On the convergence of sampling-based methods for multi-stage stochastic linear programs

A. B. Philpott and Z. Guan[*][†]

September 6, 2007

Abstract

We discuss the almost-sure convergence of a broad class of sampling algorithms for multi-stage stochastic linear programs. Although the convergence of methods of this type is part of the stochastic programming folklore, we provide an explicit convergence proof based on the finiteness of the set of distinct cut coefficients. This differs from existing published proofs in that it does not require a restrictive assumption.

1 Introduction

Multistage stochastic linear programs with recourse are well known in the stochastic programming community, and are becoming more common in applications. The typical approach to solving these problems is to approximate the random variables using a finite set of outcomes forming a *scenario tree* and then solve a large-scale mathematical programming problem (see e.g. [1]). The scenario tree can be constructed to represent certain desired properties of the uncertain parameters (see e.g. [9]), or it can be (conditionally) sampled from some probability distribution (see e.g. [10]).

[*] Department of Engineering Science, The University of Auckland

[†] The authors would like to acknowledge discussions with Geoffrey Pritchard and Claudia Sagastizábal that have improved the presentation of this paper.

One approach to solving multistage stochastic linear programs that has been widely applied in energy and logistics settings is based on the stochastic dual dynamic programming (SDDP) algorithm of Pereira and Pinto [8]. This algorithm constructs feasible dynamic programming 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 (see [2],[3],[4],[6]) that are based on the same essential idea, but seek to improve the method by exploiting the structure of particular applications.

Since its publication in 1991, a number of authors have studied the convergence behaviour of SDDP and related algorithms. In his PhD thesis [3] (and in [4]) Donohue states that "finite convergence of this algorithm follows from the finite convergence of the Nested Decomposition algorithm, since the scenarios from which the optimality cuts are generated are resampled at each iteration." This remark which, strictly speaking, should be a statement of convergence with probability 1, is not accompanied by a formal proof.

The first formal proof of the almost sure convergence of multi-stage sampling algorithms was published by Chen and Powell [2] who derived this for their CUPPS algorithm. This proof was extended by Linowsky and Philpott [7] to cover other multi-stage sampling algorithms (SDDP [8], AND [4], ReSa [6]) that use outer approximation. The convergence proofs in [2] and [7] make use of an unstated assumption regarding the independence of sampled random variables and convergent subsequences of algorithm iterates. This assumption seriously weakens the analysis in these papers, and leaves open the question of convergence in general.

The purpose of this paper is to define a broad class of sampling algorithms that include SDDP, AND, ReSa and CUPPS and give a simpler proof of almost sure convergence that does not require this assumption. Our proof follows the finiteness argument that Donohue alluded to in his thesis, and makes this argument explicit, by basing it on two well-defined properties of the sampling procedure.

In the next section we define the class of multi-stage stochastic linear programs that we shall be considering and the class of algorithms for these problems. We then discuss in section 3 the implicit assumption that is made in the proofs in [2] and [7], and give a new proof of almost sure convergence that does not require this assumption, but requires conditions on the forward and backward sampling procedures that are easily shown to hold for all the algorithms discussed above. In section 4, we give a discussion on the role of these conditions and their relationship to the properties used in Linowsky and Philpott.

2 Multistage Benders decomposition

We follow the notation and terminology of [7], and restrict attention to multistage stochastic programs with the following properties:

(A1) Random quantities appear only on the right-hand side of the linear constraints in each stage.

(A2) The set Ω_t of random outcomes in each stage $t = 2, 3, \ldots, T$ is discrete and finite
($\Omega_t = \{\omega_{ti} | \ i = 1, \ldots, q_t < \infty\}$ with probabilities $p_{ti} > 0$, $\forall i$).

(A3) Random quantities in different stages are independent.

(A4) The feasible region of the linear program in each stage is non-empty and bounded.

Under these assumptions, the multi-stage stochastic linear program can be written in the following form:

Solve the problem defined by

$$\begin{array}{rl} [LP_1] & Q_1 = \min_{x_1} c_1^\top x_1 + \mathcal{Q}_2(x_1) \\ \text{subject to} & A_1 x_1 = b_1, \\ & x_1 \geq 0, \end{array}$$

where for all $t = 2, \ldots, T$,

$$\mathcal{Q}_t(x_{t-1}) = \sum_{i=1}^{q_t} p_{ti} Q_t(x_{t-1}, \omega_{ti}),$$

$Q_t(x_{t-1}, \omega_{ti})$ is defined by the problem

$$\begin{array}{rl} [LP_t] & Q_t(x_{t-1}, \omega_t) = \min_{x_t} c_t^\top x_t + \mathcal{Q}_{t+1}(x_t) \\ \text{subject to} & A_t x_t = \omega_t - B_{t-1} x_{t-1}, \\ & x_t \geq 0, \end{array}$$

and we set $\mathcal{Q}_{T+1} \equiv 0$.

