Mathematics of Operations Research

On the Convergence of Decomposition Methods for Multistage Stochastic Convex Programs

P. Girardeau, V. Leclere, A. B. Philpott

Please scroll down for article—it is on subsequent pages

On the Convergence of Decomposition Methods for Multistage Stochastic Convex Programs

P. Girardeau

Artelys, Paris 75002, France, pierre.girardau@artelys.com

V. Leclere

CERMICS, École des Ponts ParisTech, Champs sur Marne 77455, France, vincent.leclere@cermics.enpc.fr

A. B. Philpott

Department of Engineering Science, University of Auckland, Auckland 1010, New Zealand, a.philpott@auckland.ac.nz

We prove the almost-sure convergence of a class of sampling-based nested decomposition algorithms for multistage stochastic convex programs in which the stage costs are general convex functions of the decisions and uncertainty is modelled by a scenario tree. As special cases, our results imply the almost-sure convergence of stochastic dual dynamic programming, cutting-plane and partial-sampling (CUPPS) algorithm, and dynamic outer-approximation sampling algorithms when applied to problems with general convex cost functions.

1. Introduction. Multistage stochastic programs with recourse are well known in the stochastic programming community and are becoming more common in applications. We are motivated in this paper by applications in which the stage costs are nonlinear convex functions of the decisions. Production functions are often modelled as nonlinear concave functions of allocated resources. For example, Finardi and da Silva [4] use this approach to model hydroelectricity production as a concave function of water flow. Smooth nonlinear value functions also arise when one maximizes profit with linear demand functions (see, e.g., Philpott and Guan [12]), giving a concave quadratic objective, or when coherent risk measures are defined by continuous distributions in multistage problems (Shapiro [15]).

Having general convex stage costs does not preclude the use of cutting plane algorithms to attack these problems. Kelley's cutting plane method (Kelley [7]) was originally devised for general convex objective functions, and can be shown to converge to an optimal solution (see, e.g., Ruszczyński [14, Theorem 7.7]), although on some instances this convergence might be very slow (Nesterov [9]). Our goal in this paper is to extend the convergence result of Ruszczyński [14] to the setting of multistage stochastic convex programming.

The most well-known application of cutting planes in multistage stochastic programming is the stochastic dual dynamic programming (SDDP) algorithm of Pereira and Pinto [10]. This algorithm constructs feasible dynamic programming (DP) policies using an outer approximation of a (convex) future cost function that is computed using Benders cuts. The policies defined by these cuts can be evaluated using simulation and their performance measured against a lower bound on their expected cost. This provides a convergence criterion that may be applied to terminate the algorithm when the estimated cost of the candidate policy is close enough to its lower bound. The SDDP algorithm has led to a number of related methods (Chen and Powell [1], Donohue [2], Donohue and Birge [3], Hindsberger and Philpott [6], Philpott and Guan [11]) that are based on the same essential idea but that seek to improve the method by exploiting the structure of particular applications. We call these methods DOASA (dynamic outer-approximation sampling algorithms), but they are now generically named SDDP methods.

SDDP methods are known to converge almost surely on a finite scenario tree when the stage problems are linear programs. The first formal proof of such a result was published by Chen and Powell [1], who derived this for their cutting-plane and partial-sampling (CUPPS) algorithm. This proof was extended by Linowsky and Philpott [8] to cover other SDDP algorithms. The convergence proofs in Chen and Powell [1] and Linowsky and Philpott [8] make use of an unstated assumption regarding the independence of sampled random variables and convergent subsequences of algorithm iterates. This assumption was identified by Philpott and Guan [11], who gave a simpler proof of almost sure convergence of SDDP methods based on the finite convergence of the nested decomposition algorithm (see Donohue [2]). This does not require the unstated assumption, but exploits the fact that the collection of subproblems to be solved have polyhedral value functions, and so a finite number of dual extreme points. This

begs the question of whether SDDP methods will converge almost surely for general convex stage problems, where the value functions might not be polyhedral.

In this paper we propose a different approach from the one in Chen and Powell [1] and Linowsky and Philpott [8] and show how a proof of convergence for sampling-based nested decomposition algorithms on finite scenario trees can be established for models with convex subproblems (which may not have polyhedral value functions). Our result is proved for a general class of methods, including all the variations discussed in the literature (Chen and Powell [1], Donohue [2], Donohue and Birge [3], Hindsberger and Philpott [6], Pereira and Pinto [10], Philpott and Guan [11]). The proof establishes convergence with probability 1 as long as the sampling in the forward pass is independent of previous realisations. Our proof relies heavily on the independence assumption and makes use of the Strong Law of Large Numbers.

The result we prove works in the space of state variables expressed as random variables adapted to the filtration defined by the scenario tree. Because this tree has a finite number of nodes, this space is compact, so we may extract convergent subsequences for any infinite sequence of states. Unlike the arguments in Chen and Powell [1] and Linowsky and Philpott [8], these subsequences are not explicitly constructed, so we escape the need to assume properties of them that we wish to be inherited from independent sampling. More precisely, Lemma A.3 gives us the required independence.

Although the value functions we construct admit an infinite number of subgradients, our results do require an assumption that serves to bound the norms of the subgradients used. This assumption is an extension of relatively complete recourse that ensures that some infeasible candidate solutions to any stage problem can be forced to be feasible by a suitable control. Since we are working in the realm of nonlinear programming, some constraint qualification of this form will be needed to ensure that we can extract subgradients. In practice, SDDP models use penalties on constraint violations to ensure feasibility, which implicitly bounds the subgradients of the Bellman functions. Our recourse assumptions are arguably weaker, since we do not have a result that shows that they enable an equivalent formulation with an exact penalization of infeasibility.

The paper is laid out as follows. We first consider a deterministic multistage problem, in which the proof is easily understandable. This is then extended in §3 to a stochastic problem formulated in a scenario tree. We close with some remarks about the convergence of sampling algorithms.

2. The deterministic case.

Our convergence proofs are based around showing that a sequence of outer approximations formed by cutting planes converges to the true Bellman function in the neighbourhood of the optimal state trajectories. We begin by providing a proof that Kelley's cutting plane method (Kelley [7]) converges when applied to the optimization problem:

$$W^* := \min_{u \in \mathcal{U}} W(u),$$

where $\mathcal{U}$ is a nonempty convex subset of $\mathbb{R}^m$ and W is a convex finite function on $\mathbb{R}^m$. The result we prove is not directly used in the more complex results that follow, but the main ideas on which the proofs rely are the same. We believe the reader will find it convenient to already have the scheme of the proof in mind when studying the more important results.

Kelley's method generates a sequence of iterates $(u^j)_{j \in \mathbb{N}}$ by solving, at each iteration, a piecewise linear model of the original problem. The model is then enhanced by adding a cutting plane based on the value $W(u^j)$ and subgradient g^j of W at u^j. The model at iteration k is denoted by

$$W^k(u) := \max_{1 \le j \le k} \left\{ W(u^j) + \langle g^j, u - u^j \rangle \right\},$$

and $\theta^k := \min_{u \in \mathcal{U}} W^k(u) = W^k(u^{k+1})$. We have the following result.

LEMMA 2.1. *If W is convex and $\mathcal{U}$ is compact then*

$$\lim_{k \to +\infty} W(u^k) = W^*.$$

PROOF. This proof is taken from Ruszczyński [14, Theorem 7.7] (see also Ruszczyński [13]). Let ε be an arbitrary positive number and let K_ε be the set of indices k such that $W^* + \varepsilon < W(u^k) < +\infty$. The proof consists in showing that K_ε is finite.

Suppose $k_1, k_2 \in K_\varepsilon$ and k_1 are strictly smaller than k_2. We have that $W(u^{k_1}) > W^* + \varepsilon$ and that $W^* \ge \theta^{k_1}$. Since a new cut is generated at u^{k_1}, we have

$$W(u^{k_1}) + \langle g^{k_1}, u - u^{k_1} \rangle \le W^{k_1}(u) \le W^{k_2-1}(u), \quad \forall u \in \mathcal{U},$$

where g^{k_1} is an element of $\partial W(u^{k_1})$. In particular, choosing $u = u^{k_2}$ gives

$$W(u^{k_1}) + \langle g^{k_1}, u^{k_2} - u^{k_1} \rangle \leq W^{k_1}(u^{k_2}) \leq W^{k_2-1}(u^{k_2}) = \theta^{k_2-1} \leq W^*.$$

But $\varepsilon < W(u^{k_2}) - W^*$, so

$$\varepsilon < W(u^{k_2}) - W(u^{k_1}) - \langle g^{k_1}, u^{k_2} - u^{k_1} \rangle,$$

and as $g^{k_2} \in \partial W(u^{k_2})$, the subgradient inequality for $u = u^{k_1}$ yields

$$W(u^{k_2}) - W(u^{k_1}) \leq \langle g^{k_2}, u^{k_2} - u^{k_1} \rangle.$$

Since W is finite valued, it has uniformly bounded subgradients on $\mathcal{U}$, so there exists $\kappa > 0$ such that

$$\varepsilon < 2\kappa \|u^{k_2} - u^{k_1}\|, \quad \forall k_1, k_2 \in K_\varepsilon, \ k_1 \neq k_2.$$

Because $\mathcal{U}$ is compact, K_ε has to be finite. Otherwise there would exist a convergent subsequence of $\{u^k\}_{k \in K_\varepsilon}$, and this last inequality could not hold for sufficiently large indices within K_ε. This proves the lemma. $\square$

Note that Lemma 2.1 does not imply that the sequence of iterates $(u^k)_{k \in \mathbb{N}}$ converges.[1] For instance, if the minimum of W is attained on a "flat" part (if W is not strictly convex), then the sequence of iterates may not converge. However, the lemma shows that the sequence of W values at these iterates will converge.

