $\mathcal{A}^\infty = \bigcup_{k\in\mathbb{N}} \mathcal{A}^k$.

In particular, a random variable measurable with $\mathcal{A}^k$ knows all node selection up to iteration k, which include, for example, $\underline{V}_t^k$ for all $t \in [T]$.

Lemma 4.10. *Let Assumptions* 1 – 5 *be satisfied and assume that, for all* $t \in [T-1]$, $\delta_t \in [0, D_t]$, $\eta_t \in \mathbb{R}_+$ *are given and* ε_t *defined by* (36). *Further, assume that in Algorithm* 1, *line* 5, *we draw* ξ_t^k *randomly according to the distribution of* $\boldsymbol{\xi}_t$, *and independently of all other* $\tilde{\xi}_\tau^\kappa$ *as well as* $(\boldsymbol{\xi}_\tau)_{\tau\in[T-1]}$.
Then, for all iteration $k \in \mathbb{N}$ *of Algorithm* 1 *and all event* $A^{k-1} \in \mathcal{A}^{k-1}$ *such that* $\mathbb{P}\big[A^{k-1}\big] > 0$, *we have*

$$\mathbb{P}\Big[\textit{Iteration } k \textit{ is effective.} \,\Big|\, A^{k-1}\Big] \geq \prod_{t=1}^{T} \left(1 - e^{\frac{-2\eta_t^2}{D_t^2}}\right). \tag{48}$$

Proof. Let $\mathcal{A}^\infty := \sigma\big(\tilde{\xi}_t^\kappa\big)_{t\in[T-1],\kappa\in\mathbb{N}^\star}$ and $k \in \mathbb{N}^\star$. By Lemma 4.5, we have

$$\begin{aligned}\mathbb{P}\big[&\text{Iteration } k \text{ is effective.} \,|\, A^{k-1}\big] \\ &\geq \mathbb{P}\Big[\forall t \in [T-1],\, \mathbb{E}\big[\tilde{d}_t^k(\tilde{\boldsymbol{y}}_t^k) \,|\, \mathcal{A}^\infty\big] < \tilde{d}_t^k(\tilde{x}_t^k) + \eta_t \,|\, A^{k-1}\Big].\end{aligned} \tag{49}$$

For $t \in [T-1]$, let $\mathcal{A}_t^k := \sigma\big(\mathcal{A}^{k-1}, (\tilde{\xi}_k^t)_{\tau\in[t]}\big)$. We have that $\sigma\big(\tilde{d}_t^k(\tilde{\boldsymbol{y}}_t^k)\big) \subset \sigma(\mathcal{A}_{t-1}^k, \boldsymbol{\xi}_t)$ from which we deduce that $\mathbb{E}\big[\tilde{d}_t^k(\tilde{\boldsymbol{y}}_t^k) \,|\, \mathcal{A}^\infty\big] = \mathbb{E}\big[\tilde{d}_t^k(\tilde{\boldsymbol{y}}_t^k) \,|\, \mathcal{A}_{t-1}^k\big]$. We define the events $E_t^k := \big\{\omega \in \Omega \mid \mathbb{E}\big[\tilde{d}_t^k(\tilde{\boldsymbol{y}}_t^k) \mid \mathcal{A}_{t-1}^k\big] < \tilde{d}_t^k \tilde{x}_t^k + \eta_t\big\}$. Thus,

$$\mathbb{P}\big[\text{Iteration } k \text{ is effective} \,|\, A^{k-1}\big] \geq \mathbb{P}\big[\bigcap_{t=1}^{T-1} E_t^k \,|\, A^{k-1}\big].$$

By applying Lemma D.2 with the random variables $\big(\tilde{\xi}_t^k\big)_{k\in\mathbb{N},t\in[T-1]}$, the filtration $(\mathcal{A}_t^k)_{k\in\mathbb{N},t\in[T-1]}$ and the measurable function

$$ft^k : ((\xi_\tau^\kappa)_{\tau\in[T-1],\kappa\in[k-1]}, (\xi_\tau^k)_{\tau\in[t]}) \mapsto d_t^k(x_t^k)$$

taking its value in $[0, D_t]$, we have

$$\mathbb{P}\Big[\bigcap_{t=1}^{T-1} E_t^k \,\Big|\, A^{k-1}\Big] \geq \prod_{t=1}^{T} \left(1 - e^{\frac{-2\eta_t^2}{D_t^2}}\right)$$

which gives Eq. (48). □

We now give a complexity results for all TFDP algorithms (following framework of Algorithm 1) where the choice of ξ_t^k is made randomly.

Theorem 4.11. *Let Assumptions* 1 – 5 *be satisfied and assume that in line* 5, *we draw* ξ_t^k *randomly according to the distribution of* $\boldsymbol{\xi}_t$, *and independently from the previous* ξ_τ^κ.
Further, for simplicity, let the total error be $\gamma_\Sigma := \sum_{t=1}^{T-1} \underline{\gamma}_t + \bar{\gamma}_t + \gamma_t^F$ *and choose* n, D, L *such that, for all* $t \in [T-1]$, $n_t \leq n$, $D_t = D$, $\overline{L}_t = \underline{L}_t = L$.
Then, for $\varepsilon > \gamma_\Sigma$, *sufficiently small (e.g., such that* $\varepsilon \leq 4DL + \gamma_\Sigma$), *the expected number of iterations of Algorithm* 1 *required to find an* ε*-solution* x_1^k *to problem* (35), *i.e., such that* $\ell_1(x_1^k, \xi_1) + V_2(x_1^k) \leq \mathrm{val}\,(\mathrm{MSP}) + \varepsilon$ *is bounded by* $(T-1)\Big(\frac{4DL(T-1)}{\varepsilon - \gamma_\Sigma}\Big)^{n+2(T-1)}$.

Proof. We set for all $t \in [T-1]$, $\delta_t = \eta_t = \frac{\varepsilon - \gamma_\Sigma}{4L(T-1)}$. Then, as $\varepsilon \leq 4DL + \gamma_\Sigma$ we have $\eta_t = \delta_t \leq D = D_t$. Moreover, ε_0, defined in Eq. (36), satisfies

$$\varepsilon_0 = \sum_{t=1}^{T-1} \left[(\bar{L}_t + \underline{L}_t)(\delta_t + \eta_t) + \underline{\gamma}_t + \bar{\gamma}_t + \gamma_t^F \right] = (T-1)2L \times 2\frac{\varepsilon - \gamma_\Sigma}{4L(T-1)} + \gamma_\Sigma = \varepsilon.$$

Let $\tilde{K}$ the (random) number of iterations needed to compute

$$\bar{K}_\varepsilon := \sum_{t=1}^{T-1} \left(1 + \frac{D_t}{\delta_t}\right)^{n_t} \leq \left(\frac{4DL}{\varepsilon - \gamma_\Sigma}\right)^n (T-1)^{n+1}$$

effective iterations, then by Theorem 4.2, Algorithm 1 finds an ε-solution after at most $\tilde{K}$ iterations. Let

$$p := \prod_{t=1}^{T} \left(1 - \exp\left(\frac{-2\eta_t^2}{D_t^2}\right)\right),$$

by Lemma 4.10, for $A^{k-1} \in \mathcal{A}^{k-1}$, we have $\mathbb{P}\big[\text{Iteration } k \text{ is effective} \,\big|\, A^{k-1}\big] \geq p$. Thus, by Lemma D.3, we have $\mathbb{E}\big[\tilde{K}\big] \leq \frac{K_\varepsilon}{p}$.

Moreover, since $x \mapsto \frac{x}{1-e^{-x}}$ is an increasing function on $(0, \frac{1}{2}]$, then for all $x \in (0,1]$, we have $\frac{1}{1-e^{-x}} \leq \frac{1}{1-e^{-1}} \times \frac{1}{x} \leq 1.6 \times \frac{1}{x}$. Thus, as $\frac{2\eta_t^2}{D_t^2} \leq 1$, we have that

$$\frac{1}{p} \leq \prod_{t=1}^{T-1} 1.6 \times \frac{D_t^2}{2\eta_t^2} \leq \left(\frac{4DL(T-1)}{\varepsilon - \gamma_\Sigma}\right)^{2(T-1)}.$$

We then obtain

$$\begin{aligned}\mathbb{E}\big[\tilde{K}\big] \leq \frac{K_\varepsilon}{p} &\leq \left(\frac{4DL}{\varepsilon - \gamma_\Sigma}\right)^n (T-1)^{n+1} \times \left(\frac{4DL(T-1)}{\varepsilon - \gamma_\Sigma}\right)^{2(T-1)} \\ &= (T-1)\left(\frac{4DL(T-1)}{\varepsilon - \gamma_\Sigma}\right)^{n+2(T-1)}.\end{aligned}$$

□

Remark 4.12. (Stochastic dominance and comparison with finitely supported noise) The proof of Theorem 4.11 actually gives more information on the (random) number of iteration $\tilde{K}$ after which we obtain an ε-solution: $\tilde{K}$ is stochastically dominated by a random variable with a negative binomial distribution representing the number of trials needed to obtain $\bar{K}_\varepsilon$ successes with probability of success p, (see Lemma D.3).

Further, under finitely supported noise assumption Assumption (FSN), the probability of choosing the problem child ξ_t^k (cf Lemma 4.6) for each $t \in [T-1]$ is lower bounded by $\prod_{t=1}^{T-1} \frac{1}{|\operatorname{supp}(\boldsymbol{\xi}_t)|}$. Then, Lemma 4.10 still holds after replacing the right-hand side probability of success by $\prod_{t=1}^{T-1} \frac{1}{|\operatorname{supp}(\boldsymbol{\xi}_t)|}$. We can then deduce other complexity bounds under Assumption (FSN). For example, in [37], assuming that $|\operatorname{supp}(\boldsymbol{\xi}_t)| \leq N$, for all $t \in [T-1]$, the probability of having an effective iteration is bounded by $\frac{1}{N^{T-1}}$. □

5. Extension to risk-averse setting

Finally, we now discuss how we can go farther than the risk-neutral setting. These extensions involve a maximization problem in the dynamic programming equation, arising for example from multistage risk-averse, robust or distributionally robust problems. Algorithm 1 can be adapted to such problems, by changing the definitions of the Bellman operators.

Further, in the risk-neutral case, Algorithm 1 is not symmetrical in its treatment of lower and upper approximations. As noted in Remark 2.2, for a minimization problem, in Algorithm 1, the forward phase, line 6, should be done using the lower approximations $\underline{V}_t^k$. More generally, one should use an *outer approximation* (that is under approximation for min sub-problems and upper approximations for max sub-problems) during the forward phase to be able to explore the state space. Thus, for those min-max problems, the computation of upper approximations $\overline{V}_t^k$ is not optional.

Minimax problems. Baucke, Downward and Zakeri, in [7], presented a convergent problem-child algorithm to solve stochastic minimax dynamic programs. Although our framework of Algorithm 1 do not handle such minimax problem, we can extend it to do so. More precisely, we consider a problem where the decision maker chooses $x_t \in \mathcal{X}_t(x_{t-1}, y_{t-1}, \xi_t)$, and then an adversary chooses $y_t \in \mathcal{Y}_{t-1}(x_{t-1}, y_{t-1}, x_t, \xi_t)$. Thus, the Bellman operators are now defined as

$$\begin{aligned}&\mathcal{B}_{t-1}(\tilde{V})(x_{t-1}, y_{t-1})\\ &= \mathbb{E}\big[\min_{x_t \in \mathcal{X}_t(x_{t-1}, y_{t-1}, \boldsymbol{\xi}_t)} \max_{y_t \in \mathcal{Y}_{t-1}(x_{t-1}, y_{t-1}, x_t, \boldsymbol{\xi}_t)} \ell_t(x_t, y_t, \boldsymbol{\xi}_t) + \tilde{V}(x_t, y_t)\big]. \end{aligned} \tag{50}$$

The reachable sets then become

$$X_0^r = \{x_0\}, \qquad Y_0^r = \{y_0\}, \tag{51a}$$

$$X_t^r = \bigcup_{x_{t-1} \in X_{t-1}^r} \bigcup_{y_{t-1} \in Y_{t-1}^r} \bigcup_{\xi_t \in \Xi_t} \mathcal{X}_t(x_{t-1}, y_{t-1}, \xi_t) \qquad \forall t \in [T], \tag{51b}$$

$$Y_t^r = \bigcup_{x_{t-1} \in Y_{t-1}^r} \bigcup_{y_{t-1} \in Y_{t-1}^r} \bigcup_{x_t \in X_t^r} \bigcup_{\xi_t \in \Xi_t} \mathcal{Y}_t(x_{t-1}, y_{t-1}, x_t, \xi_t) \qquad \forall t \in [T]. \tag{51c}$$

