Operations Research Letters 41 (2013) 311–316

Contents lists available at SciVerse ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

Sharing cuts under aggregated forecasts when decomposing multi-stage stochastic programs

Anderson Rodrigo de Queiroz [a,b], David P. Morton [b,*]

[a] MC&E Research and Development, Rua: Silvestre Ferraz N. 83 Bairro: Centro, Itajuba, MG 37500-054, Brazil

[b] Graduate Program in Operations Research and Industrial Engineering, The University of Texas at Austin, Austin, TX 78712-1591, USA

ARTICLE INFO

Article history:
Received 1 November 2012
Received in revised form
27 February 2013
Accepted 4 March 2013
Available online 21 March 2013

Keywords:
Sampling-based decomposition
Multi-stage stochastic linear programs
Cutting planes

ABSTRACT

Sampling-based decomposition algorithms (SBDAs) solve multi-stage stochastic programs. SBDAs can approximately solve problem instances with many time stages when the stochastic program exhibits interstage dependence in its right-hand side parameters by appropriately sharing cuts. We extend previous methods for sharing cuts in SBDAs, establishing new results under a novel interaction between a class of interstage dependency models, and how they appear in the stochastic program.

1. Introduction

We describe a sampling-based decomposition algorithm (SBDA) for a multi-stage stochastic linear program in which the stochastic process governing the right-hand side (RHS) parameters is interstage dependent. Our SBDA has its roots in the *L*-shaped method [20] and its multi-stage counterpart [2].

Sampling provides a means for dealing with models in which the underlying stochastic process is continuous or yields a scenario tree too large for direct computation. The idea of incorporating sampling in decomposition algorithms goes back to Dantzig and Madansky [5]. Such algorithms have been developed for both two-stage models [4,8,10] and multi-stage models [14,15] with the latter algorithm known as stochastic dual dynamic programming (SDDP). SDDP was motivated by multi-stage hydro-thermal scheduling under inflow uncertainty, as is our work, and a number of variants of SDDP have been developed and analyzed [3,7,9,16–19].

SDDP algorithms approximate the expected future cost function at each stage using cuts that form an outer piecewise linear convex approximation, which is iteratively refined. SDDP maintains one set of cuts at each stage under interstage independence of the stochastic parameters. Under interstage dependence the expected future cost function depends on the history of the stochastic process up to that stage, and the cuts must be adapted to reflect that conditional state. Infanger and Morton [11] show that when dependence is restricted to the RHS vector, and the structure of the dependence satisfies an additive model then the conditional history can be captured by shifting the intercept term of the cuts using a closed-form expression.

Current long-term planning models for hydro-thermal scheduling in Brazil employ an aggregate reservoir representation (ARR) [1,6,13]. An ARR model aggregates individual hydrological reservoirs with similar characteristics and represents storage in units of energy rather than in units of volume of water. Yet stochastic models that forecast water inflows are most naturally constructed, and importantly most naturally validated, using individual reservoirs. A linear transformation maps forecasts of spatial water inflows to energy inflows in aggregate reservoirs under an ARR formulation.

Forecasting at a finer resolution and then transforming such forecasts for use in a planning-based stochastic optimization model is not unique to hydro-thermal scheduling. Forecasts for demand might be constructed, and validated, at the level of individual retail stores in a chain or at individual hotels in a chain, and yet supply chain optimization models and regional capacity planning models, may take a more aggregate view. The work of Infanger and Morton [11] cannot capture this form of dependence, and the purpose of this article is to fill this gap. First, we extend the results of [11] to capture this more general form of dependence, again achieving a closed-form expression for adjustment factors for cut intercepts. Second, we describe a general procedure by which such formulas may be derived.

* Corresponding author.
E-mail address: morton@austin.utexas.edu (D.P. Morton).

0167-6377/$ – see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.orl.2013.03.003

2. A multi-stage stochastic linear program

We consider a T-stage stochastic linear program with recourse (SLP-T) of the following form:

$$\begin{aligned} \min_{x_1} \quad & c_1 x_1 + \mathbb{E}_{b_2|b_1} h_2(x_1, b_2) \\ \text{s.t.} \quad & A_1 x_1 = B_1 x_0 + \rho_1 b_1 + k_1 \\ & x_1 \geq 0, \end{aligned} \tag{1}$$

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

$$\begin{aligned} h_t(x_{t-1}, b_t) = \min_{x_t} \quad & c_t x_t + \mathbb{E}_{b_{t+1}|b_1,\ldots,b_t} h_{t+1}(x_t, b_{t+1}) \\ \text{s.t.} \quad & A_t x_t = B_t x_{t-1} + \rho_t b_t + k_t \\ & x_t \geq 0, \end{aligned} \tag{2}$$

and where $h_{T+1} \equiv 0$.