We now consider the multistage case. Let T be a positive integer. We first consider the following deterministic optimal control problem:

$$\min_{x,u} \ \sum_{t=0}^{T-1} C_t(x_t, u_t) + V_T(x_T) \tag{1a}$$

$$\text{s.t.} \ x_{t+1} = f_t(x_t, u_t), \quad \forall t = 0, \dots, T-1, \tag{1b}$$

$$x_0 \text{ is given}, \tag{1c}$$

$$x_t \in \mathcal{X}_t, \quad \forall t = 0, \dots, T, \tag{1d}$$

$$u_t \in \mathcal{U}_t(x_t), \quad \forall t = 0, \dots, T-1. \tag{1e}$$

In what follows, we let $\text{Aff}(\mathcal{X})$ denote the affine hull of $\mathcal{X}$, and define

$$B_t(\delta) = \{y \in \text{Aff}(\mathcal{X}_t) \mid \|y\| < \delta\}.$$

We make the following assumptions (H_1):
1. For $t = 0, \dots, T$, $\varnothing \neq \mathcal{X}_t \subseteq \mathbb{R}^n$.
2. For $t = 0, \dots, T-1$, multifunctions $\mathcal{U}_t \colon \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ are assumed to be convex[2] and nonempty compact valued.
3. The final cost function V_T is a convex lower semicontinuous proper function. The cost functions $C_t(x, u)$, $t = 0, \dots, T$ are assumed to be convex lower semicontinuous proper functions of x and u.
4. For $t = 0, \dots, T-1$, functions f_t are affine, namely $f_t(x_t) = A_t x_t + B_t u_t + b_t$.
5. The final cost function V_T is finite-valued and Lipschitz-continuous on $\mathcal{X}_T$.
6. For $t = 0, \dots, T-1$, there exists $\delta_t > 0$, defining $\mathcal{X}'_t := \mathcal{X}_t + B_t(\delta_t)$, such that
   (a) $\forall x \in \mathcal{X}'_t, \forall u \in \mathcal{U}_t(x), C_t(x, u) < \infty$
   (b) for every $x \in \mathcal{X}'_t$,
$$f_t(x, \mathcal{U}_t(x)) \cap \mathcal{X}_{t+1} \neq \varnothing.$$

Assumptions $(H_1(1) - (5))$ are made to guarantee that problem (1) is a convex optimization problem. Since this problem is in general nonlinear, it also requires a constraint qualification to ensure the existence of subgradients. This is the purpose of Assumption $(H_1(6))$. This assumption means that we can always move from $\mathcal{X}_t$ a distance of $\delta_t/2$ in any direction and stay in $\mathcal{X}'_t$, which is a form of recourse assumption that we call *extended relatively complete recourse* (ERCR). We note that this is less stringent than imposing complete recourse, which would require $\mathcal{X}'_t = \mathbb{R}^n$. Finally, we note that we never need to evaluate $C_t(x, u)$ with $x \in \mathcal{X}'_t \backslash \mathcal{X}_t$, so we may only assume that there exists a convex function, finite on $\mathcal{X}'_t$, that coincides with C_t on $\mathcal{X}_t$. Of course, not all convex cost functions satisfy such a property; e.g., $x \mapsto x \log(x)$ cannot be extended below $x = 0$ while maintaining convexity.

---

[1] Even though because $\mathcal{U}$ is compact, there exists a convergent subsequence.

[2] Recall that a multifunction $\mathcal{U}$ on convex set $\mathcal{X}$ is called *convex* if $(1-\lambda)\mathcal{U}(x) + \lambda\mathcal{U}(y) \subseteq \mathcal{U}((1-\lambda)x + \lambda y)$ for every $x, y \in \mathcal{X}$ and $\lambda \in (0, 1)$.

We are now in a position to describe an algorithm for the deterministic control problem (1). The DP equation associated with (1) is as follows. For all $t = 0, \ldots, T-1$, let

$$
V_t(x_t) = \begin{cases} \min_{u_t \in \mathcal{U}_t(x_t)} \underbrace{\{C_t(x_t, u_t) + V_{t+1}(f_t(x_t, u_t))\}}_{:= W_t(x_t, u_t)}, & x_t \in \mathcal{X}_t, \\ +\infty, & \text{otherwise.} \end{cases} \tag{2}
$$

Here the quantity $W_t(x_t, u_t)$ is the future optimal cost starting at time t from state x_t and choosing decision u_t, so that $V_t(x_t) = \min_{u \in \mathcal{U}_t(x_t)} W_t(x_t, u)$. By virtue of H_1, the functions $V_t(x_t)$ defined by (2) are convex.

The cutting plane method works as follows. At iteration 0, define functions V_t^0, $t = 0, \ldots, T-1$, to be identically equal to $-\infty$. At time T, since we know exactly the end value function, we impose $V_T^k = V_T$ for all iterations $k \in \mathbb{N}$. At each iteration k, the process is the following:

Starting with $x_0^k = x_0$, at any time stage t, solve

$$
\theta_t^k = \min_{u_t, x} \ \{C_t(x, u_t) + V_{t+1}^{k-1} \circ f_t(x, u_t)\}, \tag{3a}
$$

$$
\text{s.t.} \quad x = x_t^k \quad [\beta_t^k], \tag{3b}
$$

$$
f_t(x, u_t) \in \mathcal{X}_{t+1}, \tag{3c}
$$

$$
u_t \in \mathcal{U}_t(x). \tag{3d}
$$

Observe that although they appear as variables in the objective function, by virtue of (3b), $x = x_t^k$ are given parameters of the problem (3), which has an optimal value θ_t^k that varies with $x_t^k \in \text{Aff}(\mathcal{X}_t)$. We denote this optimal value function by $\hat{V}_t^k(x_t^k)$. We define $\beta_t^k \in \text{Aff}(\mathcal{X}_t)$ to be a vector of Lagrange multipliers for the constraint $x = x_t^k$. So

$$
\beta_t^k \in \partial(\hat{V}_t^k|_{\text{Aff}(\mathcal{X}_t)})(x_t^k). \tag{4}
$$

(When β_t^k is not uniquely defined, we make an arbitrary choice, say by selecting β_t^k with minimum norm.) The assumption that x_t^k and β_t^k both lie in $\text{Aff}(\mathcal{X}_t)$ loses no essential generality. In practice, we would expect $\text{Aff}(\mathcal{X}_t)$ to be the same dimension for every t. If this dimension happened to be d strictly less than n, then we might change the formulation (by a transformation of variables) so that $\text{Aff}(\mathcal{X}_t) = \mathbb{R}^d$.

We denote a minimizer of (3) by u_t^k. Its existence is guaranteed by ERCR. Note that constraint (3c) can be seen as an induced constraint on u_t. Thus, we can define the multifunctions $\tilde{\mathcal{U}}_t \colon \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ by, for all $x \in \mathbb{R}^n$,

$$
\tilde{\mathcal{U}}_t(x) := \{u \in \mathcal{U}_t(x) \mid f_t(x, u_t) \in \mathcal{X}_{t+1}\}. \tag{5}
$$

We can easily check that $\tilde{\mathcal{U}}_t$ is convex (by linearity of f_t and convexity of $\mathcal{U}_t$) and so convex compact valued (as the intersection of a compact set and a closed set). Moreover, ERCR guarantees that $\tilde{\mathcal{U}}_t(x) \neq \varnothing$ for any $x \in \mathcal{X}_t$. Thus, (3) can be written as

$$
\theta_t^k = \min_{\substack{u_t \in \tilde{\mathcal{U}}_t(x) \\ x \in \text{Aff}(\mathcal{X}_t)}} \ \{C_t(x, u_t) + V_{t+1}^{k-1} \circ f_t(x, u_t)\}, \tag{6a}
$$

$$
\text{s.t.} \quad x = x_t^k \quad [\beta_t^k]. \tag{6b}
$$

Now define, for any $x \in \text{Aff}(\mathcal{X}_t)$:

$$
V_t^k(x) := \max\{V_t^{k-1}(x), \theta_t^k + \langle \beta_t^k, x - x_t^k \rangle\} \tag{7}
$$

and move on to the next time stage $t+1$ by defining $x_{t+1}^k = f_t(x_t^k, u_t^k)$.

Observe that our algorithm uses V_{t+1}^{k-1} when solving the two-stage problem (3) at stage t, although most implementations of SDDP and related algorithms proceed backwards and are thus able to use the freshly updated V_{t+1}^k (although see, e.g., Chen and Powell [1] for a similar approach to the one proposed here). In the stochastic case we present a general framework that encompasses backward passes.

Note that only the last future cost function V_T is known exactly at any iteration. All the other ones are lower approximations consisting of the maximum of k affine functions at iteration k. We naturally have the same lower approximation for function W_t. Thus, we define for any $x \in \text{Aff}(\mathcal{X}_t)$, $u \in \mathbb{R}^m$

$$
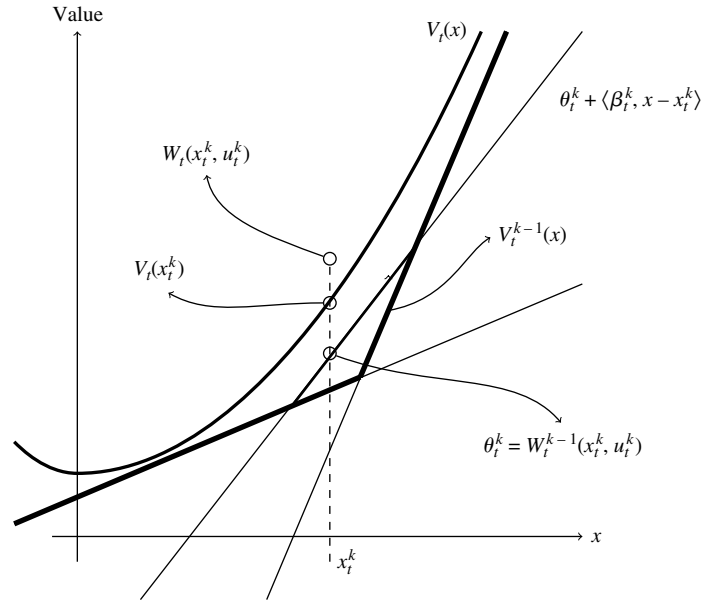W_t^k(x, u) := C_t(x, u) + V_{t+1}^k \circ f_t(x, u), \tag{8}
$$

FIGURE 1. Relation between the values at a given iteration.

and recall

$$W_t(x, u) := C_t(x, u) + V_{t+1} \circ f_t(x, u). \tag{9}$$

Using this notation we have

$$\theta_t^k = \min_{u \in \tilde{\mathcal{U}}_t(x_t^k)} W_t^{k-1}(x_t^k, u) = W_t^{k-1}(x_t^k, u_t^k). \tag{10}$$

Since by (7)

$$V_t^k(x_t^k) = \max_{k' \le k} \{\theta_t^{k'} + \langle \beta_t^{k'}, x_t^k - x_t^{k'} \rangle\},$$

it follows that

$$V_t^k(x_t^k) \ge W_t^{k-1}(x_t^k, u_t^k). \tag{11}$$

Figure 1 gives a view of the relations between all these values at a given iteration.