In the forward phase, as in Algorithm 1, the γ_t^F-optimal solution x_t^k should be chosen thanks to the approximation $\underline{V}_t^{k-1}$. However, as we maximize over y_t, y_t^k must be a γ_t^F-optimal solution of the step problem with the approximation $\overline{V}_t^{k-1}$:

$$\begin{aligned}x_t^k &= \mathcal{F}_{t-1}^{\min}(\underline{V}_t^{k-1})(x_{t-1}, y_{t-1}, \xi_t^k)\\ &\in \gamma_t^F\text{-}\operatorname*{arg\,min}_{x_t \in \mathcal{X}_t(x_{t-1}, y_{t-1}, \xi_t^k)} \max_{y_t \in \mathcal{Y}_{t-1}(x_{t-1}, y_{t-1}, x_t, \xi_t^k)} \ell_t(x_t, y_t, \xi_t^k) + \underline{V}_t^{k-1}(x_t, y_t), \end{aligned} \tag{52a}$$

$$\begin{aligned}y_t^k &= \mathcal{F}_{t-1}^{\max}(\overline{V}_t^{k-1})(x_{t-1}, y_{t-1}, x_t^k, \xi_t^k)\\ &\in \gamma_t^F\text{-}\operatorname*{argmax}_{y_t \in \mathcal{Y}_{t-1}(x_{t-1}, y_{t-1}, x_t^k, \xi_t^k)} \ell_t(x_t, y_t, \xi_t^k) + \overline{V}_t^{k-1}(x_t, y_t). \end{aligned} \tag{52b}$$

Assuming that the reachable sets X_t^r and Y_t^r have finite dimensions d_x and d_y and diameter D, and that the objective function are L-Lipschitz, the convergence and complexity results still hold developing on the ideas of [69]. The upper bound on the number of effective iterations then becomes

$$K_\varepsilon := \Big(\frac{2DL}{\varepsilon - \gamma_\Sigma}\Big)^{d_x + d_y} (T-1)^{d_x + d_y + 1}.$$

Robust. Closely related, in [27], Georghiou, Tsoukalas and Wiesemann presented the Robust Dual Dynamic Programming algorithm (RDDP) to solve multistage robust optimization problems. In such problems, instead of minimizing the expec-

tation like in Eq. (MSP), we minimize considering the worst case scenario $\xi_t \in \Xi_t$. In this setting, the Bellman operator reads

$$\mathcal{B}_{t-1}(\tilde{V}) = \max_{\xi_t \in \Xi_t} \min_{x_t \in \mathcal{X}_t(x_{t-1}, \xi_t)} \ell_t(x_t, \xi_t) + \tilde{V}(x_t). \tag{53}$$

Note that this robust setting can be seen as a particular case of minimax problems where we have deterministic random variables. Indeed, if we invert the order of max and min, either by changing the indices or by taking the opposite, and Eq. (53) can be written as Eq. (50) where ξ_t of (53) plays the role of y_t and the $\boldsymbol{\xi}_t$ of (50) are deterministic parameter. The upper bound on the number of effective iterations then becomes $K_\varepsilon := \left(\frac{2DL}{\varepsilon - \gamma_\Sigma}\right)^{d_x + d_\xi} (T-1)^{d_x + d_\xi + 1}$.

Risk-averse. Multistage stochastic problems in the risk-averse setting are MSP where the expectation is replaced by a multiperiod risk measure. In the nested coherent risk measure framework we present conditions under which Algorithm 1 can be adapted.

Let ρ be a coherent risk measure (see [3, 4] or [58, Def 6.4]) the Bellman operator in the risk-averse setting reads

$$\mathcal{B}_{t-1}(\tilde{V}) : x_{t-1} \mapsto \rho\Big(\min_{x_t \in \mathcal{X}_t(x_{t-1}, \boldsymbol{\xi}_t)} \ell_t(x_t, \boldsymbol{\xi}_t) + \tilde{V}(x_t) \Big). \tag{54}$$

We recall a classical Fenchel representation theorem for proper, lower semicontinuous, law-invariant, coherent risk measure (see [58, Thm 6.5]). For every random variable $\boldsymbol{z} \in L_1(\Omega, \mathcal{A}, \mathbb{P}, \mathbb{R})$, we have

$$\rho(\boldsymbol{z}) = \max_{\boldsymbol{y} \in \mathfrak{A}_\rho} \mathbb{E}_{\mathbb{P}}\big[\boldsymbol{y}\boldsymbol{z}\big], \tag{55}$$

$$\text{where} \quad \mathfrak{A}_\rho := \left\{ \boldsymbol{y} \in L_\infty(\Omega, \mathcal{A}, \mathbb{P}, \mathbb{R}) \;\middle|\; \begin{array}{l} \mathbb{E}\big[\boldsymbol{y}\big] = 1,\ \boldsymbol{y} \geq 0 \ a.s., \\ \mathbb{E}\big[\boldsymbol{y}\boldsymbol{z}'\big] \leq \rho(\boldsymbol{z}),\ \forall \boldsymbol{z}' \in L_1(\Omega, \mathcal{A}, \mathbb{P}, \mathbb{R}) \end{array} \right\}.$$

With this representation, we get

$$\mathcal{B}_{t-1}(\tilde{V}) = \max_{\boldsymbol{y} \in \mathfrak{A}_\rho} \mathbb{E}_{\mathbb{P}}\Big[\min_{x_t \in \mathcal{X}_t(x_{t-1}, \boldsymbol{\xi}_t)} \boldsymbol{y}\ \ell_t(x_t, \boldsymbol{\xi}_t) + \boldsymbol{y}\tilde{V}(x_t) \Big]. \tag{56}$$

Up to a slight change of notation, we can write this problem as a minimax problem. In particular, a sufficient condition to obtain convergence and complexity bounds for risk-averse MSP is that the set $\mathfrak{A}_\rho$ has a finite dimension and a finite diameter. For example, if Ω is finite, $\mathfrak{A}_\rho$ is contained in the space of random variables in Ω, isomorphic to a simplex of dimension $|\Omega| - 1$ which has finite diameter. More generally, if $\mathfrak{A}_\rho$ is contained in the convex hull of n random variables $(\boldsymbol{y}_k)_{k \in [n]}$, then $\mathfrak{A}_\rho$ has a finite diameter smaller than $\max_{k,\ell \in [n]}(\|\boldsymbol{y}_k - \boldsymbol{y}_\ell\|_\infty)$ and a finite dimension smaller than $n-1$. Thus, we obtain complexity results similar to Corollary 4.7 and Theorem 4.11 with $K_\varepsilon := \left(\frac{2DL}{\varepsilon - \gamma_\Sigma}\right)^{d+n-1} (T-1)^{d+n}$.

We now comment on the particular case of the average value at risk [52] with value $\alpha \in [0, 1)$, denoted $AV@R_\alpha$ and defined as:

$$AV@R_\alpha(\boldsymbol{z}) := \inf_{s \in \mathbb{R}} \Big\{ s + \frac{1}{1-\alpha} \mathbb{E}_{\mathbb{P}}\big[\max(\boldsymbol{z} - s, 0)\big] \Big\}. \tag{57}$$

We cannot use the dual representation Eq. (55) to derive complexity bounds as $\mathfrak{A}_{AV@R}$ has, in general, non-finite dimension. However, note that in Eq. (57), since $AV@R_\alpha(\boldsymbol{z}) \leq \mathbb{E}_{\mathbb{P}}[\boldsymbol{z}]/(1-\alpha)$ the infimum on s over $\mathbb{R}$ can be replaced by a minimum on the compact interval $[0, \frac{1}{1-\alpha}\mathbb{E}_{\mathbb{P}}\big[\boldsymbol{z}\big]]$. To obtain an upper bound that does not depend on k and x_{t-1}, we set $\boldsymbol{z} = \min_{x_t \in \mathcal{X}(x_{t-1}, \boldsymbol{\xi}_t)} \ell_t(x_t, \boldsymbol{\xi}_t) + \underline{V}_t^k(x_t)$ then $\mathbb{E}_{\mathbb{P}}\big[\boldsymbol{z}\big]$ is upper bounded by $\min_{x_t \in X_t^r} \mathbb{E}\big[\ell_t(x_t, \boldsymbol{\xi}_t) + \overline{V}_t^1(x_t)\big]$ which has a finite value by Assumption 3. Thus, MSP with nested average value at risk measure can be handled by this framework, and we can obtain complexity results similar to Corollary 4.7 and Theorem 4.11 with $K_\varepsilon := \left(\frac{2DL}{\varepsilon - \gamma_\Sigma}\right)^{d+1} (T-1)^{d+2}$.

Acknowledgements. We sincerely thank Mr. Julien Weibel for interesting and useful discussions concerning Appendix D.2, as well as anonymous referees for their insightful comments. This research benefited from the support of the FMJH Program Gaspard Monge for optimization and operations research and their interactions with data science.

A. Cut methodologies

In this section, for the sake of completeness, we give several cuts that are used in different algorithms to solve particular multistage problems.

Cut	Oracle needed	Setting and advantages
Benders	First order	Convex, simple to implement
Reverse norm	Zeroth order and Lipschitz constant	Lipschitz
Step	Zeroth order, ε and γ	Monotonic
Lagrangian	Solving a dual problem	Problem with small duality gap
Integer	Zeroth order	Binary variables
Adaptive partition	Adapted partition oracle	Linear, whole tangent cone
Generalized conjugacy	Conjugate computation	Regularisation
Saddle	First order and Lipschitz constant	Minimax problems
Fenchel conjugate	Compute Fenchel transform of Bellman equation	Linear, Exact upper bound

Table A.1: Synthesis of different cuts and oracle required

A.1. Benders cuts for convex functions

The most commonly used cuts are the Benders cuts which are affine functions. This kind of cut only works if the expected cost-to-go functions are *convex*.

The word cut is actually used because the graph of a Benders cut is a hyperplane that is tangent to the epigraph of the approximated function.

Proposition A.1. *Let F be a* convex *function and $g \in \partial F(x)$ a subgradient of F. We define the Benders cut f at $\hat{x}$ as*

$$f(x) := F(\hat{x}) + g^\top (x - \hat{x}). \tag{58}$$

Then, f is valid and tight at $\hat{x}$.

Proof. By definition of a subgradient, $f(x) := F(\hat{x}) + g^\top(x - \hat{x}) \leq F(x)$, thus f is valid. By definition of f, $f(\hat{x}) = F(\hat{x}) + g^\top(\hat{x} - \hat{x}) = F(\hat{x})$ thus f is tight at $\bar{x}$. □

We see that a first-order oracle for the function F, i.e., an oracle that returns the value $F(\hat{x})$ and a subgradient $g \in \partial F(\hat{x})$ for an input $\hat{x}$, provides a direct algorithm to compute Benders cut.

A.2. Reverse norm cuts for Lipschitz functions

Stochastic Lipschitz Dynamic Programming (SDLP) presented in [1] provides an algorithm to deal with non-convex Lipschitz multistage stochastic programs. In this setting, the cost functions ℓ_t are simply assumed to be Lipschitz continuous. Thus, the expected cost-to-go functions F_t are not necessarily convex and Benders cuts are not valid anymore. Ahmed, Cabral and da Costa replaced these cuts with reverse norm cuts or augmented lagrangian cuts using only the Lipschitz property of expected cost-to-go functions.

Proposition A.2. *Let F be a function with Lipschitz constant L (for norm $\|\cdot\|$). We define the reverse norm cut f of F at $\bar{x}$ as*

$$f(x) := F(\hat{x}) - L\|x - \hat{x}\|. \tag{59}$$

Then, f is valid and 0*-tight at $\hat{x}$.*

Proof. For any given x and $\hat{x}$

$$\begin{aligned} f(x) = F(\hat{x}) - L\|x - \hat{x}\| &= F(\hat{x}) - F(x) + F(x) - L\|x - \hat{x}\| && \text{(60a)} \\ &\leq L\|\hat{x} - x\| + F(x) - L\|x - \hat{x}\| = F(x). && \text{(60b)} \end{aligned}$$

Thus, f is valid. By definition of f, $f(\hat{x}) = F(\hat{x}) - L\|\hat{x} - \hat{x}\| = F(\hat{x})$ thus f is tight at $\bar{x}$. □

We see that a zeroth order oracle for the function F, i.e., an oracle that returns the value $F(\hat{x})$ for an input $\hat{x}$, together with a Lipschitz constant L provides a direct algorithm to compute reverse norm cuts. Thus, SDLP integrates our framework in Algorithm 1.

We can also define the norm cut $f(x) := F(\hat{x}) + L\|x - \hat{x}\|$. These norm cuts can be used to compute $\overline{V}_t^k$. The algorithm Tropical dynamic programming in [2] uses this upper cuts together with Benders cuts for lower approximation $\underline{V}_t^k$ and thus integrates the framework of Algorithm 1.

A.3. Step cuts for monotonic functions

We now look at "almost monotonic" expected cost-to-go functions.