Matrix $A_t \in \mathbb{R}^{m_t \times d_t}$, the random vector b_t has dimension q_t, and deterministic matrix $\rho_t \in \mathbb{R}^{m_t \times q_t}$, $t = 1, \ldots, T$. The remaining matrices and vectors are dimensioned to conform. We denote stage t's sample space by Ω_t, and $\omega_t \in \Omega_t$ is a sample point (scenario). A stage $t > 1$ scenario, ω_t, has a unique stage $t-1$ ancestor denoted $a(\omega_t)$, and a stage $t < T$ scenario has a set of stage $t+1$ descendants denoted $\Delta(\omega_t)$. We denote a realization of b_t by $b_t^{\omega_t}$ and when the scenario tree is finite, we denote b_t's probability mass function by $\mathbb{P}(b_t = b_t^{\omega_t}) = p_t^{\omega_t}$ and conditional probability mass function by $\mathbb{P}(b_{t+1} = b_{t+1}^{\omega_{t+1}} \mid b_t = b_t^{\omega_t}) = p_{t+1}^{\omega_{t+1}|\omega_t}$, where $\omega_{t+1} \in \Delta(\omega_t)$.

We assume that A_t, B_t, c_t, ρ_t, and k_t, $t = 2, \ldots, T$, are deterministic, and that $\{b_t\}_{t=1}^T$ comprises the only randomness in model (1). That said, all that we describe also holds when (A_t, B_t, c_t, k_t), $t = 2, \ldots, T$, are random but interstage independent. Parameters $(A_1, B_1, c_1, k_1), x_0, \rho_1, \ldots, \rho_T$, and b_1 are deterministic, but for simplicity we sometimes treat b_1 in our notation as if it were random. We assume that model (1) has relatively complete recourse; i.e., for any realizations of $b_1, \ldots, b_{t-1}$, and for any history of feasible decisions, the stage t model (2) is feasible with probability one.

If $\{b_t\}_{t=1}^T$ is interstage independent, $\mathbb{E}_{b_{t+1}|b_1,\ldots,b_t} h_{t+1}(x_t, b_{t+1})$ simplifies to $\mathbb{E}_{b_{t+1}} h_{t+1}(x_t, b_{t+1})$. This, in turn, simplifies application of a decomposition algorithm. However, in many settings, including hydro-thermal scheduling with stages representing months and b_t representing stochastic inflows, interstage independence does not provide a good probabilistic model.

An important and widely used dependency model has form:

$$b_{t+1} = R_t b_t + \eta_{t+1}, \quad t = 1, \ldots, T-1, \tag{3a}$$

$$\eta_t, \quad t = 2, \ldots, T, \text{ are independent.} \tag{3b}$$

Model (3) generalizes an autoregressive lag-one model to allow for time dependence in $R_t \in \mathbb{R}^{q_{t+1} \times q_t}$, $t = 1, \ldots, T-1$, which we assume are known. This can capture seasonality and time-dependent trends. Lag-one model (3a) generalizes to allow for dependence on multiple time stages. The cut-sharing ideas we describe also extend in a straightforward, if notationally cumbersome, way to handle higher-order lag dependence.

The setup we outline above deviates from a standard multi-stage stochastic linear program, which simply has RHS "$B_t x_{t-1} + b_t$" in model (2). Our motivation comes from the desire to develop, and validate, probabilistic models at a finer grain resolution but employ them in a more aggregate planning model. As indicated above, the random vectors b_t have dimension q_t and deterministic matrix $\rho_t \in \mathbb{R}^{m_t \times q_t}$. In a hydro-thermal scheduling model in the ARR setting, q_t represents the number of individual hydro generation plants, which in Brazil is about 150, while $m_t = 4$ is the number of aggregate reservoirs [12,13]. The matrices ρ_t transform water inflows at hydro plants into energy inflows in the aggregate reservoirs. We separate $\rho_t b_t$ and k_t to distinguish stochastic parameters from constant terms. In the hydro-thermal setting $\rho_t b_t$ denotes random energy inflows while k_t includes demand, which is not modeled as stochastic.