2.1. Proof of convergence in the deterministic case. Recall the approximate value function $\hat{V}_t^k$ defined by the optimal value of (3) (equivalently (6)). We begin by showing some regularity and monotonicity results for this and the value functions V_t^k and V_t.

Under assumptions (H_1), we define for $t = 0, \dots, T-1$, and for all $x \in \text{Aff}(\mathcal{X}_t)$, the extended value function

$$\tilde{V}_t(x) = \inf_{u \in \mathcal{U}_t(x)} \left\{ C_t(x, u) + V_{t+1} \circ f_t(x, u) \right\}. \tag{12}$$

Note that the infimum could be taken on $\tilde{\mathcal{U}}_t(x) \subseteq \mathcal{U}_t(x)$, as $V_{t+1} = \infty$ when $f_t(x, u) \notin \mathcal{X}_{t+1}$. It is convenient to extend the definition to $t = T$ by defining $\tilde{V}_T = V_T$. We also observe that $\tilde{V}_t \le V_t$ as these are identical on the domain of V_t.

LEMMA 2.2. *For $t = 0, \dots, T-1$,*
  (i) *the value function V_t is convex and Lipschitz continuous on $\mathcal{X}_t$.*
 (ii) $V_t^k \le \tilde{V}_t \le V_t$, *and at all points in $\mathcal{X}_t$, $\partial(\hat{V}_t^k|_{\text{Aff}(\mathcal{X}_t)}) \ne \varnothing$.*
(iii) *the sequences $(\beta_t^k)_{k \in \mathbb{N}}$ are bounded.*

PROOF. (i) We first show the convexity and Lipschitz continuity of V_t on $\mathcal{X}_t$. We proceed by induction backward in time. By assumption, V_T is convex and Lipschitz continuous on $\mathcal{X}_T$. Assume the result is true for V_{t+1}. First, since it is the infimum over u of a sum of convex functions of x and u, the function $\tilde{V}_t(x)$ is convex. Now by ERCR, for any $x \in \mathcal{X}_t'$, $\tilde{\mathcal{U}}_t(x) \ne \varnothing$. This implies that, for $x \in \mathcal{X}_t'$, for $u \in \tilde{\mathcal{U}}_t(x)$,

$$\tilde{V}_t(x) \le C_t(x, u) + V_{t+1} \circ f_t(x, u) < +\infty.$$

By $(H_1(3))$ and the induction hypothesis, for any $x \in \mathcal{X}'_t$,

$$u \mapsto C_t(x, u) + V_{t+1} \circ f_t(x, u)$$

is lower semicontinuous, and so the compactness of $\mathcal{U}_t(x)$ ensures that the infimum in the definition of $\tilde{V}_t(x)$ is attained, and therefore $\tilde{V}_t(x) > -\infty$. $\tilde{V}_t$ is Lipschitz continuous on $\mathcal{X}_t$, as $\mathcal{X}_t$ is a compact subset of the relative interior of its domain. Finally remarking that $V_t(x) = \tilde{V}_t(x)$ if $x \in \mathcal{X}_t$ gives the conclusion.

(ii) As observed, the inequality $\tilde{V}_t \le V_t$ is immediate, as the two functions are identical on the domain of V_t.

To show $V_t^k \le \tilde{V}_t$ let us proceed by induction forward in k. Assume that for all $t = 0, \ldots, T - 1$, β_t^{k-1} is defined and $V_t^{k-1} \le \tilde{V}_t$. Note that

$$-\infty = V_t^0 \le \tilde{V}_t,$$

so this is true for $k = 1$ (β_t^0 is never used). Recall that for all $t = 0, \ldots, T - 1$ and all $x \in \text{Aff}(\mathcal{X}_t)$, that

$$\hat{V}_t^k(x) = \min_{u \in \tilde{\mathcal{U}}_t(x)} \left\{ C_t(x, u) + V_{t+1}^{k-1} \circ f_t(x, u) \right\}.$$

The properties of $\tilde{\mathcal{U}}_t$ guarantee that $\hat{V}_t^k$ is convex and finite on $\mathcal{X}'_t$ which strictly contains $\mathcal{X}_t$. Thus, $\hat{V}_t^k$ restricted to $\text{Aff}(\mathcal{X}_t)$ is subdifferentiable at any point of $\mathcal{X}_t$, giving $\partial(\hat{V}_t^k|_{\text{Aff}(\mathcal{X}_t)}) \ne \varnothing$ at every point in $\mathcal{X}_t$. By the induction hypothesis and inequality $\tilde{V}_{t+1} \le V_{t+1}$ we have that

$$V_{t+1}^{k-1} \circ f_t \le V_{t+1} \circ f_t.$$

Thus, the definitions of $\hat{V}_t^k$ and $\tilde{V}_t$ yield

$$\hat{V}_t^k \le \tilde{V}_t. \tag{13}$$

we have by (4) that

$$\theta_t^k + \langle \beta_t^k, x - x_t^k \rangle \le \hat{V}_t^k(x) \le \tilde{V}_t(x) \tag{14}$$

by (13). The definition of V_t^k in (7) gives

$$V_t^k(x) = \max\left\{ V_t^{k-1}(x), \theta_t^k + \langle \beta_t^k, x - x_t^k \rangle \right\},$$

which shows $V_t^k(x) \le \tilde{V}_t(x)$ by (14) and the induction hypothesis. Thus, (ii) follows for all k by induction.

(iii) Finally, we show the boundedness of $(\beta_t^k)_{k \in \mathbb{N}}$. By the definition of β_t^k we have for all $y \in \text{Aff}(\mathcal{X}_t)$

$$V^k(y) \ge V^k(x_t^k) + \langle \beta_t^k, y - x_t^k \rangle. \tag{15}$$

Recall that $\mathcal{X}'_t = \mathcal{X}_t + B_t(\delta_t)$, so substituting $y = x_t^k + (\delta_t \beta_t^k)/(2\|\beta_t^k\|)$ in (15) whenever $\beta_t^k \ne 0$ yields

$$\|\beta_t^k\| \le \frac{2}{\delta_t} \left[V_t^k\left(x_t^k + \frac{\delta_t}{2} \frac{\beta_t^k}{\|\beta_t^k\|} \right) - V_t^k(x_t^k) \right].$$

We define the compact subset $\mathcal{X}''_t$ of dom $\tilde{V}_t$ as $\mathcal{X}''_t := \mathcal{X}_t + B_t(\delta_t/2)$. As $x_t^k \in \mathcal{X}_t$ we have that $x_t^k + (\delta_t/2) \cdot (\beta_t^k/\|\beta_t^k\|) \in \mathcal{X}''_t$. Consequently, by (ii),

$$V_t^k\left(x_t^k + \frac{\delta_t}{2} \frac{\beta_t^k}{\|\beta_t^k\|} \right) \le \max_{x \in \mathcal{X}''_t} \tilde{V}_t(x) < +\infty.$$

Moreover, by construction the sequence of functions $(V_t^k)_{k \in \mathbb{N}}$ is increasing; thus,

$$V_t^k(x_t^k) \ge V_t^1(x_t^k) \ge \min_{x \in \mathcal{X}_t} V_t^1(x) > -\infty.$$

Thus we have that, for all $k \in \mathbb{N}^*$ and $t = 0, \ldots, T - 1$,

$$\|\beta_t^k\| \le \frac{2}{\delta_t} \left[\max_{x \in \mathcal{X}''_t} \tilde{V}_t(x) - \min_{x \in \mathcal{X}_t} V_t^1(x) \right]. \tag{16}$$

This completes the proof. □

COROLLARY 2.1. *Under assumptions* (H_1), *the functions* V_t^k, $t = 0, 1, \ldots, T - 1$, *are* α-*Lipschtiz for some constant* α *for all* $k \in \mathbb{N}^*$.

PROOF. By (7) and (16) the subgradients of V_t^k are bounded by

$$\alpha = \max_{t=0,1,\ldots,T-1} \frac{2}{\delta_t} \Big[\max_{x \in \mathscr{X}_t''} \tilde{V}_t - \min_{x \in \mathscr{X}_t} V_t^1(x) \Big]. \quad \square$$

We now prove that both the upper and lower estimates of V_t converge to the exact value function under assumptions (H_1).

THEOREM 2.1. *Consider the sequence of decisions $(u^k)_{k \in \mathbb{N}}$ generated by (3) and (7), where each u^k is itself a sequence of decisions in time $u^k = u_0^k, \ldots, u_{T-1}^k$, and consider the corresponding sequence of state values $(x^k)_{k \in \mathbb{N}}$. Under assumptions (H_1), for any $t = 0, \ldots, T-1$ we have that*

$$\lim_{k \to +\infty} \big[W_t(x_t^k, u_t^k) - V_t(x_t^k) \big] = 0 \qquad and \qquad \lim_{k \to +\infty} \big[V_t(x_t^k) - V_t^k(x_t^k) \big] = 0.$$

PROOF. The demonstration proceeds by induction backward in time. At time $t+1$, the induction hypothesis is the second statement of the theorem. That is,

$$\lim_{k \to +\infty} \big[V_{t+1}(x_{t+1}^k) - V_{t+1}^k(x_{t+1}^k) \big] = 0.$$

In other words, the cuts for the future cost function tend to be exact at x_{t+1}^k, as k tends to ∞. The induction hypothesis is clearly true at the last time stage T for which we defined the approximate value function V_T^k as equal to the (known) end value function V_T.