The problem $[LP_t]$ depends on the choice of ω_t and x_{t-1}, and so we could write $[LP_t(x_{t-1}, \omega_t)]$, though we choose to suppress this dependence in the notation. By Assumption (A3), $[LP_t]$ is independent of $\omega_{t-1}, \omega_{t-2} \ldots.$

In the class of sampling algorithms that we consider in this paper the functions $\mathcal{Q}_t(x_{t-1})$ in each stage are approximated by the maximum of a collection of linear functions, each of which is called a *cut*. In each iteration $k = 1, 2, \ldots,$ the type of algorithm we consider computes a set of feasible solutions $\{x_t^k : t = 1, \ldots, T-1\}$, and a set of cuts, one for each stage $t = 1, \ldots, T-1$. This gives rise to a sequence of approximate problems $[AP_t^k]$, $k = 1, 2, \ldots$, for each stage. These are defined as follows:

For $t = 1$, we solve the linear program

$$\begin{array}{rll} [AP_1^k] & C_1^k = \min_{x_1, \theta_2} c_1^\top x_1 + \theta_2 & \\ \text{subject to} & A_1 x_1 = b_1, & \\ & \theta_2 + (\beta_2^j)^\top x_1 \geq \alpha_{2,j}, & j = 0, \ldots, k-1, \\ & x_1 \geq 0, & \end{array}$$

and, for $t = 2, \ldots, T-1$, we solve

$$\begin{array}{rll} [AP_t^k] & C_t^k(x_{t-1}^k, \omega_t) = \min_{x_t, \theta_{t+1}} c_t^\top x_t + \theta_{t+1} & \\ \text{subject to} & A_t x_t = \omega_t - B_{t-1} x_{t-1}^k, & \\ & \theta_{t+1} + (\beta_{t+1}^j)^\top x_t \geq \alpha_{t+1,j}, & j = 0, \ldots, k-1, \\ & x_t \geq 0. & \end{array}$$

Finally for every k, we set $[AP_T^k]$=$[LP_T]$. The problems $[AP_t^k]$ are approximations of $[LP_t]$ in the sense that $\mathcal{Q}_{t+1}(x_t)$ is approximated (below) by the polyhedral function

$$\max_{j=0,\ldots,k-1} \{\alpha_{t+1,j} - (\beta_{t+1}^j)^\top x_t\}.$$

This means that any solution to $[AP_t^k]$ has a value that is a lower bound on the optimal value of $[LP_t]$.

For all stages, the first cut ($j = 0$) is set as the trivial cut $\theta_{t+1} \geq -\infty$. We use the notation $\mathcal{C}_t^k(x_{t-1})$ to denote $\sum_{i=1}^{q_t} p_{ti} C_t^k(x_{t-1}, \omega_t)$. In the last stage, T, we have $[AP_T^k]$= $[LP_T]$, and so for every x_{T-1} and ω_T

$$C_T^k(x_{T-1}, \omega_T) = Q_T(x_{T-1}, \omega_T), \quad k = 1, 2, \ldots .$$

Since cuts are added from one iteration to the next, and no cuts are taken out, the optimal values of $[AP_t^k]$ form a monotonic sequence, i.e. for $k = 1, 2, \ldots$

$$C_t^{k+1}(x_{t-1}, \omega_t) \geq C_t^k(x_{t-1}, \omega_t), \qquad t = 2, 3, \ldots, T,$$

and

$$C_1^{k+1} \geq C_1^k.$$

Observe that under Assumption (A4),

$$\{x_t \mid A_t x_t = \omega_t - B_{t-1} x_{t-1}^k, \quad x_t \geq 0\}$$

is nonempty and bounded so $[AP_t^k]$ always has a nonempty feasible set (with θ_{t+1} chosen large enough) and hence an optimal solution. Thus its dual has an optimal solution (π_t, ρ_t), where π_t corresponds to the equality constraints, and ρ_t corresponds to the cut constraints. Furthermore, by Assumption (A1), the set of extreme points of the dual of $[AP_t^k]$ is independent of the outcomes of the random quantities, which allows us to construct a valid cut at each stage based on an assembled collection $\mathcal{D}_t^k$ of extreme-point dual solutions from different samples.

Initially at iteration $k = 0$, $\mathcal{D}_t^0 = \emptyset$. At any subsequent iteration k the coefficients of the cuts at each stage $t = 1, 2, \ldots, T-1$, are calculated as follows.

Cut Calculation Algorithm (CCA)

1. Choose a sample $\Omega_t^k \subseteq \Omega_t$, solve $[AP_t^k]$ for all $\omega_{ti} \in \Omega_t^k$, and add the optimal extreme-point dual solutions to $\mathcal{D}_t^k$.

2. Let $(\pi_t^i(x_{t-1}^k), \rho_t^i(x_{t-1}^k))$ be the best dual solution in $\mathcal{D}_t^k$ for $[AP_t^k]$ for each $\omega_{ti} \in \Omega_t$, that is,

$$\begin{aligned} & (\pi_t^i(x_{t-1}^k), \rho_t^i(x_{t-1}^k)) \\ = \ & \arg\max\{\pi_t^\top(\omega_{ti} - B_{t-1}x_{t-1}^k) + \rho_t^\top \alpha_{t+1}^{k-1} \mid (\pi_t, \rho_t) \in \mathcal{D}_t^k\}. \end{aligned}$$

3. The cut has the formula

$$\theta_t \geq \alpha_{t,k} - (\beta_t^k)^\top x_{t-1}$$

where

$$\begin{aligned}
\beta_t^k &= \sum_{i=1}^{q_t} p_{ti}\ B_{t-1}^\top\ \pi_t^i(x_{t-1}^k) && \text{for } 2 \leq t \leq T, \\
\alpha_{t,k} &= \sum_{i=1}^{q_t} p_{ti} \left[\omega_{ti}^\top\ \pi_t^i(x_{t-1}^k) + (\alpha_{t+1}^{k-1})^\top \rho_t^i(x_{t-1}^k)\right] && \text{for } 2 \leq t \leq T-1, \\
\alpha_{T,k} &= \sum_{i=1}^{q_T} p_{Ti}\ \omega_{Ti}^\top\ \pi_T^i(x_{T-1}^k).
\end{aligned}$$

Observe that $\alpha_{t,k}$ is a scalar, whereas α_{t+1}^{k-1} denotes a $(k-1)$-dimensional vector. This means that the dimensions of α_{t+1}^{k-1} and $\rho_t^i(x_{t-1}^k)$ are increasing as the iteration count k increases, and thus the collection of extreme-point solutions of the dual of $[AP_t^k]$ may be infinite. On the other hand, the collection of distinct values of $(\beta_t^k, \alpha_{t,k})$ is provably finite, as we show in the following lemma.