Proposition A.3. *Let F be a function such that there exists $\delta > 0$ and $\gamma \geq 0$ with*

$$\forall x, y, \quad x \leq y + \delta \mathbf{1} \implies F(x) \leq F(y) + \gamma, \tag{61}$$

where $\mathbf{1}$ is the vector whose coefficients are all equal to 1. We assume that F is upper bounded by $\bar{M}$. For a point $\hat{x}$, we define the upper increasing step cut *f as*

$$f(x) := \begin{cases} F(\hat{x}) + \gamma & \text{if } x \leq \hat{x} + \delta \mathbf{1} \\ \bar{M} & \text{otherwise} \end{cases}. \tag{62}$$

Then, the upper increasing step cut f satisfies

$$f(x) \geq F(x), \quad \forall x, \tag{63}$$

$$f(\check{x}) \leq F(\check{x}) + \gamma. \tag{64}$$

Proof. Since $\hat{x} \leq \hat{x} + \delta$, $f(\hat{x}) = F(\hat{x}) + \gamma$. Moreover, if $x \leq \hat{x} + \delta \mathbf{1}$, by Eq. (61) $F(x) \leq F(\hat{x}) + \gamma = f(x)$ and otherwise $f(x) = \bar{M} \geq F(x)$. □

We could also define lower increasing step cuts for functions verifying Eq. (61). These cut methods also adapt to lower bounded decreasing functions, we will define in the same way upper and lower decreasing step cuts. However, these cuts are not Lipschitz. To integrate the framework of Algorithm 1, we could adapt these step cuts by interpolating with affine functions between the constant regions of the cuts.

In [50], Philpott, Wahid and Bonnans presented an algorithm called mixed integer dynamic approximation scheme (MIDAS). This method applies to multistage mixed integer problems, given as a maximization problem. After adapting the problem by taking the opposite of the expected cost-to-go function, adding the constant γ and choosing the right affinely interpolated step cut, the algorithm MIDAS integrates the framework of Algorithm 1 with step cuts.

A.4. Lagrangian cuts

Lagrangian cuts were introduced for TFDP by Zou, Ahmed and Sun in [71]. These cuts are based on the Lagrangian relaxation of an optimization problem.

Proposition A.4. *Let F be a function, H be a convex function and $x \mapsto Y(x)$ be a graph-convex set-valued mapping see* [54, p155] *such that F is defined as*

$$F(x) := \inf_{y \in Y(x)} \ell(y). \tag{65}$$

We define the Lagrangian cut f at $\hat{x}$ as

$$f(x) := \hat{\lambda}^\top x + \hat{\beta}, \tag{66}$$

where

$$\hat{\lambda} \in \operatorname*{argmax}_{\lambda} \lambda^\top \hat{x} + \inf_{y,z \,|\, y \in Y(z)} \ell(y) - \lambda^\top z, \tag{67a}$$

$$\hat{\beta} = \inf_{y,z \,|\, y \in Y(z)} \ell(y) - \hat{\lambda}^\top z. \tag{67b}$$

Then, the Lagrangian cut is valid and tight at $\hat{x}$.

Proof. Consider $x \in \mathrm{dom}(Y)$. We rely on a strong duality result:

$$F(x) = \inf_{y \in Y(x)} \ell(y), = \inf_{y,z \,|\, y \in Y(z) \text{ and } z=x} \ell(y), \tag{68a}$$

$$= \inf_{y,z \,|\, y \in Y(z)} \max_{\lambda} \ell(y) + \lambda^\top (x - z), \tag{68b}$$

$$= \max_{\lambda} \lambda^\top x + \inf_{y,z \,|\, y \in Y(z)} \ell(y) - \lambda^\top z. \tag{68c}$$

Indeed, as Y is graph-convex, we have that $\{(y, z) \,|\, y \in Y(z)\}$ is a non-empty convex set. Thus, we have

$$f(\hat{x}) = \hat{\lambda}^\top \hat{x} + \inf_{y,z \,|\, y \in Y(z)} \ell(y) - \hat{\lambda}^\top z, \tag{69a}$$

$$= \max_{\lambda} \lambda^\top \hat{x} + \inf_{y,z \,|\, y \in Y(z)} \ell(y) - \lambda^\top z = F(x), \tag{69b}$$

and f is tight at $\hat{x}$. Moreover,

$$f(x) = \hat{\lambda}^\top x + \inf_{y,z \,|\, y \in Y(z)} \ell(y) - \hat{\lambda}^\top z, \tag{70a}$$

$$\leq \max_{\lambda} \lambda^\top x + \inf_{y,z \,|\, y \in Y(z)} \ell(y) - \lambda^\top z = F(x), \tag{70b}$$

and then f is valid. □

Note that this result is still true without the convexity assumption if we replace the tightness result with a lower γ-tighness result where γ is the duality gap.

Secondly, in this simplified setting and when the variable x takes value in a continuous space, the Lagrangian cut can be seen as a Benders cut since $\hat{\lambda}$ is a subgradient of F at $\hat{x}$. Nevertheless, the point of view of Lagrangian allows new ideas to compute cuts. In particular, one can define the Lagrangian cut with a different relaxation to deal with more complex setting such as integer cases as presented in [71]. We can also combine the Lagrangian cut with the reverse norm cut, such ideas are presented under the name *augmented Lagrangian cuts* in [1]. Thus, the algorithm SLDP from [1] integrates the framework of Algorithm 1.

Zhang and Sun in [69] generalized these Lagrangian cuts by introducing the point of view of generalized conjugacy (see [54, Chapter 11]). They called these new cuts *generalized conjugacy cuts*, and proved them to be tight and valid, thanks to the Fenchel-Young inequality.

A.5. Integer optimality cuts

The integer optimality cuts were first introduced by Laporte and Louveaux in [38] for 2-stage sochastic integer problems.

Proposition A.5. *Let F be a function taking value in $\{0,1\}^d$ and $\hat{x} \in \{0,1\}^d$ be a binary vector with $\hat{S} := \{i \,|\, x_i = 1\}$. We assume that F is lower bounded by $\underline{M}$.*

We define the integer optimality cut *f as*

$$f(x) := (F(\hat{x}) - \underline{M})\big(\sum_{i \in \hat{S}} x_i - \sum_{i \notin \hat{S}} x_i - |\hat{S}| + 1\big) + \underline{M}. \tag{71}$$

Then, f is a valid and tight at $\hat{x}$.

Proof. We have that $f(\hat{x}) = (F(\hat{x}) - \underline{M})\big(|\hat{S}| - 0 - |\hat{S}| + 1\big) + \underline{M} = F(\hat{x})$. Thus, f is tight at $\hat{x}$. Let $x \in \{0,1\}^d$ different from $\hat{x}$, we have $\sum_{i\in\hat{S}} x_i - \sum_{i\notin\hat{S}} x_i \leq |\hat{S}| - 1$. Then, $f(x) \leq \underline{M} \leq F(x)$ and thus f is valid. □

In [71], Zou, Ahmed and Sun presented an algorithm called Stochastic dual dynamic integer programming (SDDiP), suggesting to use integer optimality cuts or Lagrangian cuts instead of classical Benders cuts. By tightness and validity of these cuts, the algorithm SDDiP integrates the framework of Algorithm 1 (with a potentially large Lipschitz constant).

B. Explicit valid and adapted partition, geometric tools and proofs

In this appendix, we give an elementary definition of the partition $\mathcal{S}_W$. This then allow us to prove Theorems 3.5 and 3.7 of Section 3.

B.1. Another elementary definition of $\mathcal{S}_W$

We give an alternative and equivalent definition of the partition $\mathcal{S}_W$ without directly using the fundamental notion of the secondary fan.

Let us denote by $(w_i)_{i\in[m]}$ the columns of the matrix $W \in \mathbb{R}^{l\times m}$, and choose $q \in \mathbb{R}^m$. We shall think of $(w_i)_{i\in[m]}$ as a *vector configuration* in $\mathbb{R}^l$, and q as a *height* vector: for each $i \in [m]$, we draw the point $(w_i, q_i) \in \mathbb{R}^l \times \mathbb{R}$. We now consider the convex hull E of the points (w_i, q_i) in $\mathbb{R}^l \times \mathbb{R}$ (see Figure B.1). The *geometric regular subdivision* induced by the height vector q is the polyhedral complex defined as the projection onto $\mathbb{R}^l$ of the lower faces of the polyhedron E (i.e., the faces of E that a have a normal vector of the form $(\lambda, -1)$). A lower face F of E can be represented by the set of indices of columns of W defining F, that is $I_F = \{i \in [m] \mid (w_i, q_i) \in F\}$. For example, if (w_i, q_i) is an extreme point of F then $i \in I_F$. The collection $\mathcal{I}(W^\top, q)$ of the sets I_F, for F describing the lower faces of E, is known as a *combinatorial* regular subdivision. More details can be found in [15, Chapter 2.5].

These notions are formalized by the following definition.

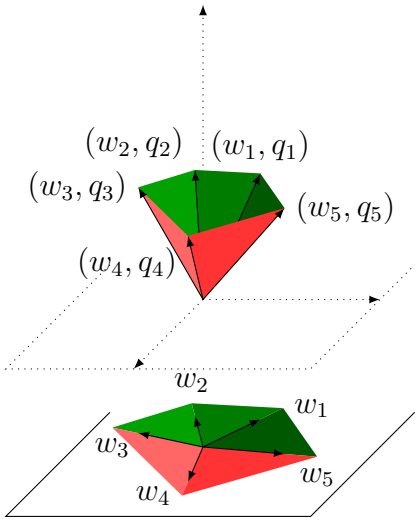
Definition B.1. (Regular subdivisions) Consider a matrix $W \in \mathbb{R}^{l\times m}$. Let us denote by $(w_i)_{i\in[m]}$ the columns of the matrix W, and let $q \in \mathbb{R}^m$. The (combinatorial) *regular subdivision* of the configuration of vectors $(w_i)_{i\in[m]}$ induced by the height vector q is the collection $\mathcal{I}(W^\top, q)$ of subsets of $[m]$ such that

$$\mathcal{I}(W^\top, q) := \left\{ I \subset [m] \;\middle|\; \exists \lambda \in \mathbb{R}^l, \quad \begin{array}{ll} w_i^\top \lambda = q_i, & \forall i \in I \\ w_j^\top \lambda < q_j, & \forall j \notin I \end{array} \right\}. \tag{72}$$

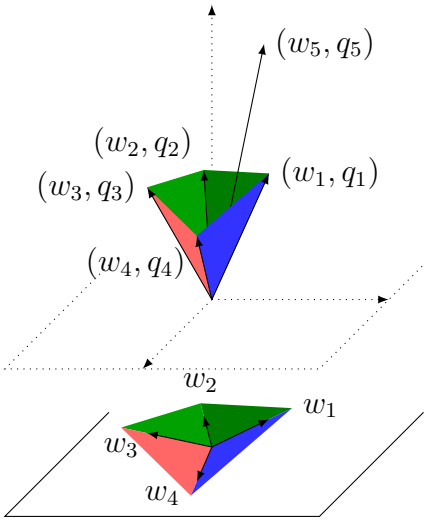
We define $\sim_W$ the equivalence relation on $\mathbb{R}^m$ such that $q \sim_W q'$ iff $\mathcal{I}(W^\top, q) = \mathcal{I}(W^\top, q')$.

We now give an equivalent characterization of the partition $\mathcal{S}_W$ defined in Eq. (29), see Chapter 5 and in particular, Theorem 5.2.16 of [15] for a proof of this equivalence.

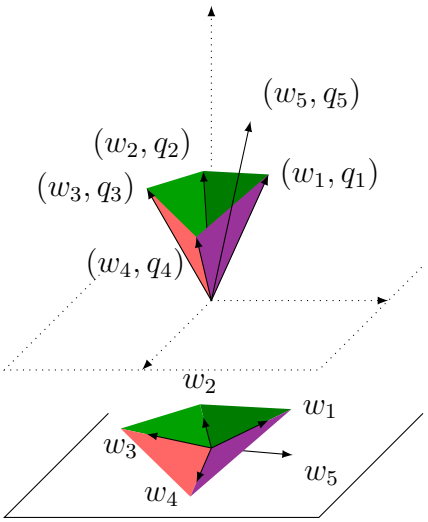
Proposition B.2. *The partition $\mathcal{S}_W$ corresponds to the collection of equivalence classes of the relation $\sim_W$.*



(a) For small q_5, the lifted vector (w_5, q_5) is salient and belongs to three lower faces:
$\mathcal{I}(W^\top, q) = \mathcal{I}_{com} \cup \{\{5\}, \{4,5\}, \{1,5\}\}$.



(b) For large q_5, the lifted vector (w_5, q_5) is pointed inward, and belongs to no lower face:
$\mathcal{I}(W^\top, q) = \mathcal{I}_{com} \cup \{\{1,4\}\}$.



(c) In the limit case, the lifted vector (w_5, q_5) belongs to one lower face:
$\mathcal{I}(W^\top, q) = \mathcal{I}_{com} \cup \{\{1,4,5\}\}$.

Figure B.1: Three lifted vector configurations, their projections and the regular subdivisions $\mathcal{I}(W^\top, q)$ induced for different values of ω_5. We define

$$\mathcal{I}_{com} := \Big\{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1,2\}, \{2,3\}, \{3,4\}\Big\}$$

We now prove that $q \mapsto \mathcal{R}_{x,W,q}$ is constant on each $S \in \mathcal{S}_W$.