We have to show the induction hypothesis, namely

$$\lim_{k \to +\infty} \big[V_t(x_t^k) - V_t^k(x_t^k) \big] = 0$$

for time t. Recall that (11) gives

$$V_t^k(x_t^k) \geq W_t^{k-1}(x_t^k, u_t^k) = C_t(x_t^k, u_t^k) + V_{t+1}^{k-1}(x_{t+1}^k).$$

Using the definition (9) of W_t, we can replace $C_t(x_t^k, u_t^k)$ to get

$$V_t^k(x_t^k) \geq W_t(x_t^k, u_t^k) + (V_{t+1}^{k-1}(x_{t+1}^k) - V_{t+1}(x_{t+1}^k)).$$

Subtracting $V_t(x_t^k)$ we obtain

$$V_t^k(x_t^k) - V_t(x_t^k) \geq W_t(x_t^k, u_t^k) - V_t(x_t^k) + (V_{t+1}^{k-1}(x_{t+1}^k) - V_{t+1}(x_{t+1}^k)).$$

Now as V_t^k is a lower approximation of V_t we have

$$V_t^k(x_t^k) - V_t(x_t^k) \leq 0,$$

and by DP equation (2)

$$W_t(x_t^k, u_t^k) - V_t(x_t^k) \geq 0.$$

Moreover, the induction hypothesis at time $t+1$ gives

$$V_{t+1}^k(x_{t+1}^k) - V_{t+1}(x_{t+1}^k) \xrightarrow{k \to \infty} 0,$$

which by virtue of Lemma A.1 (with V_{t+1} replacing f) implies[3]

$$\lim_{k \to +\infty} \big[V_{t+1}(x_{t+1}^k) - V_{t+1}^{k-1}(x_{t+1}^k) \big] = 0$$

so

$$V_t^k(x_t^k) - V_t(x_t^k) \xrightarrow{k \to \infty} 0,$$

and

$$W_t(x_t^k, u_t^k) - V_t(x_t^k) \xrightarrow{k \to \infty} 0,$$

which gives the result. $\square$

Theorem 2.1 indicates that the lower approximation at each iteration tends to be exact on the sequence of state trajectories generated throughout the algorithm. This does not mean that the future cost function will be approximated well everywhere in the state space. It only means that the approximation gets better and better in the neighborhood of an optimal state trajectory.

---

[3] Corollary 2.1 ensures the α-Lipschitz assumption on V_{t+1}^k, and the other assumptions are obviously verified.

3. The stochastic case with a finite distribution.

3.1. Stochastic multistage problem formulation. We now consider that the cost function and dynamics at each time t are influenced by a random outcome that has a discrete and finite distribution. We write the problem on the complete tree induced by this distribution. The set of all nodes is denoted by $\mathcal{N}$, and $\{0\}$ is the root node. We denote nodes by m and n. (We trust that the context will dispel any confusion from the use of m and n as dimensions of variables u and x.) A node n here represents a time interval and a state of the world (which has probability Φ_n) that pertains over this time interval. We say that a node n is an ascendent of m if it is on the path from the root node to node m (including m). We denote $a(m)$ as the set of all ascendents of m, and the depth of node n is one less than the number of its ascendents. For simplicity we identify this with a time index t, although the results hold true for scenario trees for which this is not the case. For every node $m \in \mathcal{N} \setminus \{0\}$, $p(m)$ represents its parent and $r(m)$ its set of children nodes. Finally, $\mathcal{L}$ is the set of leaf nodes of the tree (i.e., those that have degree 1).

This gives the following stochastic program:

$$\min_{x,u} \left\{ \sum_{n \in \mathcal{N} \setminus \{\mathcal{L}\}} \sum_{m \in r(n)} \Phi_m C_m(x_n, u_m) + \sum_{m \in \mathcal{L}} \Phi_m V_m(x_m) \right\} \tag{17a}$$

$$\text{s.t.} \quad x_m = f_m(x_{p(m)}, u_m), \quad \forall m \in \mathcal{N} \setminus \{0\}, \tag{17b}$$

$$x_0 \text{ is given}, \tag{17c}$$

$$x_m \in \mathcal{X}_m, \quad \forall m \in \mathcal{N}, \tag{17d}$$

$$u_m \in \mathcal{U}_m(x_{p(m)}), \quad \forall m \in \mathcal{N} \setminus \{0\}. \tag{17e}$$

The reader should note that randomness (that appears in the cost and in the dynamics) is realized before the decision is taken in this model. Hence the control affecting the stock[4] x_n is actually indexed by m, a child node of n. Put differently, the control adapts to randomness: there are as many controls as there are children nodes of n. Observe that we also now admit the possibility that $\mathcal{X}_t$ and $\mathcal{U}_t(x)$ might vary with scenario-tree node, so we denote them by $\mathcal{X}_m$ and $\mathcal{U}_m(x_{p(m)})$.

We make the following assumptions (H_2):

1. For all $n \in \mathcal{N}$, $\mathcal{X}_n$ is nonempty convex compact.
2. For all $m \in \mathcal{N} \setminus \{0\}$, the multifunction $\mathcal{U}_m$ is nonempty convex and convex compact valued.
3. All functions C_n, $n \in \mathcal{N} \setminus \mathcal{L}$, V_m, $m \in \mathcal{L}$, are convex lower semicontinuous proper functions.
4. For all $m \in \mathcal{N} \setminus \{0\}$, the functions f_m are affine.
5. For all $m \in \mathcal{L}$, V_m is Lipschitz-continuous on $\mathcal{X}_m$.
6. There exists $\delta > 0$ such that for all nodes $n \in \mathcal{N} \setminus \mathcal{L}$
   (a) $\forall x \in \mathcal{X}_n + B(\delta)$, $\forall m \in r(n)$, $f_m(x, \mathcal{U}_m(x)) \cap \mathcal{X}_m \neq \varnothing$.
   (b) $\forall x \in \mathcal{X}_n + B(\delta)$, $\forall u \in \mathcal{U}_m(x)$, $C_n(x, u) < \infty$.

The convex functions V_m define the future cost of having x_m remaining in stock at the end of the stage represented by leaf node $m \in \mathcal{L}$. Given an optimal control, we can define (applying the DP principle to Problem (17)) a future cost function V_n recursively for the other nodes $n \in \mathcal{N} \setminus \mathcal{L}$ by

$$V_n(x_n) = \sum_{m \in r(n)} \frac{\Phi_m}{\Phi_n} \min_{u_m \in \mathcal{U}_m(x_n)} \underbrace{\{C_m(x_n, u_m) + V_m(f_m(x_n, u_m))\}}_{W_m(x_n, u_m)}. \tag{18}$$

In general, the future cost function at each node can be different from those at other nodes at the same stage. In the special case where the stochastic process defined by the scenario tree is stagewise independent, the future cost function is identical at every node at stage t. Some form of stagewise independence is typically assumed in applications, as that enables cuts to be shared across nodes at the same stage; however, we do not require this for our proof.

The algorithm that we consider is an extension of the deterministic algorithm of the previous section applied, at each iteration, to a set of nodes chosen randomly in the tree at which we update estimates of the future cost function. We assume that all other nodes have null updates, in the sense that they just inherit the future cost function from the previous iteration.

---

[4] We do not make any stagewise independence assumptions on the random variables that affect the system. Hence, there is no reason why variable x_n should be called a state variable, and we prefer calling it a stock.

We now describe the algorithm formally. We start the process with $\hat{\theta}_n^0 = -\infty$, $\hat{\beta}_n^0 = 0$, for each $n \in \mathcal{N}$, and impose $V_n^k = V_n$ for all nodes $n \in \mathcal{L}$ and all $k \in \mathbb{N}$. We then carry out a sequence of simulations and updates of the future cost functions as follows.

Simulation. Starting at the root node, generate stocks and decisions for all possible successors (in other words, visit the whole tree forward) by solving (18) with V^{k-1} instead of V. Denote the obtained stock variables by $(x_n^k)_{n \in \mathcal{N}}$ and the control variables by $(u_n^k)_{n \in \mathcal{N} \setminus \{0\}}$. Also, for each node $n \in \mathcal{N}$, impose $\theta_n^k = V_n^{k-1}(x_n^k)$ and $\beta_n^k \in \partial V_n^{k-1}(x_n^k)$.

Update. Select nonleaf nodes $n_1, n_2, \ldots, n_I$ in the tree. For each i, $x_{n_i}^k$ is a random variable that is equal to one of the x_n^k. For each selected node n_i, and for every child node m of node n_i, solve

$$\hat{\theta}_m^k = \min_{\substack{u_m \in \mathbb{R}^m \\ x \in \mathrm{Aff}(\mathcal{X}_{n_i})}} \left\{ C_m(x, u_m) + V_m^{k-1} \circ f_m(x, u_m) \right\} \tag{19a}$$

$$\text{s.t.} \quad x = x_{n_i}^k \quad [\hat{\beta}_m^k], \tag{19b}$$

$$u_m \in \mathcal{U}_m(x), \tag{19c}$$

$$f_m(x, u_m) \in \mathcal{X}_m. \tag{19d}$$

As before, we define $\hat{\beta}_m^k \in \mathrm{Aff}(\mathcal{X}_{n_i}^k)$ to be a vector of Lagrange multipliers for the constraint $x = x_{n_i}^k$. We also define the multifunctions

$$\tilde{\mathcal{U}}_m : x \mapsto \left\{ u \in \mathcal{U}_m(x) \mid f_m(x, u_m) \in \mathcal{X}_m \right\}.$$

For each selected node n_i, replace the values $\theta_{n_i}^k$ and $\beta_{n_i}^k$ obtained during the simulation with

$$\theta_{n_i}^k = \sum_{m \in r(n_i)} \frac{\Phi_m}{\Phi_{n_i}} \hat{\theta}_m^k$$

and

$$\beta_{n_i}^k = \sum_{m \in r(n_i)} \frac{\Phi_m}{\Phi_{n_i}} \hat{\beta}_m^k.$$