Lemma 1 *For each $t = 2, 3, \ldots, T$, define the set*

$$\mathcal{G}_t^k = \{(\beta_t^j, \alpha_{t,j}) : j = 1, 2, \ldots, k-1\}.$$

Then for any sequence $\mathcal{G}_t^k$, $k = 1, 2, \ldots$ generated by the repeated application of CCA there exists m_t such that for all k

$$\left|\mathcal{G}_t^k\right| \leq m_t.$$

Furthermore, there exists k_t, so that if $k > k_t$ then $\mathcal{G}_t^k = \mathcal{G}_t^{k_t}$.

Proof. Consider any realization of the sequence $\mathcal{G}_t^k$, $k = 1, 2, \ldots$ generated by the repeated application of CCA. We use induction on t to construct m_t such that $\left|\mathcal{G}_t^k\right| \leq m_t$. The second part of the lemma follows immediately. First at T, $\rho_T = 0$ and π_T is an extreme point of $\{\pi \mid A_T^\top \pi \leq c_T\}$ of which there are at most m_{T+1}, say. Then the cut coefficients

$$\alpha_{T,k} = \sum_{i=1}^{q_T} p_{Ti}\ \omega_{Ti}^\top\ \pi_T^i(x_{T-1}^k),$$

$$\beta_T^k = \sum_{i=1}^{q_T} p_{Ti}\ B_{T-1}^\top\ \pi_T^i(x_{T-1}^k),$$

each can only take at most $m_{T+1}^{q_T}$ values, and thus if $m_T = m_{T+1}^{2q_T}$, then for all k

$$\left|\mathcal{G}_T^k\right| \leq m_T.$$

Now suppose at t that there exists m_{t+1} such that for all k

$$\left|\mathcal{G}_{t+1}^k\right| \leq m_{t+1}.$$

It follows that there exists k_{t+1}, so that if $k > k_{t+1}$ then $\mathcal{G}_{t+1}^k = \mathcal{G}_{t+1}^{k_{t+1}}$ and the cut at iteration $k > k_{t+1}$ is a repeat of some cut in the existing cuts. Consider the feasible region of the dual of $[AP_t^k]$, namely

$$\mathcal{H}_t^k = \{(\pi_t, \rho_t) \mid A_t^\top \pi_t + \sum_{j=1}^{k-1} \beta_{t+1}^j \rho_t^j \leq c_t, \quad \sum_{j=1}^{k-1} \rho_t^j = 1, \quad \rho_t \geq 0\}.$$

If $k > k_{t+1}$ then any extreme point (π_t^k, ρ_t^k) of $\mathcal{H}_t^k$ corresponds to an extreme point (π, ρ) of $\mathcal{H}_t^{k_{t+1}}$ with the same dual objective value, obtained by choosing $\pi = \pi_t^k$ and basic columns β_{t+1}^j for $j < k_{t+1}$ that match the basic columns β_{t+1}^j, $k_{t+1} \leq j < k$. This is because each latter column β_{t+1}^j and its cost coefficient $\alpha_{t+1,j}$ is a duplicate of some $(\beta, \alpha) \in \mathcal{G}_{t+1}^{k_{t+1}}$. Since there are a finite number, say e_t, of extreme point solutions to $\mathcal{H}_t^{k_{t+1}}$, there are at most e_t distinct values of

$$\left[\omega_{ti}^\top \ \pi_t^i(x_{t-1}^k) + (\alpha_{t+1}^{k-1})^\top \rho_t^i(x_{t-1}^k)\right]$$

and so $(e_t)^{q_t}$ distinct values of

$$\alpha_{t,k} = \sum_{i=1}^{q_t} p_{ti} \left[\omega_{ti}^\top \ \pi_t^i(x_{t-1}^k) + (\alpha_{t+1}^{k-1})^\top \rho_t^i(x_{t-1}^k)\right],$$

Similarly,

$$\beta_t^k = \sum_{i=1}^{q_t} p_{ti} \ B_{t-1}^\top \ \pi_t^i(x_{t-1}^k),$$

can take at most $(e_t)^{q_t}$ values and so if $m_t = (e_t)^{2q_t}$ then

$$\left|\mathcal{G}_t^k\right| \leq m_t,$$

which proves the result. ■

Lemma 1 states that in any realization of the algorithm there will exist finite m_t and k_t independent of k. Observe however that in the case that Ω_t^k is randomly sampled, m_t and k_t are random variables with distribution determined by the sampling distribution. So they could be arbitrarily large.

Linowsky and Philpott [7] define a class of sampling-based decomposition algorithms, the Multi-stage Sampled Benders Decomposition (MSBD), which includes SDDP, AND, ReSa and CUPPS. Here we define a different class of sampling algorithms, which we call Dynamic Outer Approximation Sampling Algorithms (DOASA). To do this we will use the terminology *scenario* to denote an element of $\prod_{t=2}^{T-1} \Omega_t$ indexed by j so

$$\prod_{t=2}^{T-1} \Omega_t = \{\omega(j) \mid j = 1, 2, \dots, \prod_{t=2}^{T-1} q_t\}.$$

Algorithms in the DOASA class perform the following steps:

Step 0: (Initialization) Set $k = 1$.