Proof of Lemma 3.4. By [15, Theorem 9.5.6], for every $q \in \mathbb{R}^n$, we have

$$\mathcal{N}(D_{W,q}) = \{W_I \mathbb{R}_+^I \mid I \in \mathcal{I}(W^\top, q)\}. \tag{73}$$

Thus, for $q, q' \in S$, we have $\mathcal{N}(D_{W,q}) = \mathcal{N}(D_{W,q'})$. Further, as, by definition (27b), $\mathcal{R}_{x,W,q} := \{E_{N,x} \mid N \in \mathcal{N}(D_{W,q})\}$, we get $\mathcal{R}_{x,W,q} = \mathcal{R}_{x,W,q'}$. □

B.2. Proof of basis decomposition theorem

The goal of this appendix is to prove Theorem 3.5. We start by recalling some usual definitions and results in linear programming's theory that can be found in any standard linear programming book, e.g. [45].

Definition B.3. (Basic point, reduced cost) We consider the linear problem

$$Q(x, W, q, T, h) := \min_{y \in \mathbb{R}^m_+} \Big\{ q^\top y \mid Tx + Wy = h \Big\}. \tag{74}$$

We say that $B \subset [m]$ is a *basis* if the submatrix $W_B = (w_i)_{i \in B}$, where w_i is the i-th column of W, is invertible. The *basic point* associated to basis B is the vector in $\mathbb{R}^m$, with coordinates $y_B := (W_B^{-1}(h - Tx))_{i \in B}$ and 0 for $i \notin B$. A base B is said to be *admissible* (resp. *optimal*) if its associated basic point is an admissible (resp. optimal), solution of Eq. (74). Finally, the *reduced cost* associated to a basis B is the vector

$$(q_j - w_j^\top W_B^{-1\top} q_B)_{j \in [m]}. \tag{75}$$

Reduced cost is a key ingredient of the simplex method. In particular, it is well known that an admissible basis is optimal if and only if its reduced cost is non-negative. More formally, we have:

Lemma B.4. *Let $B \subset [m]$ be a basis. If $y_B := (W_B^{-1}(h - Tx))_{i \in B} \geq 0$ and for all $j \in [m]$, $q_j - w_j^\top W_B^{-1\top} q_B \geq 0$. Then, B is optimal and in particular,*

$$Q(x, W, q, T, h) = q_B^\top W_B^{-1}(h - Tx). \tag{76}$$

Finally, we recall a classic generalization of Caratheodory's theorem for conic hulls.

Lemma B.5. (Caratheodory) *Let $W \in \mathbb{R}^{l \times m}$ be a matrix and a subset of indices $J \subset [m]$ such that span$((w_i)_{i \in J}) := W_J \mathbb{R}^J = \mathbb{R}^l$ where w_i is the i-th column of W. Consider a vector h in the conic hull of $(w_i)_{i \in J}$, i.e. $h \in W_J \mathbb{R}^J_+$. Then, there exists a basis $B \subset J$ such that h is in the conic hull of $(w_i)_{i \in B}$, i.e. $h \in W_B \mathbb{R}^B_+$.*

Proof. Let $I \subset J$ be such that $h \in W_I \mathbb{R}^I_+$ and $(w_i)_{i \in I}$ is spanning $\mathbb{R}^l$. There exist $(\mu_i)_{i \in I} \in \mathbb{R}^I_+$ non negative coefficients such that $h = \sum_{i=1}^m \mu_i w_i$. Assume that I is not a basis, then $(w_i)_{i \in I}$ is not linearly independent, that is there exists a collection $(\lambda_i)_{i \in I} \in \mathbb{R}^I$ such that $\sum_{i \in I} \lambda_i w_i = 0$ with at least one λ_i different of zero, that can be assumed w.l.o.g positive.

Define $j := \arg\min_{i \in I | \lambda_i > 0} \frac{\mu_i}{\lambda_i}$. Then, we have $w_j = -\sum_{i \in I \setminus \{j\}} \frac{\lambda_i}{\lambda_j} w_i$ and thus we have $h = \sum_{i \in I \setminus \{j\}} (\mu_i - \mu_j \frac{\lambda_i}{\lambda_j}) w_i$. In particular, $(w_i)_{i \in I \setminus \{j\}}$ is spanning $\mathbb{R}^l$. We now show that each coefficient in this sum is non-negative, that is $h \in W_{I \setminus \{j\}} \mathbb{R}^{I \setminus \{j\}}_+$. Note that $\lambda_j > 0$ and for all $i \in I$, $\mu_i \geq 0$. Thus, if $\lambda_i \leq 0$ we have $\mu_i - \mu_j \frac{\lambda_i}{\lambda_j} \geq 0$. Otherwise, $\lambda_i > 0$, and by definition of j, $\frac{\mu_i}{\lambda_i} \geq \frac{\mu_j}{\lambda_j}$ and thus $\mu_i - \mu_j \frac{\lambda_i}{\lambda_j} \geq 0$. Which shows that $h = \sum_{i \in I \setminus \{j\}} (\mu_i - \mu_j \frac{\lambda_i}{\lambda_j}) w_i \in W_{I \setminus \{j\}} \mathbb{R}^{I \setminus \{j\}}_+$.

By induction, we drop indices until we get a basis B. □

We now have all the tools required for the proof of Theorem 3.5.

Proof of Theorem 3.5 Let $S \in \mathcal{S}_W$, $R \in \mathcal{R}_{x,W,S}$ and $q \in S$. Recall that we have $\mathcal{R}_{x,W,q} = \{E_{N,x} \mid N \in \mathcal{N}(D_{W,q})\}$. Thus, there exists $N_0 \in \mathcal{N}(D_{W,q})$ such that $R = E_{N_0,x}$ and $N \in \mathcal{N}(D_{W,q})$ a full dimensional cone such that $N_0 \subset N$. Because $\mathcal{N}(D_{W,q}) = \{W_I \mathbb{R}^I_+ \,|\, I \in \mathcal{I}(W^\top, q)\}$, there exists $I \in \mathcal{I}(W^\top, q)$ such that $N = W_I \mathbb{R}^I_+$. Since N is full dimensional, $(w_i)_{i\in I}$ is spanning $\mathbb{R}^l$. Finally, by definition of the regular subdivision $\mathcal{I}(W^\top, q)$ in (72), there exists $\lambda(I)$ such that

$$\forall i \in I, \quad w_i^\top \lambda(I) = q_i, \tag{77a}$$

$$\forall j \notin I, \quad w_j^\top \lambda(I) < q_j. \tag{77b}$$

Let $(T, h) \in E_{N_0,x} = \{(T', h') \mid h' - T'x \in \mathrm{ri}(N)\}$. In consequence we have that $h - Tx \in N_0 \subset N = W_I \mathbb{R}^I_+$. By Caratheodory's Lemma B.5, as $(w_i)_{i\in I}$ is spanning $\mathbb{R}^l$, there exists a basis $B \subset I$, such that $h - Tx \in W_B \mathbb{R}^B_+ = W_B^\top \mathbb{R}^l_+$. In particular, we have that $y_B = W_B^{-1}(h - Tx) \geq 0$, thus B is an admissible basis. Moreover, as $B \subset I$, by (77a), we have that, for all $i \in B$, $w_i^\top \lambda(I) = q_i$ which in turn implies $\lambda(I) = W_B^{-1\top} q_B$. Thus, for all $j \in [m]$, we can compute the reduced cost coordinate $q_j - w_j^\top W_B^{-1\top} q_B = q_j - w_j^\top \lambda(I) \geq 0$, by (77a) and (77b). By Lemma B.4, B is an optimal basis, leading to

$$Q(x, W, q, T, h) = q_B^\top W_B^{-1}(h - Tx).$$

Note that the resulting formula does not depend on the choice of the extracted basis B. Indeed, let $B' \subset I$ be a basis. As for all $i \in B'$, $w_i^\top \lambda(I) = q_i$, we also have $W_{B'}^{-1\top} q_{B'} = \lambda(I) = W_B^{-1\top} q_B$. Thus,

$$Q(x, W, q, T, h) = q_B^\top W_B^{-1}(h - Tx) = q_{B'}^\top W_{B'}^{-1}(h - Tx). \qquad \square$$

B.3. Proof of explicit valid and adapted partitions

We now prove Theorem 3.7.

Proof of Theorem 3.7. We have

$$V_{\mathcal{P}_{\check{x}}}(\check{x}) := \sum_{P \in \mathcal{P}_{\check{x}}} \mathbb{P}\big[(\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \in P\big]\; Q\big(\check{x}, \mathbb{E}\big[(\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \,|\, (\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \in P\big]\big), \tag{78a}$$

$$= \sum_{W \in \mathrm{supp}(\boldsymbol{W})} \sum_{S \in \mathcal{S}_W} \sum_{R \in \mathcal{R}_{\check{x},W,S}} \mathbb{P}\big[\boldsymbol{W} = W, \boldsymbol{q} \in S, (\boldsymbol{T}, \boldsymbol{h}) \in R\big] \cdot Q\big(\check{x}, \mathbb{E}\big[(\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \,|\, \boldsymbol{W} = W, \boldsymbol{q} \in S, (\boldsymbol{T}, \boldsymbol{h}) \in R\big]\big), \tag{78b}$$

$$= \sum_{W \in \mathrm{supp}(\boldsymbol{W})} \sum_{S \in \mathcal{S}_W} \sum_{R \in \mathcal{R}_{\check{x},W,S}} \mathbb{P}\big[\boldsymbol{W} = W, \boldsymbol{q} \in S, (\boldsymbol{T}, \boldsymbol{h}) \in R\big] \cdot Q(\check{x}, W, \mathbb{E}\big[(\boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \,|\, \boldsymbol{W} = W, \boldsymbol{q} \in S, (\boldsymbol{T}, \boldsymbol{h}) \in R\big]), \tag{78c}$$

$$= \sum_{W \in \mathrm{supp}(\boldsymbol{W})} \sum_{S \in \mathcal{S}_W} \sum_{R \in \mathcal{R}_{\check{x},W,S}} \mathbb{P}\big[\boldsymbol{W} = W, \boldsymbol{q} \in S, (\boldsymbol{T}, \boldsymbol{h}) \in R\big] \cdot \mathbb{E}\big[Q(\check{x}, W, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \,|\, \boldsymbol{W} = W, \boldsymbol{q} \in S, (\boldsymbol{T}, \boldsymbol{h}) \in R\big], \tag{78d}$$

$$= \sum_{W \in \mathrm{supp}(\boldsymbol{W})} \sum_{S \in \mathcal{S}_W} \sum_{R \in \mathcal{R}_{\check{x},W,S}} \mathbb{P}\big[\boldsymbol{W} = W, \boldsymbol{q} \in S, (\boldsymbol{T}, \boldsymbol{h}) \in R\big] \cdot \mathbb{E}\big[Q(\check{x}, \boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \,|\, \boldsymbol{W} = W, \boldsymbol{q} \in S, (\boldsymbol{T}, \boldsymbol{h}) \in R\big], \tag{78e}$$

$$= \mathbb{E}\big[Q(\check{x}, \boldsymbol{W}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h})\big] = V(\check{x}). \tag{78f}$$

Eq. (78a) comes from the definition of the partitioned expected cost-to-go function $V_{\mathcal{P}_{\check{x}}}$ (see (19)), and Eq. (78b) from the definition of $\mathcal{P}_{\check{x}}$. The equality (78b)=(78c) is simply the abuse of notation presented in (20). Conditioned by $\boldsymbol{W} = W$, we can use Eq. (32) to obtain (78c)=(78d). Finally, the law of total expectation yields (78e)=(78f).