Finally, we update all future cost functions. For every node n, and any $x \in \mathcal{X}_t$,

$$V_n^k(x) := \max \left\{ V_n^{k-1}(x), \theta_n^k + \langle \beta_n^k, x - x_n^k \rangle \right\} = \max_{k' \leq k} \left\{ \theta_n^k + \langle \beta_n^k, x - x_n^k \rangle \right\}. \tag{20}$$

We will make use of the following definitions, where $m \in r(n)$

$$W_m(x_n, u_m) := C_m(x_n, u_m) + V_m(f_m(x_n, u_m)) \tag{21}$$

$$W_m^k(x_n, u_m) := C_m(x_n, u_m) + V_m^k(f_m(x_n, u_m)). \tag{22}$$

In the case where node $n \in \mathcal{N}$ is selected at iteration k—in other words, $n = n_i$—these definitions then give

$$\hat{\theta}_m^k = \min_{u \in \tilde{\mathcal{U}}_m(x_n^k)} W_m^{k-1}(x_n^k, u) = W_m^{k-1}(x_n^k, u_m^k).$$

This leads to

$$V_n^k(x_n^k) \geq \sum_{m \in r(n)} \frac{\Phi_m}{\Phi_n} W_m^{k-1}(x_n^k, u_m^k). \tag{23}$$

Note that we actually only update future cost functions on the selected nodes. Since the cuts we add at all other nodes are binding on the current model (by construction in the simulation), there is no point in storing them. Therefore, in practice, one does not need to sample the whole scenario tree but just enough to attain all selected nodes. In our proof, we need to look at what happens even on the nodes that are not selected.

The way we select nodes at which to compute cuts varies with the particular algorithm implementation. For example, DOASA uses a single forward pass to select nodes and then computes cuts in a backward pass. We represent these selections of nodes using a *selection random variable* $y^k = (y_n^k)_{n \in \mathcal{N}}$ that is equal to 1 if node n is selected at iteration k and 0 otherwise. This gives a *selection stochastic process* $(y^k)_{k \in \mathbb{N}}$, taking values in $\{0, 1\}^{|\mathcal{N} \setminus \mathcal{L}|}$, that describes a set of nodes in the tree at which we will compute new cuts in iteration k. We let $(\mathcal{F}_k)_{k \in \mathbb{N}}$ denote the filtration generated by $(y^k)_{k \in \mathbb{N}}$.

To encompass algorithms such as DOASA and SDDP the selection stochastic process can be viewed as consisting of infinitely many finite subsequences, each consisting of $\tau > 0$ selections (consisting, e.g., of a sequence of selections of nodes in a backward pass). This cannot be done arbitrarily, and the way that $(y^k)_{k \in \mathbb{N}}$ is constructed must satisfy some independence conditions from one iteration to the next.

DEFINITION 1.   Let τ be a positive integer. The process $(y^k)_{k \in \mathbb{N}}$ is called a τ-*admissible selection process* if
(i) $\forall m \in \mathcal{N} \backslash \mathcal{L}$, $\forall k \in \mathbb{N}$, $\forall \kappa \in \{0, \ldots, \tau - 1\}$,

$$y_m^{k\tau+\kappa} = 1 \implies \forall n \in a(m), \quad y_n^{k\tau} = y_n^{k\tau+1} = \cdots = y_n^{k\tau+\kappa-1} = 0;$$

and the process defined by

$$\tilde{y}_n^k := \max\left\{ y_n^{k\tau}, y_n^{k\tau+1}, y_n^{k\tau+2}, \ldots, y_n^{k\tau+\tau-1} \right\} \tag{24}$$

satisfies
(ii) for all $m \in \mathcal{N} \backslash \mathcal{L}$, $(\tilde{y}_m^k)_{k \in \mathbb{N}}$ is i.i.d. and for all $k \in \mathbb{N}$, and all $m \in \mathcal{N} \backslash \mathcal{L}$, $\tilde{y}_m^k$ is independent of $\mathcal{F}_{k\tau-1}$.
(iii) $\forall n \in \mathcal{N} \backslash \mathcal{L}$, $\mathbb{P}(\tilde{y}_n^k = 1) > 0$.

Property (i) guarantees that when $\tau > 1$, the updating of cutting planes is done backwards between steps $k\tau$ and $(k+1)\tau$. This means that if the linear approximation of the value function V_n is updated at step $k\tau + \kappa$, then neither it nor any approximation at any ascendant node has been updated since step $k\tau - 1$. This implies, as shown in Lemma A.2, that $x^{k\tau+\kappa}$ has not changed since the step $k\tau$; i.e., if $y_n^{k\tau+\kappa} = 1$, then $x^{k\tau+\kappa} = x^{k\tau}$. We explain in §3.3 how the selection processes of CUPPS (with $\tau = 1$) and SDDP (with $\tau = T - 1$) are τ-admissible.

Property (ii) provides the independence of the selections that we will use to prove convergence, and property (iii) guarantees that all nodes are selected with positive probability. Without any independence assumption it would be easy to create a case in which the future cost function at a given node is updated only when the current stock variable on this node is in a given region, for instance. In such a case, the future cost function could not gather any information about the other parts of the space that the stock variable might visit. In other words, this independence assumption ensures that the values that are optimal can be attained an infinite number of times. We remark that there is no independence assumption over the nodes n for $(y_n^k)_{n \in \mathcal{N} \backslash \mathcal{L}}$ at k fixed. Thus the selection process could be forced to select whole branches of the tree, for example, as it would for the CUPPS algorithm. More generally, we have independence when for fixed τ, $(y^{k\tau})_{k \in \mathbb{N}}$ is i.i.d and the next $\tau - 1$ selection values are determined deterministically from $y^{k\tau}$, more precisely, if for all $\kappa \in \{0, \ldots, \tau - 1\}$, there is a deterministic function ϕ_κ such that $y^{k\tau+\kappa} = \phi_\kappa(y^{k\tau})$. Yet we also have independence when the selection subsequence $(y^{k\tau}, y^{k\tau+1}, \ldots, y^{k\tau+\tau-1})_{k \in \mathbb{N}}$ is i.i.d.

3.2. Proof of convergence in the stochastic case.   For every $n \in \mathcal{N} \backslash \mathcal{L}$ we can define under assumptions (H_2) the extended value function

$$\tilde{V}_n(x) = \sum_{m \in r(n)} \frac{\Phi_m}{\Phi_n} \inf_{u \in \tilde{\mathcal{U}}_m(x)} \left\{ C_m(x, u) + V_m \circ f_m(x, u) \right\},$$

and we note that $\tilde{V}_n$ is finite on $\mathcal{X}'_n$. We now state a lemma analogous to Lemma 2.2.

LEMMA 3.1.   *For every $n \in \mathcal{N}$,*
(i) *The value function V_n is convex and Lipschitz-continuous on $\mathcal{X}_t$.*
(ii) *$V_n^k \leq \tilde{V}_n \leq V_n$, and β_n^k is defined.*
(iii) *The sequences $(\beta_n^k)_{k \in \mathbb{N}}$ are bounded, thus there is α_n such that V_n^k is α_n-Lipschitz.*

PROOF.   We give only a sketch of the proof, as it follows exactly the proof of its deterministic counterpart Lemma 2.2.
(i) By induction, backward on the tree, $\tilde{V}_n$ is convex and finite valued on $\mathcal{X}'_n$ as the positive sum of convex finite valued functions, and thus Lipschitz continuous on $\mathcal{X}_n$ leading to the result as $\tilde{V}_n = V_n$ on $\mathcal{X}_n$.
(ii) Assume that for all $n \in \mathcal{N} \backslash \mathcal{L}$ we have $V_n^{k-1} \leq \tilde{V}_n$. We define, for a node $n \in \mathcal{N} \backslash \mathcal{L}$ $x \in \mathbb{R}^n$,

$$\hat{V}_n^k(x) = \sum_{m \in r(n)} \frac{\Phi_m}{\Phi_n} \min_{u \in \tilde{\mathcal{U}}_n(x)} \left\{ C_m(x, u) + V_m^{k-1} \circ f_m(x, u) \right\}.$$

By hypothesis on $\tilde{\mathcal{U}}_m$, $\hat{V}_n^k$ is convex and finite on $\mathcal{X}'_n$, thus, its restriction on $\text{Aff}(\mathcal{X}_t)$ is subdifferentiable on $\mathcal{X}_t$. By definition, $\hat{\beta}_m^k \in \partial \hat{V}_n^k(x_m^k)$, thus, $\hat{\beta}_m^k$ is defined. By the induction hypothesis and inequality $\tilde{V}_m \leq V_m$ we have that

$$\forall m \in r(n), \quad \hat{V}_m^{k-1} \circ f_m \leq V_m \circ f_m.$$

Thus, definitions of $\hat{V}_n^k$ and $\tilde{V}_n$ yield $\hat{V}_n^k \leq \tilde{V}_n$. By definition of β_n^k and construction of V_n^k we have that $V_n^k \leq \tilde{V}_n$. Induction leads to inequality (ii).