Step 1: (Forward pass)
Sample a single outcome ω_t of the random variable in each stage $t = 2, 3, \dots, T-1$, to give a single scenario $\{\omega_t^k\}$. For each stage $t = 1, 2, \dots, T-1$, compute the primal solution (x_t^k, θ_{t+1}^k) of the problem $[AP_t^k]$.

Step 2: (Cut Generation)
For each stage $t = T, T-1, \dots, 2$, generate a cut at x_{t-1}^k with a sample Ω_t^k.

Step 3: Set $k = k + 1$ and go to Step 1.

Algorithms in the DOASA class also require the following properties of the sampling methods used to obtain $\{\omega_t^k\}$ and Ω_t^k:

Forward Pass Sampling Property (FPSP):
For each $j = 1, 2, \dots, \prod_{t=2}^{T-1} q_t$, with probability 1

$$\left|\{k : \{\omega_t^k \mid t = 2, 3, \dots, T-1\} = \omega(j)\}\right| = \infty.$$

Backward Pass Sampling Property (BPSP):
For each $t = 2, 3, \ldots, T$ and $i = 1, 2, \ldots, q_t$, with probability 1

$$\left|\left\{k : \omega_{ti} \in \Omega_t^k\right\}\right| = \infty.$$

FPSP states that each scenario $\omega(j)$ is traversed infinitely many times with probability 1 in the forward pass. BPSP states that each scenario outcome ω_{ti} is visited infinitely many times with probability 1 in the backward pass. There are many sampling methods satisfying these two properties. For example, consider independently sampling a single outcome in each stage with a positive probability for each ω_{ti} in the forward pass and backward pass respectively. Then by the Borel-Cantelli lemma (see [5]) this method satisfies both properties. Another sampling method that satisfies FPSP and BPSP is to repeat an exhaustive enumeration of each scenario $\omega(j)$, $j = 1, 2, \ldots, \prod_{t=2}^{T-1} q_t$ in both the forward pass and the backward pass, although such a method would be prohibitively expensive in all but the smallest examples.

3 Convergence of DOASA algorithms

3.1 Previous results

Previous published results in [2] and [7] give proofs for the almost sure convergence of CUPPS and MSBD respectively. The proofs in both of these papers require an important but unstated assumption. Here we state this assumption formally and discuss it.

Let the iterations of the algorithm be indexed by $\mathcal{N} = \{1, 2, \ldots\}$ and suppose $t \in \{1, \ldots, T-1\}$. Let $\{\omega_t^n, x_t^n\}_{n \in \mathcal{N}}$ be the sequence generated by the sampling algorithm at stage t.

Assumption 1: For any infinite subsequence $\{x_t^k\}_{k \in \mathcal{K}}$ of $\{x_t^n\}_{n \in \mathcal{N}}$ there exists a convergent subsequence $\{x_t^j\}_{j \in \mathcal{J}}$ that is independent of $\{\omega_{t+1}^j\}_{j \in \mathcal{J}}$.

Remark 4.1 in [2] correctly claims that if $\mathcal{N}$ is infinite then with probability one $\mathcal{N}$ has an infinite subset $\mathcal{N}_{ti}$ corresponding to draws of outcome ω_{ti} for any $i = 1, \ldots, q_t$ and $t = 2, \ldots, T$. This follows by an application of the Borel-Cantelli lemma, because each ω_t^n in $\{\omega_t^n\}_{n \in \mathcal{N}}$ is independently sampled and $\Pr[\omega_t^n = \omega_{ti}] > 0$.

However, the situation becomes more subtle in the proof of Lemma 5.2 in [2]. Here the authors claim that for any infinite subset $\mathcal{K}$ of $\mathcal{N}$, there exists an infinite subset $\mathcal{J}$ with a convergent subsequence $\{x^j_{T-1}\}_{j\in\mathcal{J}}$ such that with probability one there exists an infinite subset $\mathcal{J}_i$ of $\mathcal{J}$ corresponding to draws of each sample ω_{Ti} for $i = 1, \ldots, q_T$. The convergent subsequence $\{x^j_{T-1}\}_{j\in\mathcal{J}}$ in this lemma is constructed using the assumed compactness of the set X in which x_{T-1} lies. Of course, compactness guarantees a convergent subsequence $\{x^j_{T-1}\}_{j\in\mathcal{J}}$ of $\{x^k_{T-1}\}_{k\in\mathcal{K}}$, but it cannot be deduced from this and Remark 4.1 in [2] that there are infinite number of ω_{Ti} in $\{\omega^j_T\}_{j\in\mathcal{J}}$ for every $i = 1, \ldots, q_T$. (The problem here is that for every *convergent* subsequence it might be the case that there are only finitely many ω_{Ti} for some $i = 1, \ldots, q_T$, and this possibility needs to be ruled out somehow.)

In claiming the independence of the sampling procedure from the convergence of the subsequence, the authors of [2] are making an implicit assumption (Assumption 1), which is needed to make the proof of Lemma 5.2 valid. The proof in [7] is based on Lemma 5.2 in [2], and so it is also flawed in the absence of Assumption 1.

In this section we give a direct proof of almost sure convergence that does not rely on Assumption 1. The new proof formalizes the assertion by Donohue [3] that convergence follows from resampling. It also clarifies the role that extreme-point dual solutions play in the almost sure convergence of these sampling algorithms.

3.2 Single-scenario Multistage Benders Decomposition

To demonstrate the convergence of our class of sampling methods it is helpful to first understand the convergence of an algorithm that uses a single scenario. This algorithm will construct a cut for every $x^k_t, \quad t = 1, \ldots, T-1, \quad k = 1, 2, \ldots$ that is visited by simulating the solution forward over a single sample scenario $\omega(j)$ that remains the same throughout the course of the algorithm. We call this algorithm SSMBD.