We now prove that $V_{\mathcal{P}_{\check{x}}} \leq V$. For all $W \in \operatorname{supp}(\boldsymbol{W})$ and $S \in \mathcal{S}_W$, we denote by $\mathbb{E}_{W,S}$ (resp $\mathbb{P}_{W,S}$) the expectation (resp. the probability) conditional to the event $\{\boldsymbol{W} = W, \boldsymbol{q} \in S\}$. By the law of total expectation, we have

$$\begin{aligned} V_{\mathcal{P}_{\check{x}}}(x) = &\sum_{W \in \operatorname{supp}(\boldsymbol{W})} \sum_{S \in \mathcal{S}_W} \mathbb{P}\big[\boldsymbol{W} = W, \boldsymbol{q} \in S\big] \\ &\cdot \sum_{R \in \mathcal{R}_{\check{x},W,S}} \mathbb{P}_{W,S}\big[\boldsymbol{\xi} \in R\big]\; Q(x, W, \mathbb{E}_{W,S}\big[(\boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \mid (\boldsymbol{T}, \boldsymbol{h}) \in R\big]. \end{aligned} \tag{79}$$

Now by the independence of $\boldsymbol{q}$ and $(\boldsymbol{T}, \boldsymbol{h})$

$$\begin{aligned} &Q(x, W, \mathbb{E}_{W,S}\big[(\boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \mid (\boldsymbol{T}, \boldsymbol{h}) \in R\big]) \\ &= Q(x, W, \mathbb{E}_{W,S}\big[\boldsymbol{q}\big], \mathbb{E}_{W,S}\big[(\boldsymbol{T}, \boldsymbol{h}) \mid (\boldsymbol{T}, \boldsymbol{h}) \in R\big]). \end{aligned} \tag{80}$$

By the convexity of $(T, h) \mapsto Q(x, W, q, T, h)$ and Jensen's inequality, we have that

$$\begin{aligned} &Q(x, W, \mathbb{E}_{W,S}\big[\boldsymbol{q}\big], \mathbb{E}_{W,S}\big[(\boldsymbol{T}, \boldsymbol{h}) \mid (\boldsymbol{T}, \boldsymbol{h}) \in R\big]) \\ &\leq \mathbb{E}_{W,S}\big[Q(x, W, \mathbb{E}_{W,S}\big[\boldsymbol{q}\big], \boldsymbol{T}, \boldsymbol{h}) \mid (\boldsymbol{T}, \boldsymbol{h}) \in R)\big]. \end{aligned} \tag{81}$$

Now, for an event A, note that we have, by applying the law of total expectation and Lemma 3.6 twice, and with the abuse of notation Eq. (20),

$$\mathbb{E}_{W,S}\big[Q(x, W, \mathbb{E}_{W,S}\big[\boldsymbol{q}\big], \boldsymbol{T}, \boldsymbol{h}) | A\big], \tag{82a}$$

$$= \sum_{R \in \mathcal{R}_{x,W,S}} \mathbb{E}_{W,S}\big[\mathbb{1}_{(\boldsymbol{T},\boldsymbol{h}) \in R}\; Q(x, W, \mathbb{E}_{W,S}\big[\boldsymbol{q}\big], \boldsymbol{T}, \boldsymbol{h}) | A\big] \tag{82b}$$

$$\begin{aligned} = &\sum_{R \in \mathcal{R}_{x,W,S}} \mathbb{P}_{W,S}\big[(\boldsymbol{T}, \boldsymbol{h}) \in R\big] \\ &\cdot \mathbb{E}_{W,S}\big[Q(x, W, \mathbb{E}_{W,S}\big[\boldsymbol{q}\big], \boldsymbol{T}, \boldsymbol{h}) \mid A \cap (\boldsymbol{T}, \boldsymbol{h}) \in R\big] \end{aligned} \tag{82c}$$

$$\begin{aligned} = &\sum_{R \in \mathcal{R}_{x,W,S}} \mathbb{P}_{W,S}\big[(\boldsymbol{T}, \boldsymbol{h}) \in R\big] \\ &\cdot Q\left(x, \mathbb{E}_{W,S}\big[(W, \mathbb{E}_{W,S}\big[\boldsymbol{q}\big], \boldsymbol{T}, \boldsymbol{h}) \mid A \cap (\boldsymbol{T}, \boldsymbol{h}) \in R\big]\right) \text{ (by Lemma 3.6),} \end{aligned} \tag{82d}$$

$$\begin{aligned} = &\sum_{R \in \mathcal{R}_{x,W,S}} \mathbb{P}_{W,S}\big[(\boldsymbol{T}, \boldsymbol{h}) \in R\big] Q\big(x, \mathbb{E}_{W,S}\big[(W, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \mid A \cap (\boldsymbol{T}, \boldsymbol{h}) \in R\big]\big) \\ &\text{(by Eq. (20))} \end{aligned} \tag{82e}$$

$$\begin{aligned} = &\sum_{R \in \mathcal{R}_{x,W,S}} \mathbb{P}_{W,S}\big[(\boldsymbol{T}, \boldsymbol{h}) \in R\big] \mathbb{E}_{W,S}\big[Q(x, W, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{h}) \mid A \cap (\boldsymbol{T}, \boldsymbol{h}) \in R\big] \\ &\text{(by Lemma 3.6),} \end{aligned} \tag{82f}$$

$$= \sum_{R\in\mathcal{R}_{x,W,S}} \mathbb{P}_{W,S}\big[(\boldsymbol{T},\boldsymbol{h})\in R\big]\mathbb{E}_{W,S}\big[Q(x,\boldsymbol{W},\boldsymbol{q},\boldsymbol{T},\boldsymbol{h}) \mid A\cap(\boldsymbol{T},\boldsymbol{h})\in R\big] \quad (\text{under } \boldsymbol{W}=W) \tag{82g}$$

$$= \mathbb{E}_{W,S}\big[Q(x,\boldsymbol{W},\boldsymbol{q},\boldsymbol{T},\boldsymbol{h})|A\big]. \tag{82h}$$

By replacing A by $(\boldsymbol{T},\boldsymbol{h})\in R$, for $R\in\mathcal{R}_{\check{x},W,S}$ to Eq. (82h), we have

$$\begin{aligned}&\mathbb{E}_{W,S}\big[Q(x,W,\mathbb{E}_{W,S}\big[\boldsymbol{q}\big],\boldsymbol{T},\boldsymbol{h}) \mid (\boldsymbol{T},\boldsymbol{h})\in R\big]\\ &= \mathbb{E}_{W,S}\big[Q(x,\boldsymbol{W},\boldsymbol{q},\boldsymbol{T},\boldsymbol{h}) \mid (\boldsymbol{T},\boldsymbol{h})\in R\big].\end{aligned} \tag{83}$$

Combining(79),(80) and (81), we now get

$$\begin{aligned}V_{\mathcal{P}_{\check{x}}}(x) \le &\sum_{W\in\mathrm{supp}(\boldsymbol{W})}\sum_{S\in\mathcal{S}_W}\mathbb{P}\big[\boldsymbol{W}=W,\boldsymbol{q}\in S\big]\\ &\cdot\sum_{R\in\mathcal{R}_{\check{x},W,S}}\mathbb{P}_{W,S}\big[\boldsymbol{\xi}\in R\big]\mathbb{E}_{W,S}\big[Q(x,\boldsymbol{W},\boldsymbol{q},\boldsymbol{T},\boldsymbol{h}) \mid (\boldsymbol{T},\boldsymbol{h})\in R\big].\end{aligned} \tag{84}$$

By the law of total expectation, we see that the right term is equal to $V(x)$. Thus, $V_{\mathcal{P}_{\check{x}}}\le V(x)$. □

C. Sufficient conditions for effective iterations

In this appendix, we want to prove Lemma 4.5. We start with a technical lemma linking the gap at $t-1$ with the expected gap for tentative points at t.

Lemma C.1. *Let Assumptions 1 – 5 be satisfied and $t\in[T-1]$, assume that $\delta_t\in[0,D_t]$ and $\eta_t\in\mathbb{R}_+$ are given and ε_t defined by (36). Then, for all algorithms satisfying the framework of Algorithm 1, we have for $t\in[T-1]$*

$$0\le \overline{V}_{t-1}^k(x_{t-1}^k)-\underline{V}_{t-1}^k(x_{t-1}^k)\le \mathbb{E}\big[\overline{V}_t^{k-1}\big(\boldsymbol{y}_t^k\big)-\underline{V}_t^{k-1}\big(\boldsymbol{y}_t^k\big)\big]+\underline{\gamma}_{t-1}+\bar{\gamma}_{t-1}+\gamma_t^F, \tag{85}$$

where we recall that $\boldsymbol{y}_t^k := \mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k,\boldsymbol{\xi}_t)$.

Proof.

$$\underline{V}_{t-1}^k(x_{t-1}^k)\ge \mathcal{B}_{t-1}^k(\underline{V}_t^k)(x_{t-1}^k)-\underline{\gamma}_{t-1} \quad (\text{backward phase: } \underline{\gamma}_{t-1}\text{-tight cut}) \tag{86a}$$

$$= \mathbb{E}\big[\min_{x\in\mathcal{X}_t(x_{t-1}^k,\boldsymbol{\xi}_t)}\ell_t(x,\boldsymbol{\xi}_t)+\underline{V}_t^k(x)\big]-\underline{\gamma}_{t-1} \quad (\text{definition of } \mathcal{B}_t) \tag{86b}$$

$$\ge \mathbb{E}\big[\min_{x\in\mathcal{X}_t(x_{t-1}^k,\boldsymbol{\xi}_t)}\ell_t(x,\boldsymbol{\xi}_t)+\underline{V}_t^{k-1}(x)\big]-\underline{\gamma}_{t-1} \quad (\text{monotonicity of approx.}) \tag{86c}$$

$$\ge \mathbb{E}\big[\ell_t\big(\boldsymbol{y}_t^k,\boldsymbol{\xi}_t\big)+\underline{V}_t^{k-1}\big(\boldsymbol{y}_t^k\big)\big]-\gamma_t^F-\underline{\gamma}_{t-1} \quad (\text{definition of } \mathcal{F}_{t-1}). \tag{86d}$$

$$\overline{V}_{t-1}^k(x_{t-1}^k)\le \mathcal{B}_{t-1}^k(\overline{V}_t^k)(x_{t-1}^k)+\bar{\gamma}_{t-1} \quad (\text{backward phase}) \tag{87a}$$

$$= \mathbb{E}\big[\min_{x\in\mathcal{X}_t(x_{t-1}^k,\boldsymbol{\xi}_t)}\ell_t(x,\boldsymbol{\xi}_t)+\overline{V}_t^k(x)\big]+\bar{\gamma}_{t-1} \quad (\text{definition of } \mathcal{B}_t) \tag{87b}$$

$$\le \mathbb{E}\big[\min_{x\in\mathcal{X}_t(x_{t-1}^k,\boldsymbol{\xi}_t)}\ell_t(x,\boldsymbol{\xi}_t)+\overline{V}_t^{k-1}(x)\big]+\bar{\gamma}_{t-1} \quad (\text{monotonicity of approx.}) \tag{87c}$$

$$\le \mathbb{E}\big[\ell_t\big(\boldsymbol{y}_t^k,\boldsymbol{\xi}_t\big)+\overline{V}_t^{k-1}\big(\boldsymbol{y}_t^k\big)\big]+\bar{\gamma}_{t-1} \quad (\text{as } \boldsymbol{y}_t^k\in\mathcal{X}_t(x_{t-1}^k,\boldsymbol{\xi}_t)\ \ \mathbb{P}\text{-a.s.}). \tag{87d}$$

Combining these two results we get Eq. (85). □

Proof of Lemma 4.5. Let $t \in [T-1]$. We first prove that if one of the inequalities Eqs. (44a) and (44b) is satisfied then, x_{t-1}^k is ε_{t-1}-saturated as soon as x_t^k is not δ_t-distinguishable. Recall that $d_t^k(x) := \min_{\kappa<k | x_t^\kappa \text{ is } \varepsilon_t\text{-saturated}} \|x - x_t^\kappa\|$.

Assume now that x_{t+1}^k is not δ_{t+1}-distinguishable, then $d_t^k(x_t^k) \leq \delta_t$ and there exists $j < k$ such that x_t^j is ε_t-satured and $\|x_t^j - x_t^k\| \leq \delta_t$. If Eq. (44a) is satisfied, we have

$$
\begin{aligned}
&\mathbb{E}\big[\overline{V}_t^{k-1}(\boldsymbol{y}_t^k) - \underline{V}_t^{k-1}(\boldsymbol{y}_t^k)\big] \\
&\leq \overline{V}_t^{k-1}(x_t^k) - \underline{V}_t^{k-1}(x_t^k) + (\bar{L}_t + \underline{L}_t)\eta_t && \text{(Eq. (44a))} && \text{(88a)}\\
&\leq \overline{V}_t^{k-1}(x_t^k) - \overline{V}_t^{k-1}(x_t^j) + \overline{V}_t^{k-1}(x_t^j) - \underline{V}_t^{k-1}(x_t^j) + \underline{V}_t^{k-1}(x_t^j) \\
&\quad - \underline{V}_t^{k-1}(x_t^k) + (\bar{L}_t + \underline{L}_t)\eta_t, && && \text{(88b)}\\
&\leq \overline{L}_t\|x_t^k - x_t^j\| + \overline{V}_t^{k-1}(x_t^j) - \underline{V}_t^{k-1}(x_t^j) + \underline{L}_t\|x_t^j - x_t^k\| + (\bar{L}_t + \underline{L}_t)\eta_t \\
& && \text{(Lipschitz)} && \text{(88c)}\\
&\leq (\overline{L}_t + \underline{L}_t)\delta_t + \overline{V}_t^j(x_t^j) - \underline{V}_t^j(x_t^j) + (\bar{L}_t + \underline{L}_t)\eta_t && \text{(monotonicity)} && \text{(88d)}\\
&\leq (\overline{L}_t + \underline{L}_t)(\delta_t + \eta_t) + \varepsilon_t && (\varepsilon_t\text{-saturation}). && \text{(88e)}
\end{aligned}
$$

Similarly, if Eq. (44b) is satisfied, we define $j(\xi)$ such that

$$
j(\xi) \in \operatorname*{arg\,min}_{j \leq k-1, x_t^j \varepsilon_t\text{-saturated}} \|x_t^j - \mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \xi)\|. \tag{89}
$$