(iii) Finally, we show the boundedness of $(\beta_n^k)_{k\in\mathbb{N}}$. As β_n^k is an element of $\partial V^k(x_n^k)$, we have

$$V^k(y) \geq V^k(x_n^k) + \langle \beta_n^k, y - x_n^k \rangle. \tag{25}$$

so substituting, if $\beta_n^k \neq 0$, $y = x_n^k + (\delta\beta_n^k)/(2\|\beta_n^k\|)$ in (25) yields

$$\|\beta_n^k\| \leq \frac{2}{\delta}\left[V_n^k\left(x_n^k + \frac{\delta}{2}\frac{\beta_n^k}{\|\beta_n^k\|} \right) - V_n^k(x_n^k) \right].$$

Thus, we have that, for all $k \in \mathbb{N}$ and $n \in \mathcal{N}$,

$$\|\beta_n^k\| \leq \frac{2}{\delta}\left[\max_{x\in\mathscr{X}_n+B(\delta/2)} \tilde{V}_n(x) - \min_{x\in\mathscr{X}_n} V_n^1(x) \right], \tag{26}$$

which completes the proof. $\square$

THEOREM 3.1. *Consider the sequence of decisions $(u^k)_{k\in\mathbb{N}}$ generated by the above procedure under assumptions (H_2), where each u^k is a set of decisions on the complete tree, and consider the corresponding sequence of state values $(x^k)_{k\in\mathbb{N}}$. Assume that the selection process is τ-admissible for some integer $\tau > 0$. Then we have for every node n that $\mathbb{P}$-almost surely*

$$\lim_{k\to+\infty}\left[\sum_{m\in r(n)} \frac{\Phi_m}{\Phi_n} W_m(x_n^{k\tau}, u_m^k) - V_n(x_n^{k\tau}) \right] = 0$$

and

$$\lim_{k\to+\infty}\left[V_n(x_n^{k\tau}) - V_n^{k\tau}(x_n^{k\tau}) \right] = 0.$$

PROOF. Because the selection process for nodes in the update step is stochastic, decision variables as well as approximate future cost functions are stochastic throughout the course of the algorithm. Thus, during the whole proof, all equalities or inequalities are taken $\mathbb{P}$-almost surely.

The demonstration follows the same approach as the proof of Theorem 2.1. Let T be the maximum depth of the tree. We procede by backward induction on nodes of fixed depth. The induction hypothesis is

$$\lim_{k\to+\infty}\left[V_m(x_m^{k\tau}) - V_m^{k\tau}(x_m^{k\tau}) \right] = 0$$

for each node m of depth $t+1$. Since for every leaf of the tree those two quantities are equal, by definition, the induction hypothesis is true for every node $n \in \mathscr{L}$.

We start by proving the result for iterations $k\tau$ such that n is selected in the next $\tau - 1$ steps, i.e., such that $\tilde{y}_n^k = 1$. Define $\kappa_k \in \{0, \ldots, \tau - 1\}$ such that $y_n^{k\tau+\kappa_k} = 1$. Recall that by Lemma A.2 we have $x_n^{k\tau+\kappa_k} = x_n^{k\tau}$.

We have by (23)

$$
\begin{aligned}
V_n^{k\tau+\kappa_k}(x_n^{k\tau}) &= V_n^{k\tau+\kappa_k}(x_n^{k\tau+\kappa_k}) \\
&\geq \sum_{m\in r(n)} \frac{\Phi_m}{\Phi_n} \min_{u_m\in\tilde{\mathcal{U}}_m(x_n^{k\tau})} \left\{ W_m^{k\tau+\kappa_k-1}(x_n^{k\tau}, u_m) \right\} \\
&\geq \sum_{m\in r(n)} \frac{\Phi_m}{\Phi_n} \min_{u_m\in\tilde{\mathcal{U}}_m(x_n^{k\tau})} \left\{ W_m^{k\tau-1}(x_n^{k\tau}, u_m) \right\} \\
&= \sum_{m\in r(n)} \frac{\Phi_m}{\Phi_n} W_m^{k\tau-1}(x_n^{k\tau}, u_m^{k\tau}),
\end{aligned}
$$

which implies

$$
\begin{aligned}
V_n^{k\tau+\kappa_k}(x_n^{k\tau}) &\geq \sum_{m\in r(n)} \frac{\Phi_m}{\Phi_n}\left[C_m(x_n^{k\tau}, u_m^{k\tau}) + V_m^{k\tau-1}(x_m^{k\tau}) \right], \\
&= \sum_{m\in r(n)} \frac{\Phi_m}{\Phi_n}\left[W_m(x_n^{k\tau}, u_m^{k\tau}) + \left(V_m^{k\tau-1}(x_m^{k\tau}) - V_m(x_m^{k\tau}) \right) \right].
\end{aligned}
$$

Thus,

$$V_n^{k\tau+\kappa_k}(x_n^{k\tau}) - V_n(x_n^{k\tau}) \geq \sum_{m \in r(n)} \left(\frac{\Phi_m}{\Phi_n} W_m(x_n^{k\tau}, u_m^{k\tau}) \right) - V_n(x_n^{k\tau})$$

$$+ \sum_{m \in r(n)} \frac{\Phi_m}{\Phi_n} (V_m^{k\tau-1}(x_m^{k\tau}) - V_m(x_m^{k\tau})).$$

Note that, as $V_n^{k\tau+\kappa_k}$ is a lower approximation of V_n, we know that

$$V_n^{k\tau+\kappa_k}(x_n^{k\tau}) - V_n(x_n^{k\tau}) \leq 0,$$

and, by DP equation (18), that

$$\sum_{m \in r(n)} \left(\frac{\Phi_m}{\Phi_n} W_m(x_n^{k\tau}, u_m^{k\tau}) \right) - V_n(x_n^{k\tau}) \geq 0.$$

The induction hypothesis

$$\lim_{k \to +\infty} \left[V_m(x_m^{k\tau}) - V_m^{k\tau}(x_m^{k\tau}) \right] = 0$$

and Lemma A.1 (with V_m replacing f)[5] imply

$$\lim_{k \to +\infty} \left[V_m(x_m^{k\tau}) - V_m^{k\tau-1}(x_m^{k\tau}) \right] = 0.$$

Thus,

$$V_n(x_n^{k\tau}) - V_n^{k\tau+\kappa_k}(x_n^{k\tau}) \xrightarrow[\tilde{y}_n^k=1]{k\to\infty} 0,$$

and

$$\sum_{m \in r(n)} \left(\frac{\Phi_m}{\Phi_n} W_m(x_n^{k\tau}, u_m^{k\tau}) \right) - V_n(x_n^{k\tau}) \xrightarrow[\tilde{y}_n^k=1]{k\to\infty} 0.$$

Thus, Lemma A.1 applied with $\kappa = \tau$ gives

$$V_n(x_n^{k\tau}) - V_n^{k\tau+\kappa_k-\tau}(x_n^{k\tau}) \xrightarrow[\tilde{y}_n^k=1]{k\to\infty} 0,$$

and by monotonicity we have $V_n^{k\tau+\kappa_k-\tau} \leq V_n^{k\tau} \leq V_n$, which finally yields

$$V_n(x_n^{k\tau}) - V_n^{k\tau}(x_n^{k\tau}) \xrightarrow[\tilde{y}_n^k=1]{k\to\infty} 0. \tag{27}$$

Now we prove that the values also converge for the iterations k such that $\tilde{y}_n^k = 0$, i.e., the iterations for which node n is not selected between step $k\tau$ and step $(k+1)\tau - 1$. By contradiction, suppose the values do not converge. Then by Lemma A.1 we have that $V_n(x_n^{k\tau}) - V_n^{k\tau-1}(x_n^{k\tau})$ does not converge to 0. It follows that there is some $\varepsilon > 0$ such that $\mathcal{K}_\varepsilon$ is infinite where

$$\mathcal{K}_\varepsilon := \left\{ k \in \mathbb{N} \mid V_n(x_n^{k\tau}) - V_n^{k\tau-1}(x_n^{k\tau}) \geq \varepsilon \right\}. \tag{28}$$

Let z^j denote the jth element of the set $\{ y_n^{k\tau} \mid k \in \mathcal{K}_\varepsilon \}$. Note that the random variables $V^{k\tau-1}$ and $x_n^{k\tau}$ are measurable with respect to $\mathscr{F}_{k\tau-1} := \sigma((y^{k'})_{k' < k\tau})$, and thus so is $\mathbf{1}_{k \in \mathcal{K}_\varepsilon}$, from which $\tilde{y}_n^k$ is independent. Moreover, the σ-algebra generated by the past realisations of $\tilde{y}_n^k$ is included in $\mathscr{F}_{k\tau-1}$. This implies by Lemma A.3 that random variables $(z^j)_{j \in \mathbb{N}}$ are i.i.d. and share the same probability law as $\tilde{y}_n^0$.

According to the Strong Law of Large Numbers (Grimmett and Stirzaker [5, p. 294]) applied to the random sequence $(z^j)_{j \in \mathbb{N}}$, we should then have

$$\frac{1}{N} \sum_{j=1}^{N} z^j \xrightarrow{N \to +\infty} \mathbb{E}(z^1) = \mathbb{E}(\tilde{y}_n^0) = \mathbb{P}(\tilde{y}_n^0 = 1) > 0.$$

---

[5] Lemma 3.1(iii) provides a Lipschitz condition on V_m^k.

However, $\mathcal{K}_\varepsilon \cap \{\tilde{y}_n^k = 1\}$ is finite because of (27); thus, we know that there is only a finite number of indices j such that $z^j = 1$, the rest being equal to 0. So

$$\frac{1}{N} \sum_{j=1}^{N} z^j \xrightarrow{N \to +\infty} 0,$$

which is a contradiction. This shows that

$$V_n(x_n^{k\tau}) - V_n^{k\tau-1}(x_n^{k\tau}) \xrightarrow[\tilde{y}_n^k = 0]{k \to \infty} 0,$$

and monotonicity shows that

$$V_n(x_n^{k\tau}) - V_n^{k\tau}(x_n^{k\tau}) \xrightarrow[\tilde{y}_n^k = 0]{k \to \infty} 0,$$

which completes the induction. □

3.3. Application to known algorithms. To illustrate our result we will apply it to two well-known algorithms. For simplicity we will assume that the tree represents a T-step stochastic decision problem in which every leaf of the tree is of depth T.