SSMBD

Step 0: (Initialization) Set iteration counter $k = 1$. Select at each stage $t = 2, 3, \ldots, T-1$, a single outcome ω_t of the random variable to give a single scenario.

Step 1: (Forward pass)

For each stage $t = 1, 2, \ldots, T-1$, solve $[AP_t^k]$ to yield the primal solution (x_t^k, θ_{t+1}^k).

Step 2: (Cut Generation)

For each stage $t = T, T-1, \ldots, 2$, generate a cut at x_{t-1}^k with a sample Ω_t^k.

Step 3: Set $k = k+1$ and go to Step 1.

We can apply Lemma 1 to give the following result.

Lemma 2 *Under every realization of iterations, SSMBD converges in a finite number of iterations to a policy giving* $\lim_k C_1^k$ *which is at most equal to the optimal expected cost of [*LP_1*].*

Proof. Under every realization of iterations, by Lemma 1, for $t \in \{2, \ldots, T\}$, there exists k_t, so that if $k > k_t$ then $\mathcal{G}_t^k = \mathcal{G}_t^{k_t}$ and thus there is no further change in the cuts defining $\mathcal{C}_t^k(x_{t-1})$, that is, for every x_{t-1}

$$\max_{j=0,\ldots,k-1}\{\alpha_{t,j} - (\beta_t^j)^\top x_{t-1}\} = \max_{j=0,\ldots,k_t-1}\{\alpha_{t,j} - (\beta_t^j)^\top x_{t-1}\}.$$

Thus all solutions (x_1^k, θ_2^k) to $[AP_1^k]$ are the same for $k > k_2$, as are all solutions (x_t^k, θ_{t+1}^k) to $[AP_t^k]$, $t = 2, 3, \ldots, T$, so the SSMBD algorithm terminates after iteration k_2.

Since any solution to $[AP_1^k]$ has a value that is a lower bound on the optimal value of $[LP_1]$, this will be true at termination of SSMBD. ■

Remark 3.1: The solution obtained from SSMBD defines a set of cuts at each stage. If for every k, $\Omega_t^k = \Omega_t$ then this set of cuts at termination will be the same every time the algorithm is run (assuming the single scenario $\omega(j)$ remains fixed). On the other hand if Ω_t^k is a random sample then the set of cuts will also be random, and defined by the sampling distribution. Every time the algorithm is run (with different random number seeds) we should expect to obtain a different sequence of cuts.

Remark 3.2: The cuts at stage 1 define a lower bound on the expected cost of any policy. Every time the algorithm is run, with a possibly different

$\omega(j)$, this lower bound will be (possibly) different. However every realization of this value will be a lower bound on the expected cost of any policy, and so the maximum of these values will be the best lower bound of those available.

Remark 3.3: The solution obtained from SSMBD is not the same as the optimal solution of the mathematical program obtained by using a single scenario and solving a deterministic problem. The latter solution would define a single set of actions, one for each stage t, that may not be feasible for some scenarios in the original problem. On the other hand, the SSMBD solution is a set of (possibly) random cuts defining a policy that is feasible for the original problem. The simulation of this policy using a randomly sampled forward pass, yields a random value having an expectation that is greater than or equal to the optimal expected cost of the underlying stochastic program. A simulation of the policy with the single scenario used in SSMBD gives the cost of the policy when implemented in the single scenario. The observed value of this simulation depends on the outcomes in the single scenario. This means that it may be significantly lower or significantly higher than the true expected cost of the policy.

3.3 Multiscenario Multistage Benders Decomposition

We now consider a multiple-scenario version of SSMDB called MSMBD. In this version a finite set of N scenarios is sampled in advance. The algorithm then constructs an optimal solution corresponding to a scenario tree consisting of these scenarios.

MSMBD

Step 0: (Initialization) Set $k = 1$. For $s = 1$ to N, select at each stage $t = 2, 3, \ldots, T-1$, a single outcome ω_{st} of the random variable to give a set of N scenarios.

Step 1: (Forward pass)
For each scenario s, and stage $t = 1, 2, \ldots, T-1$, compute the primal solution $(x_{st}^k, \theta_{s,t+1}^k)$ of the problem $[AP_t^k]$.

Step 2: (Cut Generation)
For each stage $t = T, T-1, \ldots, 2$, generate N cuts at the states $x_{s,t-1}^k$ with samples $\Omega_{s,t}^k$, $s = 1, 2, \ldots, N$.

Step 3: Set $k = k + 1$ and go to Step 1.

Lemma 3 *In every realization of iterations, MSMBD converges in a finite number of iterations to a policy giving value* $\lim_k C_1^k$ *which is at most equal to the optimal expected cost of [*LP_1*].*

Proof. The proof is similar to that for SSMBD. For each $s = 1, 2, \ldots, N$, since $k = 1, 2, \ldots$ and one cut is constructed in each iteration k, then by Lemma 1, for $t \in \{2, \ldots, T\}$, there exists $k_{s,t}$, so that if $k > k_{s,t}$ then $\mathcal{G}_t^k = \mathcal{G}_t^{k_{s,t}}$ and thus there is no further change in the cuts defining $\mathcal{C}_{s,t}^k(x_{s,t-1})$, that is, for every $x_{s,t-1}$

$$\max_{j=0,\ldots,k-1}\{\alpha_{t,j} - (\beta_t^j)^\top x_{s,t-1}\} = \max_{j=0,\ldots,k_{s,t}-1}\{\alpha_{t,j} - (\beta_t^j)^\top x_{s,t-1}\}.$$

For $t \in \{2, \ldots, T\}$, if we choose $k_t = \max_{s=1}^N\{k_{s,t}\}$, then for each $k > k_t$ there is no change in the cuts defining $\mathcal{C}_t^k(x_{t-1})$, that is, for every x_{t-1}

$$\max_{j=0,\ldots,k-1}\{\alpha_{t,j} - (\beta_t^j)^\top x_{t-1}\} = \max_{j=0,\ldots,k_t-1}\{\alpha_{t,j} - (\beta_t^j)^\top x_{t-1}\}.$$

Thus all solutions (x_1^k, θ_2^k) to $[AP_1^k]$ are the same for $k > k_2$, as are all solutions (x_t^k, θ_{t+1}^k) to $[AP_t^k]$, $t = 2, 3, \ldots, T$, so the MSMDB algorithm terminates after iteration k_2.