In particular, $d_t^k\big(\mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \xi)\big) = \|x_t^{j(\xi)} - \mathcal{F}_{t-1}(\underline{V}_t^{k-1})(x_{t-1}^k, \xi)\|$ and therefore $\mathbb{E}\big[d_t^k(\boldsymbol{y}_t^k)\big] = \mathbb{E}\big[\|x_t^{j(\boldsymbol{\xi})} - \boldsymbol{y}_t^k\|\big]$ and

$$
\begin{aligned}
&\mathbb{E}\big[\overline{V}_t^{k-1}(\boldsymbol{y}_t^k) - \underline{V}_t^{k-1}(\boldsymbol{y}_t^k)\big] \\
&\leq \mathbb{E}\big[\overline{V}_t^{k-1}(\boldsymbol{y}_t^k) - \overline{V}_t^{k-1}(x_t^{j(\boldsymbol{\xi}_t)}) + \overline{V}_t^{k-1}(x_t^{j(\boldsymbol{\xi}_t)}) - \underline{V}_t^{k-1}(x_t^{j(\boldsymbol{\xi}_t)}) + \underline{V}_t^{k-1}(x_t^{j(\boldsymbol{\xi}_t)}) - \underline{V}_t^{k-1}(\boldsymbol{y}_t^k)\big] \\
&\leq \mathbb{E}\big[\bar{L}_t\|\boldsymbol{y}_t^k - x_t^{j(\boldsymbol{\xi}_t)}\| + \varepsilon_t + \underline{L}_t\|x_t^{j(\boldsymbol{\xi}_t)} - \boldsymbol{y}_t^k\|\big] \\
&= (\bar{L}_t + \underline{L}_t)\mathbb{E}\big[d_t^k(\boldsymbol{y}_t^k)\big] + \varepsilon_t \leq (\bar{L}_t + \underline{L}_t)(\delta_t + \eta_t) + \varepsilon_t.
\end{aligned}
$$

Then, in both cases, $\mathbb{E}\big[\overline{V}_t^{k-1}(\boldsymbol{y}_t^k) - \underline{V}_t^{k-1}(\boldsymbol{y}_t^k)\big] \leq (\bar{L}_t + \underline{L}_t)(\delta_t + \eta_t) + \varepsilon_t$. By Lemma C.1, we have

$$
\begin{aligned}
&\overline{V}_{t-1}^k(x_{t-1}^k) - \underline{V}_{t-1}^k(x_{t-1}^k) \leq \mathbb{E}\big[\overline{V}_t^{k-1}(\boldsymbol{y}_t^k) - \underline{V}_t^{k-1}(\boldsymbol{y}_t^k)\big] + \underline{\gamma}_{t-1} + \bar{\gamma}_{t-1} + \gamma_t^F && \text{(90a)}\\
&\leq (\bar{L}_t + \underline{L}_t)(\delta_t + \eta_t) + \varepsilon_t + \underline{\gamma}_{t-1} + \bar{\gamma}_{t-1} + \gamma_t^F = \varepsilon_{t-1}. && \text{(90b)}
\end{aligned}
$$

Thus, x_t^k is ε_t-saturated as soon as x_{t+1}^k is not δ_{t+1}-distinguishable.

We now prove by backward induction on t that iteration k is effective. We first prove that, for all $k \in \mathbb{N}^\star$, x_{T-1}^k is ε_{T-1}-saturated.

$$
\begin{aligned}
&\overline{V}_{T-1}(x_{T-1}^k) - \underline{V}_{T-1}(x_{T-1}^k) \\
&\leq \mathcal{B}_{T-1}(\overline{V}_T^k)(x_{T-1}^k) + \bar{\gamma}_{T-1} - \mathcal{B}_{T-1}(\underline{V}_T^k)(x_{T-1}^k) + \underline{\gamma}_{T-1} && \text{(91a)}
\end{aligned}
$$

$$= \mathcal{B}_{T-1}(0)(x^k_{T-1}) + \bar{\gamma}_{T-1} - \mathcal{B}_{T-1}(0)(x^k_{T-1}) + \underline{\gamma}_{T-1} \tag{91b}$$

$$= \bar{\gamma}_{T-1} + \underline{\gamma}_{T-1} = \varepsilon_{T-1}. \tag{91c}$$

Let $t \geq 2$ such that, for every $\tau \geq t$, x^k_τ is ε_τ-saturated. If x^k_t is δ_t-distinguishable, then iteration k is effective. Otherwise, x^k_t is not δ_t-distinguishable and by the previous paragraph, it implies that x^k_{t-1} is ε_{t-1}-saturated. Eventually, assuming that x^k_1 is ε_1-saturated. If x^k_1 is δ_1-distinguishable, then iteration k is effective. Otherwise, there exists $j < k$ such that $\|x^j_1 - x^k_1\| \leq \delta_1$ and x^j_1 is ε_1 saturated. We get

$$V_1(x^k_1) \leq \overline{V}^j_1(x^k_1) = \overline{V}^j_1(x^k_1) - \overline{V}^j_1(x^j_1) + \overline{V}^j_1(x^j_1), \tag{92a}$$

$$\leq \bar{L}_1\|x^k_1 - x^j_1\| + \overline{V}^j_1(x^j_1) \leq \bar{L}_1\delta_1 + \varepsilon_1 + \underline{V}^j_1(x^j_1) \tag{92b}$$

$$= \bar{L}_1\delta_1 + \varepsilon_1 + \underline{V}^j_1(x^j_1) - \underline{V}^j_1(x^k_1) + \underline{V}^j_1(x^k_1) \tag{92c}$$

$$\leq \bar{L}_1\delta_1 + \varepsilon_1 + \underline{L}_1\|x^k_1 - x^j_1\| + \underline{V}^{k-1}_1(x^k_1) \tag{92d}$$

$$\leq (\bar{L}_1 + \underline{L}_1)\delta_1 + \varepsilon_1 + \underline{V}^{k-1}_1(x^k_1). \tag{92e}$$

Then,

$$\ell_1(x^k_1, \xi_1) + V_1(x^k_1) \leq (\bar{L}_1 + \underline{L}_1)\delta_1 + \varepsilon_1 + \ell_1(x^k_1, \xi_1) + \underline{V}^{k-1}_1(x^k_1) \tag{93a}$$

$$\leq (\bar{L}_1 + \underline{L}_1)\delta_1 + \varepsilon_1 + \gamma^F_1 + \min_{x_1 \in X_1(x_0)} \ell_1(x_1, \xi_1) + \underline{V}^{k-1}_1(x_1) \tag{93b}$$

$$\leq \varepsilon_0 + \min_{x_1 \in X_1(x_0)} \ell_1(x_1, \xi_1) + V_1(x_1). \tag{93c}$$

Thus, in all the covered cases, iteration k is effective. □

D. Probabilistic lemmas

In this appendix, we present useful probabilistic lemmas to prove the convergence of SDDP with randomized choice of $\boldsymbol{\xi}^k_t$.

D.1. A nested Hoeffding lemma

Lemma D.1. *Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space, $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two independent random variables taking values respectively in the euclidean spaces $\mathbb{X}$ and $\mathbb{Y}$.*

Let $r > 0$ be a positive real and $f : \mathbb{X} \times \mathbb{Y} \mapsto \mathbb{R}$ be a measurable function such that $0 \leq f(\boldsymbol{X}, \boldsymbol{Y}) \leq r$ almost surely.

Then for every $\eta > 0$ and $A \in \sigma(\boldsymbol{X})$ such that $\mathbb{P}\big[A\big] > 0$, we have

$$\mathbb{P}\Big[f(\boldsymbol{X}, \boldsymbol{Y}) > \mathbb{E}\big[f(\boldsymbol{X}, \boldsymbol{Y})|\sigma(X)\big] - \eta \,\Big|\, A\Big] \geq 1 - e^{\frac{-2\eta^2}{r^2}}. \tag{94}$$

Proof. Recall that the Hoeffding lemma states that if $\boldsymbol{Z}$ is a real random variable such that there exists $a, b \in \mathbb{R}$ with $a \leq \boldsymbol{Z} \leq b$ almost surely then for every $\eta > 0$ we have

$$\mathbb{P}\Big[\boldsymbol{Z} - \mathbb{E}\big[\boldsymbol{Z}\big] \leq -\eta\Big] \leq e^{\frac{-2\eta^2}{(b-a)^2}}. \tag{95}$$

By taking the complementary event, we have

$$\mathbb{P}\Big[\boldsymbol{Z} > \mathbb{E}\big[\boldsymbol{Z}\big] - \eta\Big] \geq 1 - e^{\frac{-2\eta^2}{(b-a)^2}}. \tag{96}$$

Then for every $x \in \mathbb{X}$, by applying the Hoeffding lemma to the random variable $\boldsymbol{Z} = f(x, \boldsymbol{Y})$, $a = 0$ and $b = r$, we have

$$\mathbb{P}\big[f(x, \boldsymbol{Y}) > \mathbb{E}\big[f(x, \boldsymbol{Y})\big] - \eta\big] \geq 1 - e^{\frac{-2\eta^2}{r^2}}. \tag{97}$$

Let $A \in \sigma(\boldsymbol{X})$ and $B \subset \mathbb{X}$ such that $A = \boldsymbol{X}^{-1}(B)$

$$\begin{aligned}
&\mathbb{P}\Big[\{f(\boldsymbol{X}, \boldsymbol{Y}) > \mathbb{E}\big[f(\boldsymbol{X}, \boldsymbol{Y})|\sigma(\boldsymbol{X})\big] - \eta\} \cap A\Big] \\
&= \int_\Omega \mathbb{1}_{\left\{f(\boldsymbol{X}(\omega), \boldsymbol{Y}(\omega)) > \mathbb{E}[f(\boldsymbol{X}, \boldsymbol{Y})|\sigma(\boldsymbol{X})](\omega) - \eta\right\}} \mathbb{1}_{\omega \in A} \mathbb{P}(d\omega) && \text{(98a)} \\
&= \int_\Omega \mathbb{1}_{\{f(\boldsymbol{X}(\omega), \boldsymbol{Y}(\omega)) > \mathbb{E}_{\boldsymbol{Y}}[f(\boldsymbol{X}(\omega), \boldsymbol{Y})] - \eta\}}(\omega) \mathbb{1}_{X(\omega) \in B} \mathbb{P}(d\omega) && \text{(98b)} \\
&= \int_{\mathbb{X}} \int_{\mathbb{Y}} \mathbb{1}_{f(x,y) > \mathbb{E}[f(x, \boldsymbol{Y})] - \eta} \mathbb{1}_{x \in B} \mathbb{P}_{\boldsymbol{Y}}(dy) \mathbb{P}_{\boldsymbol{X}}(dx), && \text{(98c)} \\
&= \int_{\mathbb{X}} \mathbb{1}_{x \in B} \Big(\int_{\mathbb{Y}} \mathbb{1}_{f(x,y) > \mathbb{E}[f(x, \boldsymbol{Y})] - \eta} \mathbb{P}_{\boldsymbol{Y}}(dy) \Big) \mathbb{P}_{\boldsymbol{X}}(dx) && \text{(98d)} \\
&= \int_{\mathbb{X}} \mathbb{1}_{x \in B} \mathbb{P}\Big[f(x, \boldsymbol{Y}) > \mathbb{E}\big[f(x, \boldsymbol{Y})\big] - \eta\Big] \mathbb{P}_{\boldsymbol{X}}(dx) && \text{(98e)} \\
&\geq \int_{\mathbb{X}} \mathbb{1}_{x \in B} (1 - e^{\frac{-2\eta^2}{r^2}}) \mathbb{P}_{\boldsymbol{X}}(dx) && \text{(98f)} \\
&= (1 - e^{-\frac{-2\eta^2}{r^2}}) \mathbb{P}_{\boldsymbol{X}}\big[B\big] = (1 - e^{-\frac{-2\eta^2}{r^2}}) \mathbb{P}\big[A\big]. && \text{(98g)}
\end{aligned}$$

Thus, by dividing by $\mathbb{P}\big[A\big]$, we get

$$\mathbb{P}\Big[f(\boldsymbol{X}, \boldsymbol{Y}) > \mathbb{E}\big[f(\boldsymbol{X}, \boldsymbol{Y})|\sigma(X)\big] - \eta \,\Big|\, A\Big] \geq 1 - e^{\frac{-2\eta^2}{r^2}}. \qquad \square$$

Lemma D.2. *Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space, $(\boldsymbol{X}_n)_{n \in \mathbb{N}}$ be a sequence of independent random variables taking values in the Euclidean space $\mathbb{X}$ and $\mathcal{A}_n = \sigma(\boldsymbol{X}_k)_{k \in \mathbb{N}}$ be its adapted filtration.*

For every $n \in \mathbb{N}$, let r_n and η_n be two positive real and $f_n : \mathbb{X}^n \mapsto \mathbb{R}$ be a measurable function such that $0 \leq f_n(X_1, \dots, X_n) \leq r_n$ almost-surely.