We first define the CUPPS algorithm (Chen and Powell [1]) in this setting. Here at each major iteration we choose a $T - 1$-step scenario and compute the optimal trajectory while at the same time updating the value function for each node of the branch. In our setting, this uses a 1-admissible selection process $(y^k)_{k \in \mathbb{N}}$ defined by an i.i.d. sequence of random variables, with y^0 selecting a single branch of the tree. Theorem 3.1 shows that for every node n the upper and lower bound converges, that is

$$\lim_{k \to +\infty} \left[\sum_{m \in r(n)} \left(\frac{\Phi_m}{\Phi_n} W_m(x_n^k, u_m^k) \right) - V_n(x_n^k) \right] = 0$$

and

$$\lim_{k \to +\infty} \left[V_n(x_n^k) - V_n^k(x_n^k) \right] = 0.$$

We now place the SDDP algorithm (Pereira and Pinto [10]) and DOASA algorithm (Philpott and Guan [11]) in our framework. There are two phases in each major iteration of the SDDP algorithm—namely a forward pass and a backward pass of $T - 1$ steps. Given a current polyhedral outer approximation of the Bellman function $(V_n^{\tilde{k}-1})_{n \in \mathcal{N} \setminus \mathcal{L}}$, a major iteration $\tilde{k}$ of the SDDP algorithm consists in:

• Selecting uniformly a number N of scenarios ($N = 1$ for DOASA).

• Simulating the optimal strategy for the problem, that is solving problem (19) to determine a trajectory (for each scenario) $(x_{n_t}^{\tilde{k}})_{t \in \{0, \dots, T-1\}}$ where $(n_t)_{t \in \{0, \dots, T-1\}}$ defines one of the selected scenarios.

• For $t = T - 1$ down to $t = 0$, for each scenario solving problem (19) with $V_m^{\tilde{k}}$ instead of $V_m^{\tilde{k}-1}$, and defining

$$V_{n_t}^{\tilde{k}}(x) = \max\left\{ V_{n_t}^{\tilde{k}-1}(x), \theta_{n_t}^{\tilde{k}} + \langle \beta_{n_t}^{\tilde{k}}, x - x_{n_t}^{\tilde{k}} \rangle \right\}.$$

SDDP fits into our framework as follows. Given N, we define the $T - 1$-admissible selection process, $(y^{(T-1)k})_{k \in \mathbb{N}}$ by an i.i.d. sequence of random variables with y^0 selecting uniformly a set of N preleaves (i.e., nodes whose children are leaves) of the tree. Then for $\kappa \in \{1, \dots, T - 2\}$, $k \in \mathbb{N}$, $n \in \mathcal{N} \setminus \mathcal{L}$, we define

$$y_n^{k(T-1)+\kappa} := \begin{cases} 1 & \text{if there exist } m \in r(n) \text{ such that } y_m^{k(T-1)+\kappa-1} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

This algorithm is the same as SDDP with N randomly sampled forward passes per stage, but without the cut sharing feature used when random variables are stagewise independent. Since for every node n of the tree (except the leaves) there is a κ such that $\mathbb{P}(y_n^{k(T-1)+\kappa} = 1) > 0$, Theorem 3.1 guarantees the convergence of the lower bound for every node. This remains true when cuts are shared, since the proof of almost-sure convergence is unaffected by the addition of extra valid cutting planes at any point during the course of the algorithm. The proof of Theorem 3.1 gives

$$V_n(x_n^{k\tau}) - V_n^{k\tau}(x_n^{k\tau}) \xrightarrow[\tilde{y}_n^k = 0]{k \to \infty} 0,$$

RIGHTSLINK

and with shared cuts we obtain an improved value function $\check{V}_n^{k\tau}$ satisfying

$$V_n^{k\tau}(x_n^{k\tau}) \leq \check{V}_n^{k\tau}(x_n^{k\tau}) \leq V_n(x_n^{k\tau})$$

that must satisfy

$$V_n(x_n^{k\tau}) - \check{V}_n^{k\tau}(x_n^{k\tau}) \xrightarrow[\check{y}_n^k = 0]{k \to \infty} 0.$$

4. Discussion. The convergence result we have proved assumes that we compute new cuts at scenario tree nodes that are selected independently from the history of the algorithm. This enables us to use the Strong Law of Large Numbers in the proof. Previous results for multistage stochastic linear programming (Philpott and Guan [11]) require a selection process that visits each node in the tree infinitely often, which is a weaker condition than independence, since it follows from the Borel-Cantelli lemma (Grimmett and Stirzaker [5, p. 288]). An example would be the deterministic round robin selection mentioned in Philpott and Guan [11]. We do not have a proof of convergence for such a process in the nonlinear case. It is important to observe that the polyhedral form of V_t that was exploited in the proof of Philpott and Guan [11] is absent in our problem, and this difference could prove to be critical.

The convergence result is proved for a general scenario tree. In SDDP algorithms, the random variables are usually assumed to be stagewise independent (or made so by adding state variables). This means that the future cost functions $V_m(x)$ are the same at each node m at depth t. This allows cutting planes in the approximations to be shared across these nodes. As we have shown above, the convergence result we have shown here applies to this situation as a special case. It is worth noting that the class of algorithms covered by our result is larger than the examples presented in the literature. For example, an algorithm where we randomly select a node on the whole tree and then update backwards from there is proven to converge. One could also combine SDDP and CUPPS algorithms.

In the case where one would want to add cuts at different nodes in the tree in the update step of our procedure, the solving of the subproblems can be done in parallel. This is the case in CUPPS, where a whole branch of the tree is selected at each iteration. It also allows us to consider different selection strategies, where nodes at a given iteration could be selected throughout the tree depending on some criteria defined by the user. In the first few iterations, this could highly increase efficiency of the approximation and, because the solving of the subproblems can be parallelized, would not be very time consuming. One should bear in mind, however, that, at some point, the algorithm has to come back to an appropriate selection procedure, i.e., one that satisfies the independence assumption, to ensure convergence of the algorithm.

Appendix. Technical lemmas.

LEMMA A.1. *Suppose f is convex and $\mathscr{X}$ is compact, and suppose for any integer κ, the sequence of α-Lipschitz convex functions f^k, $k \in \mathbb{N}$ satisfies*

$$f^{k-\kappa}(x) \leq f^k(x) \leq f(x), \quad \text{for all } x \in \mathscr{X}.$$

Then for any infinite sequence $x^k \in \mathscr{X}$

$$\lim_{k \to +\infty} \left[f(x^k) - f^k(x^k) \right] = 0 \iff \lim_{k \to +\infty} \left[f(x^k) - f^{k-\kappa}(x^k) \right] = 0.$$

PROOF. If $\lim_{k \to +\infty}[f(x^k) - f^{k-\kappa}(x^k)] = 0$ then pointwise monotonicity of f^k shows that $\lim_{k \to +\infty}[f(x^k) - f^k(x^k)] = 0$. For the converse, suppose that the result is not true. Then there is some subsequence $(f^{k(l)})_{l \in \mathbb{N}}$ and $x^{k(l)} \in \mathscr{X}$ with

$$\lim_{k \to +\infty} \left[f(x^{k(l)}) - f^{k(l)}(x^{k(l)}) \right] = 0 \tag{29}$$

and $\varepsilon > 0, L \in \mathbb{N}$ with

$$f(x^{k(l)}) - f^{k(l)-\kappa}(x^{k(l)}) > \varepsilon$$

for every $l > L$. Since $\mathscr{X}$ is compact, we may assume (by taking a further subsequence) that $(x^{k(l)})_{l \in \mathbb{N}}$ converges to $x^* \in \mathscr{X}$. For sufficiently large l, the Lipschitz continuity of $f^{k(l)}$ and $f^{k(l)-\kappa}$ gives

$$\left| f^{k(l)}(x^*) - f^{k(l)}(x^{k(l)}) \right| \leq \alpha \|x^{k(l)} - x^*\| < \frac{\varepsilon}{4},$$

$$\left| f^{k(l)-\kappa}(x^{k(l)}) - f^{k(l)-\kappa}(x^*) \right| \leq \alpha \|x^{k(l)} - x^*\| < \frac{\varepsilon}{4},$$

and (29) implies that for sufficiently large l

$$f(x^{k(l)}) - f^{k(l)}(x^{k(l)}) < \frac{\varepsilon}{4}.$$

It follows that

$$
\begin{aligned}
f^{k(l)}(x^*) - f^{k(l)-\kappa}(x^*) &= f^{k(l)}(x^*) - f^{k(l)}(x^{k(l)}) \\
&\quad + f^{k(l)}(x^{k(l)}) - f(x^{k(l)}) \\
&\quad + f(x^{k(l)}) - f^{k(l)-\kappa}(x^{k(l)}) \\
&\quad + f^{k(l)-\kappa}(x^{k(l)}) - f^{k(l)-\kappa}(x^*) \\
&> \frac{\varepsilon}{4},
\end{aligned}
$$

since $f(x^{k(l)}) - f^{k(l)-\kappa}(x^{k(l)})$ is greater than ε, and the other three terms each have an absolute value smaller than $\varepsilon/4$. Consequently, $f^{k(l)}(x^*) > f^{k(l)-\kappa}(x^*) + \varepsilon/4$, for infinitely many l, which contradicts the fact that $f^k(x^*)$ is bounded above by $f(x^*)$. □

LEMMA A.2. *If $(y^k)_{k\in\mathbb{N}}$ is a τ-admissible selection process, then for all $k \in \mathbb{N}$, $\kappa \in \{0, \dots, \tau-1\}$, and all $n \in \mathcal{N}\backslash\mathcal{L}$ we have*

$$y_n^{k\tau+\kappa} = 1 \implies \begin{cases} x_n^{k\tau+\kappa} = x_n^{k\tau}, \\ V_n^{k\tau+\kappa-1} = V_n^{k\tau-1} & \text{if } k \geq 1. \end{cases}$$