It is easy to see that for every k the optimal value of $[AP_1^k]$ is a lower bound on the optimal expected cost of $[LP_1]$. ■

The algorithm MSMBD works with N scenarios that do not change over the course of the algorithm. All of the remarks that were made for SSMBD apply in this case also. In particular we observe that a termination criterion that uses the N scenarios to simulate the candidate policy defined by the cuts might give a misleading indication of convergence. Lemma 3 demonstrates that MSMBD will terminate at some policy that gives a lower bound on the optimal expected cost of $[LP_1]$. Simulating this policy using a set of randomly sampled scenarios will give a statistical estimate of an upper bound on the optimal expected cost of $[LP_1]$. If the sample of N scenarios is small then we might expect termination of MSMBD at a poor policy. In this case the standard termination criterion using the statistical estimate of the upper bound might fail to be met, even though no further improvement in the policy is possible by continuing to iterate MSMBD.

A special case of MSMBD uses the universe of $N = \prod_{t=2}^{T-1} q_t$ scenarios.

Lemma 4 *Under BPSP, MSMBD with the universe of scenarios converges with probability 1 to an optimal solution to [LP_1] in a finite number of iterations.*

Proof. From Lemma 3 in every realization of iterations MSMBD will converge in a finite number of steps to a policy that has $\lim_k C_1^k$ giving a lower bound on the true expected cost. Now consider a realization of MSMBD iterations, and denote the limiting policy by $(\bar{x}_1, \bar{x}_2(\omega_2), \bar{x}_3(\omega_2, \omega_3), \dots)$, which is obtained at iteration $\bar{k}$, say. For any scenario $\omega_2, \omega_3, \dots, \omega_T$, we denote $\bar{x}_t(\omega_2, \dots, \omega_t)$ by $\bar{x}_t(\omega)$. We claim that for every $k > \bar{k}$, and any scenario ω,

$$\mathcal{C}_T^k(\bar{x}_{T-1}(\omega)) = \mathcal{Q}_T(\bar{x}_{T-1}(\omega)), \tag{1}$$

with probability 1, which implies $C_T^k(\bar{x}_{T-1}(\omega), \omega_T) = Q_T(\bar{x}_{T-1}(\omega), \omega_T)$ for all ω_T. Otherwise for some particular outcome $\hat{\omega}_T$, we have $\hat{\omega}_T \notin \Omega_t^k$, for every $k > \bar{k}$, with positive probability which violates BPSP.

Now we claim that if $k > \bar{k}$ then for every scenario ω

$$\mathcal{C}_{T-1}^k(\bar{x}_{T-2}(\omega)) = \mathcal{Q}_{T-1}(\bar{x}_{T-2}(\omega)). \tag{2}$$

Otherwise for some particular outcome $\hat{\omega}_{T-1}$,

$$C_{T-1}^k(\bar{x}_{T-2}(\omega), \hat{\omega}_{T-1}) < Q_{T-1}(\bar{x}_{T-2}(\omega), \hat{\omega}_{T-1}). \tag{3}$$

But

$$\begin{array}{rll} C_{T-1}^k(\bar{x}_{T-2}(\omega), \hat{\omega}_{T-1}) = & \min_{x_{T-1}, \theta_T} c_{T-1}^\top x_{T-1} + \theta_T & \\ \text{subject to} & A_{T-1}x_{T-1} = \hat{\omega}_{T-1} - B_{T-2}\bar{x}_{T-2}(\omega), & \\ & \theta_T + (\beta_T^j)^\top x_{T-1} \geq \alpha_{T,j}, & j = 0, \dots, k-1, \\ & x_{T-1} \geq 0, & \end{array}$$

which has optimal solution

$$(x_{T-1}^*, \theta_T^*) = (\bar{x}_{T-1}(\omega), \max_{j=0,\dots,k-1}\{\alpha_{T,j} - (\beta_T^j)^\top \bar{x}_{T-1}(\omega)\})$$

with $\omega_{T-1} = \hat{\omega}_{T-1}$.

If $\theta_T^* < \mathcal{C}_T^k(x_{T-1}^*)$, then for any $k > \bar{k}$

$$\max_{j=0,\dots,k-1}\{\alpha_{T,j} - (\beta_T^j)^\top \bar{x}_{T-1}(\omega)\}) < \mathcal{C}_T^k(x_{T-1}^*) = \mathcal{Q}_T(\bar{x}_{T-1}(\omega)) \tag{4}$$