We denote by E_n the event

$$\Big\{\omega \,\Big|\, f_n(\boldsymbol{X}_1, \dots, \boldsymbol{X}_n) > \mathbb{E}\big[f_n(\boldsymbol{X}_1, \dots, \boldsymbol{X}_n)|\mathcal{A}_{n-1}\big] - \eta_n\Big\} \in \mathcal{A}_n.$$

Then, for all $m \leq n \in \mathbb{N}$ and $A_{m-1} \in \mathcal{A}_{m-1}$ such that $\mathbb{P}\big[A_{m-1}\big] > 0$, we have

$$\mathbb{P}\left[\bigcap_{k=m}^{n} E_k \,\middle|\, A_{m-1}\right] \geq \prod_{k=m}^{n} \left(1 - e^{\frac{-2\eta_k^2}{r_k^2}}\right). \tag{99}$$

Proof. For every $n \in \mathbb{N}^\star$, $\eta > 0$ and $A_{n-1} \in \mathcal{A}_{n-1}$ such that $\mathbb{P}\big[A\big] > 0$, by the previous lemma applied to $X = (X_1, \dots, X_{n-1})$, $Y = X_n$, $f = f_n$, $\eta = \eta_n$ and $r = r_n$, we have

$$\mathbb{P}\Big[E_n \,\Big|\, A_{n-1}\Big] \geq 1 - e^{\frac{-2\eta_n^2}{r_n^2}}. \tag{100}$$

Let $m \in \mathbb{N}$, we now prove our lemma by induction on n. If $n = m = 1$ the result is true by the Hoeffding lemma and for $n = m > 1$ the result is true by Eq. (100) with $A_{n-1} = A_{m-1}$.

Let $n \geq m$ and assume that $\mathbb{P}\Big[\bigcap_{k=m}^{n} E_k \cap A_{m-1}\Big] > 0$ and

$$\mathbb{P}\Big[\bigcap_{k=m}^{n} E_k \,\Big|\, A_{m-1}\Big] \geq \prod_{k=m}^{n} \left(1 - e^{\frac{-2\eta_k^2}{r_k^2}}\right).$$

Then,

$$\mathbb{P}\Big[\bigcap_{k=m}^{n+1} E_k \,\Big|\, A_{m-1}\Big] = \mathbb{P}\Big[E_{n+1} \,\Big|\, \bigcap_{k=m}^{n} E_k \cap A_{m-1}\Big]\mathbb{P}\Big[\bigcap_{k=m}^{n} E_k \,\Big|\, A_{m-1}\Big], \tag{101a}$$

$$\geq \left(1 - e^{\frac{-2\eta_{n+1}^2}{r_{n+1}^2}}\right) \prod_{k=m}^{n} \left(1 - e^{\frac{-2\eta_k^2}{r_k^2}}\right), \tag{101b}$$

where we underestimate the first factor thanks to Eq. (100) and $\bigcap_{k=m}^{n} E_k \in \mathcal{A}_n$ and the second factor thanks to the induction hypothesis.

In particular, $\mathbb{P}\Big[\bigcap_{k=m}^{n+1} E_k \cap A_{m-1}\Big] > 0$ and induction ends the proof. □

D.2. Stochastic dominance by geometric random variables

Recall that a real random variable $\boldsymbol{X}$ is (first-order) stochastically dominated by a real random variable $\boldsymbol{Y}$ if the cumulative density function of $\boldsymbol{X}$ is smaller than the cumulative density function of $\boldsymbol{Y}$. If $\boldsymbol{X}$ and $\boldsymbol{Y}$ are integer random variables, $\boldsymbol{X}$ is stochastically dominated by $\boldsymbol{Y}$ is equivalent to $\mathbb{P}\big[\boldsymbol{X} \geq n\big] \leq \mathbb{P}\big[\boldsymbol{Y} \geq n\big]$, for all $n \in \mathbb{N}^\star$. We now present a lemma where we leverage this notion to bound the number of effective iteration in randomized algorithm in Theorem 4.11.

Lemma D.3. *Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a space of probability, $(\boldsymbol{X}_n)_{n\in\mathbb{N}}$ be a sequence of independent and identically distributed random variables and $\mathcal{A}_n = \sigma(\boldsymbol{X}_k)_{k\in\mathbb{N}^\star}$ be its adapted filtration. Let $(\boldsymbol{Y}_n)_{n\in\mathbb{N}}$ be a sequence of (non necessarily independent neither identically distributed) binary random variables, i.e. taking values in $\{0, 1\}$, such that $\sigma(\boldsymbol{Y}_n) \subset \mathcal{A}_n$. Assume that there exists $p \in (0, 1)$ such that for all $n \in \mathbb{N}^\star$ and all $A_n \in \mathcal{A}_n$ such that $\mathbb{P}\big[A_n\big] > 0$, we have*

$$\mathbb{P}\big[\boldsymbol{Y}_{n+1} = 1 \,|\, A_n\big] \geq p. \tag{102}$$

For $m \in \mathbb{N}$, we define the stopping time $\boldsymbol{\tau}_m := \inf\{n \in \mathbb{N} \,|\, \sum_{k=1}^{n} \boldsymbol{Y}_i \geq m\}$. Let $B_{m,p}$ be a random variable with a negative binomial distribution representing the number of trials to obtain m successes with probability of success p, i.e.

$$\mathbb{P}\big[\boldsymbol{B}_{m,p} = n\big] = \binom{n-1}{m-1} p^m (1-p)^{n-m}, \quad \text{for all } n \geq m.$$

Then, $\boldsymbol{\tau}_m$ is stochastically dominated by $\boldsymbol{B}_{m,p}$ i.e.

$$\mathbb{P}\big[\boldsymbol{\tau}_m \geq n\big] \leq \mathbb{P}\big[\boldsymbol{B}_{m,p} \geq n\big], \qquad \forall n \in \mathbb{N}^\star. \tag{103}$$

$$\textit{In particular,} \quad \mathbb{E}\big[\boldsymbol{\tau}_m\big] \leq \mathbb{E}\big[\boldsymbol{B}_{m,p}\big] = \frac{m}{p}. \tag{104}$$

Proof. Let $(\tilde{\boldsymbol{Y}}_n)_{n\in\mathbb{N}^\star}$ a sequence of independent and identically distributed Bernoulli random variables with parameter p. For all $n \in \mathbb{N}^\star$, we define the random variables $\boldsymbol{S}_n := \sum_{k=1}^n \boldsymbol{Y}_n$ and $\tilde{\boldsymbol{S}}_n := \sum_{k=1}^n \tilde{\boldsymbol{Y}}_n$. We first show by induction on n that $\tilde{\boldsymbol{S}}_n$ is stochastically dominated by $\boldsymbol{S}_n$, i.e. for all $a \in \mathbb{N}^\star$, we have $\mathbb{P}\big[\boldsymbol{S}_n \geq a\big] \geq \mathbb{P}\big[\tilde{\boldsymbol{S}}_n \geq a\big]$. Indeed, for $n = 1$ we have $\boldsymbol{S}_1 = \boldsymbol{Y}_1$ and $\boldsymbol{S}_2 = \boldsymbol{Y}_2$ then $\mathbb{P}\big[\boldsymbol{S}_n \geq 0\big] = \mathbb{P}\big[\tilde{\boldsymbol{S}}_n \geq 0\big] = 1$ and $\mathbb{P}\big[\boldsymbol{S}_n \geq 1\big] = \mathbb{P}\big[\boldsymbol{Y}_n = 1\big] \geq p = \mathbb{P}\big[\tilde{\boldsymbol{S}}_n \geq 1\big]$. Finally, for all $a \geq 2$, $\mathbb{P}\big[\boldsymbol{S}_n \geq 2\big] = \mathbb{P}\big[\tilde{\boldsymbol{S}}_n \geq 2\big] = 0$.

We now assume that there exists $n \in \mathbb{N}^\star$ such that $\mathbb{P}\big[\boldsymbol{S}_n \geq a\big] \geq \mathbb{P}\big[\tilde{\boldsymbol{S}}_n \geq a\big]$ for all $a \in \mathbb{N}^\star$. We then have

$$\begin{aligned}
\mathbb{P}\big[\boldsymbol{S}_{n+1} \geq a\big] &= \mathbb{P}\big[\boldsymbol{S}_{n+1} \geq a, \boldsymbol{S}_n \leq a-2\big] + \mathbb{P}\big[\boldsymbol{S}_{n+1} \geq a, \boldsymbol{S}_n = a-1\big] \\
&\qquad + \mathbb{P}\big[\boldsymbol{S}_{n+1} \geq a, \boldsymbol{S}_n \geq a\big] && \text{(105a)}\\
&= 0 + \mathbb{P}\big[\boldsymbol{S}_{n+1} \geq a \,|\, \boldsymbol{S}_n = a-1\big]\mathbb{P}\big[\boldsymbol{S}_n = a-1\big] + \mathbb{P}\big[\boldsymbol{S}_n \geq a\big] && \text{(105b)}\\
&= \mathbb{P}\big[Y_{n+1} = 1 \,|\, \boldsymbol{S}_n = a-1\big]\mathbb{P}\big[\boldsymbol{S}_n = a-1\big] + \mathbb{P}\big[\boldsymbol{S}_n \geq a\big] && \text{(105c)}\\
&\geq p\mathbb{P}\big[\boldsymbol{S}_n = a-1\big] + \mathbb{P}\big[\boldsymbol{S}_n \geq a\big] \qquad \text{(by assumption (102))} && \text{(105d)}\\
&= p(\mathbb{P}\big[\boldsymbol{S}_n \geq a-1\big] - \mathbb{P}\big[\boldsymbol{S}_n \geq a\big]) + \mathbb{P}\big[\boldsymbol{S}_n \geq a\big] && \text{(105e)}\\
&= p\mathbb{P}\big[\boldsymbol{S}_n \geq a-1\big] + (1-p)\mathbb{P}\big[\boldsymbol{S}_n \geq a\big] && \text{(105f)}\\
&\geq p\mathbb{P}\big[\tilde{\boldsymbol{S}}_n \geq a-1\big] + (1-p)\mathbb{P}\big[\tilde{\boldsymbol{S}}_n \geq a\big] \qquad \text{(by induction assumption)} && \text{(105g)}\\
&= \mathbb{P}\big[\tilde{\boldsymbol{Y}}_{n+1} = 1 \,|\, \tilde{\boldsymbol{S}}_n \geq a-1\big]\mathbb{P}\big[\tilde{\boldsymbol{S}}_n \geq a-1\big] \\
&\quad + \mathbb{P}\big[\tilde{\boldsymbol{Y}}_{n+1} = 0 \,|\, \tilde{\boldsymbol{S}}_n \geq a\big]\mathbb{P}\big[\tilde{\boldsymbol{S}}_n \geq a\big] && \text{(105h)}\\
&= \mathbb{P}\big[\tilde{\boldsymbol{S}}_n \geq a-1, \tilde{\boldsymbol{Y}}_{n+1} = 1\big] + \mathbb{P}\big[\tilde{\boldsymbol{S}}_n \geq a, \tilde{\boldsymbol{Y}}_{n+1} = 0\big] && \text{(105i)}\\
&= \mathbb{P}\big[\tilde{\boldsymbol{S}}_{n+1} \geq a, \tilde{\boldsymbol{Y}}_{n+1} = 1\big] + \mathbb{P}\big[\tilde{\boldsymbol{S}}_{n+1} \geq a, \tilde{\boldsymbol{Y}}_{n+1} = 0\big] && \text{(105j)}\\
&= \mathbb{P}\big[\tilde{\boldsymbol{S}}_{n+1} \geq a\big]. && \text{(105k)}
\end{aligned}$$

Then, by induction, $\tilde{\boldsymbol{S}}_n$ is stochastically dominated by $\boldsymbol{S}_n$. For $m \in \mathbb{N}^\star$, we recall that we have $\boldsymbol{\tau}_m = \inf\{n \in \mathbb{N} \,|\, \boldsymbol{S}_n \geq m\}$. We now define $\tilde{\boldsymbol{\tau}}_m := \inf\{n \in \mathbb{N} \,|\, \tilde{\boldsymbol{S}}_n \geq m\}$ similarly. As $\tilde{\boldsymbol{S}}_n$ is stochastically dominated by $\boldsymbol{S}_n$, it is easy to see that the stopping time $\boldsymbol{\tau}_m$ is stochastically dominated by the stopping time $\tilde{\boldsymbol{\tau}}_m$. Indeed,

$$\begin{aligned}
\mathbb{P}\big[\boldsymbol{\tau}_m \geq a\big] &= \mathbb{P}\big[\boldsymbol{S}_a < m\big] = 1 - \mathbb{P}\big[\boldsymbol{S}_a \geq m+1\big] \leq 1 - \mathbb{P}\big[\tilde{\boldsymbol{S}}_a \geq m+1\big] \\
&= \mathbb{P}\big[\tilde{\boldsymbol{S}}_a < m\big] = \mathbb{P}\big[\tilde{\boldsymbol{\tau}}_m \geq a\big].
\end{aligned}$$

Finally, the random variable $\tilde{\boldsymbol{\tau}}_1$ and the random variables $\tilde{\boldsymbol{\tau}}_{k+1} - \tilde{\boldsymbol{\tau}}_k$, for all $k \in \mathbb{N}^\star$, are independent and identically distributed geometric random variables with probability of success p. Thus, $\boldsymbol{B}_{m,p} := \tilde{\boldsymbol{\tau}}_m = \tilde{\boldsymbol{\tau}}_1 + \sum_{k=1}^{m-1}(\tilde{\boldsymbol{\tau}}_{k+1} - \tilde{\boldsymbol{\tau}}_k)$ is a random

variable with negative binomial distribution representing the number of trials to obtain m successes with probability of success p and $\boldsymbol{\tau}_m$ is stochastically dominated by $\boldsymbol{B}_{m,p}$. □

References

[1] S. Ahmed, F. G. Cabral, B. F. P. da Costa: *Stochastic Lipschitz dynamic programming*, Math. Programming (2020) 1–39.