PROOF.   Let n, k, and κ be such that $y_n^{k\tau+\kappa} = 1$. Let $a(n) := (n_0, n_1, \dots, n_t)$ be the sequence of ascendents of $n_t := n$—i.e., n_0 is the root node—and for all $t' < t$, $n_{t'} = p(n_{t'+1})$. Define the hypothesis $H(t, \kappa)$:
(a) $x_{n_t}^{k\tau+\kappa} = x_{n_t}^{k\tau}$.
(b) $V_{n_t}^{k\tau+\kappa-1} = V_{n_t}^{k\tau-1}$, if $t \geq 1$.
Let $\kappa' < \kappa$ and assume that for κ' and all $t' \leq t$, $H(t', \kappa')$ holds true. This is satisfied for $\kappa' = 0$. Let $t' < t$ and assume $H(t', \kappa'+1)$ is true. Since x_0 is fixed, this is satisfied for $t' = 0$. By definition of $u_{n_{t'+1}}^{k\tau+\kappa'+1}$ we have

$$u_{n_{t'+1}}^{k\tau+\kappa'+1} \in \operatorname*{arg\,min}_{u \in \tilde{\mathcal{U}}(x_{n_{t'}}^{k\tau+\kappa'+1})} \left\{ C_{n_{t'+1}}(x_{n_{t'}}^{k\tau+\kappa'+1}, u) + V_{n_{t'+1}}^{k\tau+\kappa'} \circ f_{n_{t'+1}}(x_{n_{t'}}^{k\tau+\kappa'+1}, u) \right\};$$

Thus, by $H(t', \kappa'+1)$ (a) we have

$$u_{n_{t'+1}}^{k\tau+\kappa'+1} \in \operatorname*{arg\,min}_{u \in \tilde{\mathcal{U}}(x_{n_{t'}}^{k\tau})} \left\{ C_{n_{t+1}}(x_{n_{t'}}^{k\tau}, u) + V_{n_{t'+1}}^{k\tau+\kappa'} \circ f_{n_{t+1}}(x_{n_{t'}}^{k\tau}, u) \right\}.$$

As $n_{t'+1}$ is an ascendent of n and $\kappa' < \kappa$ by property (i) of Definition 1, we have that the representation of $V_{n_{t'+1}}$ is not updated at iteration κ'; i.e.,

$$V_{n_{t'+1}}^{k\tau+\kappa'} = V_{n_{t'+1}}^{k\tau+\kappa'-1}.$$

And thus $H(t'+1, \kappa')$ (b) gives $H(t'+1, \kappa'+1)$ (b); i.e.,

$$V_{n_{t'+1}}^{k\tau+\kappa'} = V_{n_{t'+1}}^{k\tau-1}.$$

Therefore

$$u_{n_{t'+1}}^{k\tau+\kappa'+1} \in \operatorname*{arg\,min}_{u \in \tilde{\mathcal{U}}(x_{n_{t'}}^{k\tau})} \left\{ C_{n_{t+1}}(x_{n_{t'}}^{k\tau}, u) + V_{n_{t'+1}}^{k\tau-1} \circ f_{n_{t+1}}(x_{n_{t'}}^{k\tau}, u) \right\},$$

and consequently[6]

$$u_{n_{t'+1}}^{k\tau+\kappa'+1} = u_{n_{t'+1}}^{k\tau},$$

which gives by definition $H(t'+1, \kappa'+1)$ (a). Induction on t' gives $H(t', \kappa'+1)$ for all $t' \leq t$, and induction on κ' establishes $H(t, \kappa)$ for all $\kappa \in \{0, 1, \dots, \tau\}$.   □

LEMMA A.3.   *Let $(w^k)_{k\in\mathbb{N}}$ be a stochastic process with value in $\{0, 1\}$ adapted to a filtration $(\mathcal{F}_k)_{k\in\mathbb{N}}$, such that the number of terms that are nonzero is almost surely infinite. Let $(y^k)_{k\in\mathbb{N}}$ be a sequence of i.i.d discrete random variables. Define the filtration $\mathcal{B}_k := \sigma(\mathcal{F}_k \cup \sigma(y^1, \dots, y^{k-1}))$ and assume that for all $k \in \mathbb{N}$, y^k is independent of $\mathcal{B}_k$. Let $k(j)$ denote the jth integer such that $w^k = 1$—i.e., $k(0) = 0$—and for all $j > 0$,*

$$k(j) := \min\{l > k(j-1) \mid w^l = 1\}.$$

Finally, we define for all $j > 0$ the jth value of (y^k) such that $w^k = 1$; i.e.,

$$z^j := y^{k(j)}.$$

Then $(z^k)_{k\in\mathbb{N}}$ is a sequence of i.i.d. random variables equal in law to y^0.

---

[6] This requires that the choice of optimal control among the set of minimizers is deterministic (say that with minimum norm).

PROOF. Let Y denote the support of y^0. We start with z^1. For $i \in Y$,

$$
\begin{aligned}
\mathbb{P}(z^1 = i) &= \sum_{l=1}^{\infty} \mathbb{P}\big(\{\forall l' < l,\, w^{l'} = 0\} \cap \{w^l = 1\} \cap \{y^l = i\}\big) \quad \text{by } \{0,1\} \text{ definition}\\
&= \sum_{l=1}^{\infty} \mathbb{P}(\{y^l = i\})\,\mathbb{P}\big(\{\forall l' < l,\, w^{l'} = 0\} \cap \{w^l = 1\}\big) \quad \text{by independence}\\
&= \mathbb{P}(\{y^0 = i\}) \sum_{l=1}^{\infty} \mathbb{P}\big(\{\forall l' < l,\, w^{l'} = 0\} \cap \{w^l = 1\}\big) \quad \text{as } (y^l) \text{ is i.i.d.}\\
&= \mathbb{P}(\{y^0 = i\})
\end{aligned}
$$

as the sequence $(w^k)_{k \in \mathbb{N}}$ must contain a 1 almost surely. Thus z^1 is equal in law to y^0.

Now suppose that $\mathbf{z} = (z^1, \dots, z^m)$ is a sequence of i.i.d. random variables. Let $k_1, \dots, k_m$ be m ordered integers, and fix $\mathbf{b} \in \{0,1\}^n$ and $i \in Y$. We have

$$
\begin{aligned}
\mathbb{P}\big(\{\mathbf{z} = \mathbf{b}\} &\cap \{z^{m+1} = i\} \cap \{k(1) = k_1, \dots, k(m) = k_m\}\big)\\
&= \sum_{\nu=0}^{\infty} \mathbb{P}\big(\{\mathbf{z} = \mathbf{b}\} \cap \{k(1) = k_1, \dots, k(m) = k_m\} \cap \{y^{\nu} = i\} \cap \{\nu = k(m+1)\}\big)\\
&= \sum_{\nu=0}^{\infty} \mathbb{P}\big(y^{\nu} = i\big)\,\mathbb{P}(\{\mathbf{z} = \mathbf{b}\} \cap \{k(1) = k_1, \dots, k(m) = k_m\} \cap \{\nu = k(m+1)\})\\
&= \mathbb{P}(y^0 = i)\,\mathbb{P}\big(\{\mathbf{z} = \mathbf{b}\} \cap \{k(1) = k_1, \dots, k(m) = k_m\}\big).
\end{aligned}
$$

For the last equality we have used the fact that (y^k) is i.i.d. and the fact that $k(m+1)$ is almost surely finite and thus $(\{\nu = k(m+1)\})_{\nu \in \mathbb{N}}$ is a partition of the set of events.

Summing over the possible realisations of $k(1), \dots, k(m)$, we obtain

$$
\mathbb{P}\big(\{\mathbf{z} = \mathbf{b}\} \cap \{z^{m+1} = i\}\big) = \mathbb{P}(\mathbf{z} = \mathbf{b})\,\mathbb{P}(y_0 = i).
$$

Now summing over the possible realisations of $\mathbf{b}$ shows that z^{m+1} is equal in law to y^0. Thus

$$
\begin{aligned}
\mathbb{P}\big(\{\mathbf{z} = \mathbf{b}\} \cap \{z^{m+1} = i\}\big) &= \mathbb{P}\big(\{\mathbf{z} = \mathbf{b}\} \cap \{y^0 = i\}\big)\\
&= \mathbb{P}(\mathbf{z} = \mathbf{b})\,\mathbb{P}(y^0 = i)\\
&= \mathbb{P}(\mathbf{z} = \mathbf{b})\,\mathbb{P}(z^{m+1} = i)
\end{aligned}
$$

which shows that z^{m+1} is independent of $\mathbf{z}$ and equal in law to y^0. Induction over m completes the proof. $\square$

References

[1] Chen ZL, Powell WB (1999) A convergent cutting-plane and partial-sampling algorithm for multistage linear programs with recourse. *J. Optim. Theory Appl.* 102(3):497–524.

[2] Donohue CJ (1996) Stochastic network programming and the dynamic vehicle allocation problem. Unpublished doctoral dissertation, University of Michigan, Ann Arbor.

[3] Donohue CJ, Birge JR (2006) The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse. *Algorithmic Oper. Res.* 1:20–30.

[4] Finardi EC, da Silva EL (2006) Solving the hydro unit commitment problem via dual decomposition and sequential quadratic programming. *IEE Trans. Power Systems* 21(2):835–844.

[5] Grimmett GR, Stirzaker DR (1992) *Probability and Random Processes*, 2nd ed. (Oxford University Press, Oxford, U.K.).

[6] Hindsberger M, Philpott AB (2001) ReSa: A method for solving multi-stage stochastic linear programs. SPIX Stochastic Programming Symposium, Berlin.

[7] Kelley JE Jr (1960) The cutting-plane method for solving convex programs. *J. Soc. Indust. Appl. Math.* 8(4):703–712.

[8] Linowsky K, Philpott AB (2005) On the convergence of sampling-based decomposition algorithms for multistage stochastic programs. *J. Optim. Theory Appl.* 125:349–366.

[9] Nesterov Y (2004) *Introductory Lectures on Convex Optimization* (Kluwer, Boston).

[10] Pereira MVF, Pinto LMVG (1991) Multi-stage stochastic optimization applied to energy planning. *Math. Programming* 52(2):359–375.

[11] Philpott AB, Guan Z (2008) On the convergence of stochastic dual dynamic programming and related methods. *Oper. Res. Lett.* 36(4):450–455.

[12] Philpott AB, Guan Z (2011) A multistage stochastic programming model for the New Zealand dairy industry. *Internat. J. Production Econom.* 134(2):289–299.

[13] Ruszczyński A (2003) Decomposition methods. *Handbooks Oper. Res. Management Sci.* 10:141–211.

[14] Ruszczyński A (2011) *Nonlinear Optimization* (Princeton University Press, Princeton, NJ).

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