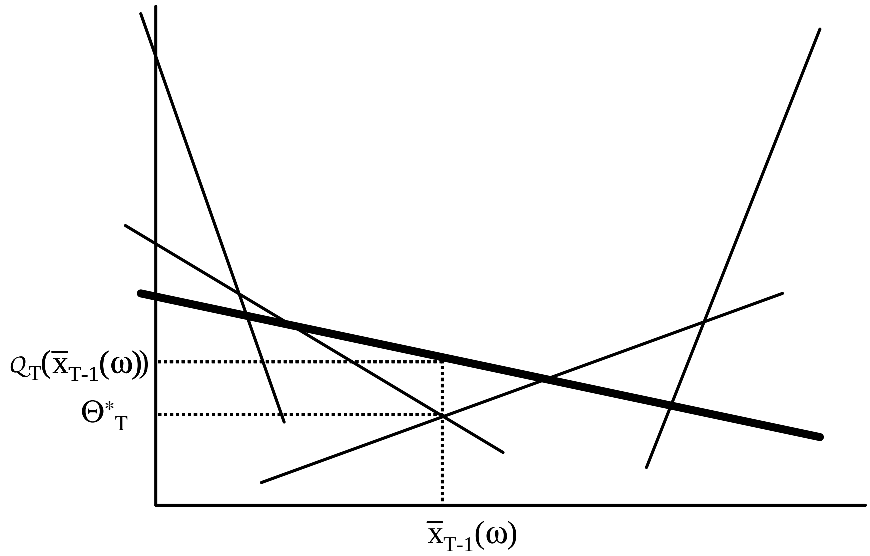

Figure 1: A new cut shown in bold would be created if $\theta_T^* < \mathcal{C}_T^k(x_{T-1}^*)$.

by (1). But by BPSP we have with probability 1 that for each ω_T there is some $k(\omega_T) > \bar{k}$ with $\omega_T \in \Omega_T^{k(\omega_T)}$. If we let $\hat{k}$ denote the maximum of the $k(\omega_T)$ then the height of the cut at $\bar{x}_{T-1}(\omega)$ evaluated at iteration $\hat{k}$ is $\mathcal{Q}_T(\bar{x}_{T-1}(\omega))$ contradicting (4) (see Figure 1). Thus we have

$$\theta_T^* = \mathcal{C}_T^k(x_{T-1}^*) = \mathcal{Q}_T(x_{T-1}^*)$$

and

$$C_{T-1}^k(\bar{x}_{T-2}(\omega), \hat{\omega}_{T-1}) = c_{T-1}^\top x_{T-1}^* + \mathcal{Q}_T(x_{T-1}^*) = Q_{T-1}(\bar{x}_{T-2}(\omega), \hat{\omega}_{T-1})$$

contradicting (3), thereby demonstrating (2). Observe that since $\hat{\omega}_{T-1}$ was arbitrary this shows that $\bar{x}_{T-1}(\omega)$ solves $[LP_{T-1}(\bar{x}_{T-2}(\omega), \omega_{T-1})]$ for any ω_{T-1}.

In a similar way, it is easy to show by induction that $\bar{x}_{t-1}(\omega)$ solves $[LP_{t-1}(\bar{x}_{t-2}(\omega), \omega_{t-1})]$ thus demonstrating that $(\bar{x}_1, \bar{x}_2(\omega_2), \bar{x}_3(\omega_2, \omega_3), \dots)$ is an optimal policy. ■

We now return to the DOASA class of algorithms, in which a single scenario is re-sampled in each forward pass, in contrast to the methods above when these are sampled once and then fixed.

Theorem 5 *Under FPSP and BPSP, DOASA converges with probability 1 to an optimal solution to [LP_1] in a finite number of iterations.*

Proof. By FPSP, each scenario in the finite collection of $N = \prod_{t=2}^{T-1} q_t$ scenarios will occur an infinite number of times in the course of the algorithm with probability 1. Thus with probability 1, DOASA will contain a sequence of iterations that are equivalent to MSMBD applied to the universe of scenarios. We may then apply Lemma 4 which shows that with probability 1, DOASA will converge in a finite number of steps to an optimal solution to [LP_1] in a finite number of iterations. ■

4 Discussion

The proof of almost-sure convergence above assumes the sampling procedures satisfy FPSP and BPSP. The proof of convergence in [7] makes some different assumptions, namely the Cut Sampling Property and the Sample Intersection Property. The Cut Sampling Property (CSP) states that there are only a finite number of iterations in the algorithm where Ω_t^k is empty. Since we are investigating convergence as $k \to \infty$, CSP is effectively the same as assuming that Ω_t^k is nonempty for all k.

The Sample Intersection Property (SIP) states that for any t, each $\omega_{ti} \in \Omega_t$ and each k (given $\Omega_t^k \neq \emptyset$),

$$\Pr[(\omega_{ti} \in \Omega_t^k) \cap (\omega_t^k = \omega_{ti})] > 0.$$

SIP is sufficient to guarantee FPSP and BPSP if it is accompanied by independent sampling in the forward pass and the backward pass. We state this formally.

Lemma 6 *Given independent sampling in the forward pass, SIP implies FPSP. Given independent sampling in the backward pass, SIP implies BPSP.*

Proof. By SIP, for each $\omega_{ti} \in \Omega_t$ and each k (given $\Omega_t^k \neq \emptyset$),

$$\begin{aligned} \Pr[\omega_t^k &= \omega_{ti}] > 0, & (5) \\ \Pr[\omega_{ti} &\in \Omega_t^k] > 0. & (6) \end{aligned}$$

By (5) and independent sampling in the forward pass, for any scenario $\omega(j)$ with $\omega_{ti} \in \omega(j)$, $t = 2, 3, \ldots, T-1$,

$$\Pr[\{\omega_t^k\} = \omega(j)] = \prod_{t=2}^{T-1} \Pr[\omega_t^k = \omega_{ti}] > 0.$$

Then with independent sampling in the forward pass, by the Borel-Cantelli lemma, there are infinite traversals of each scenario $\omega(j)$, $j = 1, 2, \ldots, \prod_{t=2}^{T-1} q_t$ with probability 1, and thus FPSP is satisfied.