[2] M. Akian, J.-P. Chancelier, B. Tran: *Tropical dynamic programming for Lipschitz multistage stochastic programming*, arXiv: 2010.10619 (2020)

[3] P. Artzner, F. Delbaen, J.-M. Eber, D. Heath: *Coherent measures of risk*, Math. Finance 9/3 (1999) 203–228.

[4] P. Artzner, F. Delbaen, J.-M. Eber, D. Heath, H. Ku: *Coherent multiperiod risk adjusted values and Bellman's principle*, Ann. Oper. Research 152/1 (2007) 5–22.

[5] T. Asamov, W. B. Powell: *Regularized decomposition of high-dimensional multistage stochastic programs with markov uncertainty*, SIAM J. Optimization 28/1 (2018) 575–595.

[6] M. Bandarra, V. Guigues: *Single cut and multicut stochastic dual dynamic programming with cut selection for multistage stochastic linear programs: convergence proof and numerical experiments*, Comp. Manage. Sci. 18/2 (2021) 125–148.

[7] R. Baucke, A. Downward, G. Zakeri: *A deterministic algorithm for solving multistage stochastic programming problems*, Optimization Online (2017) 1–25.

[8] R. Baucke, A. Downward, G. Zakeri: *A deterministic algorithm for solving multistage stochastic minimax dynamic programmes*, Optimization Online (2018).

[9] F. Beltrán, E. C. Finardi, G. M. Fredo, W. de Oliveira: *Improving the performance of the stochastic dual dynamic programming algorithm using Chebyshev centers*, Optim. Eng. (2020) 1–22.

[10] J. R. Birge: *Decomposition and partitioning methods for multistage stochastic linear programs*, Oper. Research 33/5 (1985) 989–1007.

[11] J. R. Birge, R. J-B Wets: *Designing approximation schemes for stochastic optimization problems, in particular for stochastic programs with recourse*, Math. Program. Study 27 (1986) 54–102.

[12] M. Casey, S. Sen: *The scenario generation algorithm for multistage stochastic linear programming*, Math. Oper. Research 30/3 (2005) 615–631.

[13] S. Cerisola, J. M. Latorre, A. Ramos: *Stochastic dual dynamic programming applied to nonconvex hydrothermal models*, Eur. J. Oper. Research 218/3 (2012) 687–697.

[14] B. F. P. da Costa, V. Leclère: *Dual SDDP for risk-averse multistage stochastic programs*, Oper. Res. Letters 51/3 (2023) 332–337.

[15] J. A. De Loera, J. Rambau, F. Santos: *Triangulations Structures for Algorithms and Applications*, Springer, Berlin (2010).

[16] A. Downward, O. Dowson, R. Baucke: *Stochastic dual dynamic programming with stagewise-dependent objective uncertainty*, Oper. Research Letters 48/1 (2020) 33–39.

[17] O. Dowson: *Applying Stochastic Optimisation to the New Zealand Dairy Industry*, PhD thesis, University of Auckland (2018).

[18] O. Dowson, D. P. Morton, B. K. Pagnoncelli: *Partially observable multistage stochastic programming*, Oper. Research Letters 48/4 (2020) 505–512.

[19] J. Dupačová: *Applications of stochastic programming: Achievements and questions*, Eur. J. Oper. Research 140/2 (2002) 281–290.

[20] D. Duque, D. P. Morton: *Distributionally robust stochastic dual dynamic programming*, SIAM J. Optimization 30/4 (2020) 2841–2865.

[21] N. C. P. Edirisinghe, W. T. Ziemba: *Bounding the expectation of a saddle function with application to stochastic programming*, Math. Oper. Research 19/2 (1994) 314–340.

[22] M. Forcier: *Multistage Stochastic Optimization and Polyhedral Geometry*, PhD thesis, École des Ponts, Marne-la-Vallée (2022).

[23] M. Forcier, S. Gaubert, V. Leclère: *Exact quantization of multistage stochastic linear problems*, arXiv: 2107.09566 (2021).

[24] M. Forcier, V. Leclère: *Generalized adaptive partition-based method for two-stage stochastic linear programs: geometric oracle and analysis*, Oper. Research Letters 50/5 (2022) 552–557.

[25] K. Frauendorfer: *Barycentric scenario trees in convex multistage stochastic programming*, Math. Programming 75/2 (1996) 277–293.

[26] H. Gassmann, W. T. Ziemba: *Stochastic Programming: Applications in Finance, Energy, Planning and Logistics*, World Scientific Series in Finance Vol. 4, World Scientific, Hackensack (2013).

[27] A. Georghiou, A. Tsoukalas, W. Wiesemann: *Robust dual dynamic programming*, Oper. Research 67/3 (2019) 813–830.

[28] P. Girardeau, V. Leclère, A. B. Philpott: *On the convergence of decomposition methods for multistage stochastic convex programs*, Math. Oper. Research 40/1 (2015) 130–145.

[29] A. Gjelsvik, B. Mo, A. Haugstad: *Long-and medium-term operations planning and stochastic modelling in hydro-dominated power systems based on stochastic dual dynamic programming*, in: *Handbook of Power Systems I*, S. Rebennack et al. (eds.), Springer, Berlin (2010) 33–55.

[30] V. Guigues: *Convergence analysis of sampling-based decomposition methods for risk-averse multistage stochastic convex programs*, SIAM J. Optimization 26/4 (2016) 2468–2494.

[31] V. Guigues: *Inexact cuts in stochastic dual dynamic programming*, SIAM J. Optimization 30/1 (2020) 407–438.

[32] V. Guigues, M. A. Lejeune, W. Tekaya: *Regularized stochastic dual dynamic programming for convex nonlinear optimization problems*, Optim. Eng. 21/3 (2020) 1133–1165.

[33] G. A. Hanasusanto, D. Kuhn, W. Wiesemann: *A comment on "Computational complexity of stochastic programming problems"*, Math. Programming 159/1-2 (2016) 557–569.

[34] T. Homem-de-Mello, V. L. De Matos, E. C. Finardi: *Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling*, Energy Systems 2/1 (2011) 1–31.

[35] P. Kall, J. Mayer: *Stochastic Linear Programming*, Springer, Berlin (1976).

[36] D. Kuhn: *Generalized Bounds for Convex Multistage Stochastic Programs*, Lecture Notes in Economics and Mathematical Systems 548, Springer, Berlin (2005).

[37] G. Lan: *Complexity of stochastic dual dynamic programming*, Math. Programming A 191/2 (2020) 717–754; A 194/1-2 (2022) 1187–1189.

[38] G. Laporte, F. V. Louveaux: *The integer L-shaped method for stochastic integer programs with complete recourse*, Oper. Research Letters 13/3 (1993) 133–142.

[39] V. Leclere, P. Carpentier, J.-P. Chancelier, A. Lenoir, F. Pacaud: *Exact converging bounds for stochastic dual dynamic programming via fenchel duality*, SIAM J. Optimization 30/2 (2020) 1223–1250.

[40] N. Loehndorf: *An empirical analysis of scenario generation methods for stochastic optimization*, Eur. J. Oper. Research 255/1 (2016) 121–132.

[41] F. V. Louveaux: *A solution method for multistage stochastic programs with recourse with application to an energy investment problem*, Oper. Research 28/4 (1980) 889–902.

[42] M. E. P. Maceiral, D. D. J. Penna, A. L. Diniz, R. J. Pinto, A. C. G. Melo, C. V. Vasconcellos, C. B. Cruz: *Twenty years of application of stochastic dual dynamic programming in official and agent studies in Brazil – main features and improvements on the NEWAVE model*, in: 2018 Power Systems Computation Conference, IEEE (2018) 1–7.

[43] F. Maggioni, E. Allevi, M. Bertocchi: *Bounds in multistage linear stochastic programming*, J. Optimization Theory Appl. 163/1 (2014) 200–229.

[44] F. Maggioni, G. Pflug: *Guaranteed bounds for general non-discrete multistage risk-averse stochastic optimization programs*, SIAM J. Optimization 29/1 (2019) 454–483.

[45] J. Matousek, B. Gärtner: *Understanding and Using Linear Programming*, Springer, Berlin (2007).

[46] M. V. F. Pereira, L. M. V. G. Pinto: *Multi-stage stochastic optimization applied to energy planning*, Math. Programming 52/1-3 (1991) 359–375.

[47] G. C. Pflug, A. Pichler: *A distance for multistage stochastic optimization models*, SIAM J. Optimization 22/1 (2012) 1–23.

[48] A. Philpott, V. de Matos, E. Finardi: *On solving multistage stochastic programs with coherent risk measures*, Oper. Research 61/4 (2013) 957–970.

[49] A. B. Philpott, Z. Guan: *On the convergence of stochastic dual dynamic programming and related methods*, Oper. Research Letters 36/4 (2008) 450–455.

[50] A. B. Philpott, F. Wahid, J. F. Bonnans: *Midas: A mixed integer dynamic approximation scheme*, Math. Programming 181/1 (2020) 19–50.

[51] C. Ramirez-Pico, E. Moreno: *Generalized adaptive partition-based method for two-stage stochastic linear programs with fixed recourse*, Math. Programming B 196/1-2 (2022) 755–774.

[52] R. T. Rockafellar, S. Uryasev: *Optimization of conditional value-at-risk*, J. Risk 2/3 (2000) 21–41.

[53] R. T. Rockafellar, R. J-B Wets: *Scenarios and policy aggregation in optimization under uncertainty*, Math. Oper. Research 16/1 (1991) 119–147.

[54] R. T. Rockafellar, R. J-B Wets: *Variational Analysis*, Grundlehren der Mathematischen Wissenschaften Vol. 317, Springer, Berlin (1998).

[55] J. O. Royset, R. J-B Wets: *An Optimization Primer*, Springer, Cham (2021).

[56] A. Shapiro: *On complexity of multistage stochastic programs*, Oper. Research Letters 34/1 (2006) 1–8.

[57] A. Shapiro: *Analysis of stochastic dual dynamic programming method*, Eur. J. Oper. Research 209/1 (2011) 63–72.

[58] A. Shapiro, D. Dentcheva, A. Ruszczyński: *Lectures on Stochastic Programming: Modeling and Theory*, 2nd ed., MOS/SIAM Series on Optimization Vol. 16, Society for Industrial and Applied Mathematics, Philadelphia (2014).

[59] A. Shapiro, L. Ding: *Periodical multistage stochastic programs*, SIAM J. Optimization 30/3 (2020) 2083–2102.

[60] A. Shapiro, A. Nemirovski: *On complexity of stochastic programming problems*, in: *Continuous Optimization. Current Trends and Modern Applications*, V. Jeyakumar et al. (eds.), Springer, New York (2005) 111–146.

[61] A. Shapiro, W. Tekaya, J. P. da Costa, M. P. Soares: *Risk neutral and risk averse stochastic dual dynamic programming method*, European J. Oper. Research 224/2 (2013) 375–391.

[62] M. Siddig, Y. Song: *Adaptive partition-based SDDP algorithms for multistage stochastic linear programming with fixed recourse*, Comp. Optim. Appl. 81/1 (2022) 201–250.

[63] Y. Song, J. Luedtke: *An adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse*, SIAM J. Optimization 25/3 (2015) 1344–1367.

[64] B. Sturmfels, R. R. Thomas: *Variation of cost functions in integer programming*, Math. Programming 77/2 (1997) 357–387.

[65] W. Van Ackooij, W. de Oliveira, Y. Song: *On level regularization with normal solutions in decomposition methods for multistage stochastic programming problems*, Comp. Optim. Appl. 74/1 (2019) 1–42.

[66] R. M. Van Slyke, R. Wets: *L-shaped linear programs with applications to optimal control and stochastic programming*, SIAM J. Appl. Math. 17/4 (1969) 638–663.

[67] D. Walkup, R. J-B Wets: *Lifting projections of convex polyhedra*, Pacific J. Math. 28/2 (1969) 465–475.

[68] S. W. Wallace, W. T. Ziemba (eds.): *Applications of Stochastic Programming*, MPS-SIAM Series on Optimization 5, Society for Industrial and Applied Mathematics, Philadelphia (2005).

[69] S. Zhang, X. A. Sun: *Stochastic dual dynamic programming for multistage stochastic mixed-integer nonlinear optimization*, Math. Programming 196/1 (2022) 935–985.

[70] S. Zhang, X. A. Sun: *On distributionally robust multistage convex optimization: new algorithms and complexity analysis*, arXiv: 2010.06759 (2020).

[71] J. Zou, S. Ahmed, X. A. Sun: *Stochastic dual dynamic integer programming*, Math. Programming 175/1 (2019) 461–502.