And with (6) and independent sampling in the backward pass, by the Borel-Cantelli lemma, there are infinite visits to each ω_{ti} with probability 1, and thus BPSP is satisfied. ■

Remark 4.1: Independent sampling is necessary in Lemma 6. If independent sampling in the forward pass is not assured, then FPSP is not guaranteed. For example, suppose for $t = 2, 3, \ldots, T-1$, $\Omega_t = \{\omega_1, \omega_2\}$ and we choose ω_t^k with

$$\begin{aligned} \Pr[\omega_t^1 &= \omega_1] = \Pr[\omega_t^1 = \omega_2] = \frac{1}{2}, \\ \omega_t^k &= \omega_t^1, \quad k \geq 2. \end{aligned}$$

Then for ω_1,

$$\Pr[\omega_t^k = \omega_1] = \Pr[\omega_t^1 = \omega_1] > 0,$$

and it is easy to show that $\Pr[\omega_t^k = \omega_2] > 0$ for each k, and thus this sampling method satisfies (5). But obviously some of the scenarios will never be visited, and thus the sampling method does not satisfy FPSP.

Similarly if independent sampling in the backward pass is not assured, then BPSP is not guaranteed. For example, suppose for $t = 2, 3, \ldots, T$, $\Omega_t = \{\omega_1, \omega_2\}$, and we choose Ω_t^k with

$$\begin{aligned} \Pr[\Omega_t^1 &= \{\omega_1\}] = \Pr[\Omega_t^1 = \{\omega_2\}] = \frac{1}{2}, \\ \Omega_t^k &= \Omega_t^1, \quad k \geq 2. \end{aligned}$$

Then for example for ω_1 and $k \geq 2$,

$$\Pr[\Omega_t^k = \{\omega_1\}] = \Pr[\Omega_t^1 = \{\omega_1\}] > 0.$$

Similarly $\Pr[\Omega_t^k = \{\omega_2\}] > 0$ for each k, and thus this sampling method satisfies (6), but does not satisfy BPSP.

Remark 4.2: The CUPPS algorithm ([2]) comprises independent sampling in the forward pass, and cuts computed using $\Omega_t^k = \{\omega_t^k\}$. This is easily seen to satisfy SIP, and FPSP and BPSP, even though the backward pass is not sampled independently, but constructed from the forward pass.

Remark 4.3: SIP is not necessary for FPSP and BPSP to hold. Consider a version of CUPPS in which cuts are computed using $\Omega_t^k = \Omega_t \setminus \{\omega_t^k\}$. This does not satisfy SIP, but it does satisfy FPSP and BPSP.

Remark 4.4: The algorithms SDDP[1], AND, and ReSa all use independent sampling in the forward pass, and set $\Omega_t^k = \Omega_t$. In this case BPSP is trivially true, and FPSP follows by the Borel-Cantelli lemma. These algorithms also satisfy SIP trivially.

Remark 4.5: Lemma 8 in [7] asserts that CSP, SIP and independent sampling in the forward pass are sufficient for almost sure convergence. As discussed above there is an implicit independence assumption in the proof of Lemma 8. It is tempting to suppose that independent sampling in the forward pass and SIP give BPSP, which would make Lemma 8 true. However this is not true in general, as shown by Remark 4.1. Thus, in the absence of independent sampling in the backward pass, Lemma 8 in [7] remains unproven.

References

[1] Birge, J.R., and Louveaux, F. Introduction to Stochastic Programming, Springer Verlag, New York, 1997.

[2] Chen, Z. L., and Powell, W. B. Convergent Cutting Plane and Partial-Sampling Algorithm for Multistage Stochastic Linear Programs with Recourse, Journal of Optimization Theory and Applications, Vol. 102, pp. 497-524, 1999.

[1] We are assuming here that SDDP re-samples in its forward pass. Some commercial implementations of SDDP do not re-sample and so are more akin to MSMBD than DOASA.

[3] DONOHUE, C. J. Stochastic Network Programming and the Dynamic Vehicle Allocation Problem, PhD Dissertation, University of Michigan, Ann Arbor, Michigan, 1996.

[4] DONOHUE, C. J., and BIRGE, J.R. The Abridged nested Decomposition Method for Multistage Stochastic Linear Programs with Relatively Complete Recourse, Algorithmic Operations Research, 1, 20-30, 2006.

[5] GRIMMETT, G.R., and STIRZAKER, D.R. Probability and Random Processes, Oxford University Press, Oxford, UK, 1992.

[6] HINDSBERGER, M., and PHILPOTT, A.B. ReSa: A Method for Solving Multi-stage Stochastic Linear Programs, SPIX Stochastic Programming Symposium, Berlin, 2001.

[7] LINOWSKY, K., and PHILPOTT, A.B. On the Convergence of Sampling-Based Decomposition Algorithms for Multistage Stochastic Programs, Journal of Optimization Theory and Applications, 125, 349-366, 2005.

[8] PEREIRA, M.V.F., and PINTO, L.M.V.G. Multi-Stage Stochastic Optimization Applied to Energy Planning, Mathematical Programming, 52, pp. 359-375, 1991.

[9] PFLUG, G.CH. Scenario Tree Generation for Multiperiod Financial Optimization by Optimal Discretization, Mathematical Programming B, **89**, 251-271, 2001.

[10] SHAPIRO, A. Monte Carlo Sampling Methods, in: Ruszczyński, A. and Shapiro, A., (Eds.), Stochastic Programming, Handbook in OR & MS, Vol. 10, North-Holland Publishing Company, Amsterdam, 2003.