Our stochastic program (1)/(2) offers no new modeling flexibility relative to the standard model with RHS $B_t x_{t-1} + b_t$ until we consider its interaction with stochastic model (3). Suppose $\{b_t\}_{t=1}^T$ satisfies (3), and consider process $\{\rho_t b_t\}_{t=1}^T$. Random vector $\rho_{t+1} b_{t+1}$ *cannot* be expressed in the form of (3) with predictors $\rho_t b_t$ and independent increment $\rho_t \eta_t$. The aggregated predictor $\rho_t b_t$ is insufficient to forecast $\rho_{t+1} b_{t+1}$. Instead we require the unaggregated predictor b_t, and this is the central reason why the results of Infanger & Morton [11] do not apply.

There is a compelling argument for forecasting water inflows at individual hydro plants rather than energy inflows into aggregate reservoirs. The former allows us to exploit local predictors like precipitation and soil type in hydrological run-off models. Forecasting water inflows better lends itself to validation because individual reservoir inflows are measurable. Moreover, when the configuration of the hydro power system changes over time this complicates an energy-inflow forecast because a different forecasting model is required for each configuration. Forecasting energy inflows unnecessarily ties a forecasting model of a *natural process* to the *decision process* associated with the hydro-thermal system.

SDDP requires a finite scenario tree of a particular form as input, and above we allow η_t and b_t to be continuous. Below we assume $\{b_t\}_{t=1}^T$ are interstage dependent according to model (3). Thus the expectation in (2) simplifies to $\mathbb{E}_{b_{t+1}|b_t} h_{t+1}(x_t, b_{t+1})$, and we form a sample scenario tree under this interstage dependency model using the following algorithm.

SSTIDM algorithm:

Input: Multivariate distributions F_t governing η_t, $t = 2, \ldots, T$, assumed to be interstage independent. A procedure for drawing i.i.d. observations from each F_t. Dependency model (3) with known R_t matrices. Branch size $n(t)$ for each stage, $t = 2, \ldots, T$; e.g., $n(t) = 20$, $\forall t$.

Output: A finite sample scenario tree with the property that its random vectors satisfy model (3), where the independent increments η_t have an empirical distribution.

1. Let b_1 denote the known first stage realization.

2. Sample $\eta_2^1, \ldots, \eta_2^{n(2)}$ i.i.d. from F_2. Use Eq. (3a) with $t = 1$ to form $b_2^1, \ldots, b_2^{n(2)}$; i.e., $b_2^i = R_1 b_1 + \eta_2^i$, $i = 1, \ldots, n(2)$, form the descendant nodes of b_1.

3. Sample $\eta_3^1, \ldots, \eta_3^{n(3)}$ i.i.d. from F_3, independent of those in stage 2. For each of the $i = 1, \ldots, n_2 \equiv n(2)$ stage 2 nodes use Eq. (3a) with $t = 2$ to form the descendant nodes $b_3^{i,1}, \ldots, b_3^{i,n(3)}$ using the *same* set of increments $\{\eta_3^1, \ldots, \eta_3^{n(3)}\}$.

$\vdots$

t. Sample $\eta_t^1, \ldots, \eta_t^{n(t)}$ i.i.d. from F_t, independent of those in stages $2, \ldots, t-1$. For each of the $i = 1, \ldots, n_{t-1} \equiv \prod_{t=2}^{t-1} n(t)$ stage $t-1$ nodes use Eq. (3a) to form the descendant nodes $b_t^{i,1}, \ldots, b_t^{i,n(t)}$ using the same set of increments $\{\eta_t^1, \ldots, \eta_t^{n(t)}\}$.

$\vdots$

T. Sample $\eta_T^1, \ldots, \eta_T^{n(T)}$ i.i.d. from F_T, independent of those in stages $2, \ldots, T-1$. For each of the $i = 1, \ldots, n_{T-1} \equiv \prod_{t=2}^{T-1} n(t)$ stage $T-1$ nodes use Eq. (3a) with $t = T-1$ to form the descendant nodes $b_T^{i,1}, \ldots, b_T^{i,n(T)}$ using the same set of increments $\{\eta_T^1, \ldots, \eta_T^{n(T)}\}$. □

In the SSTIDM Algorithm we sample $\eta_t^1, \ldots, \eta_t^{n(t)}$ in an i.i.d. manner for simplicity and concreteness. However, other schemes that sample independently at each stage are also viable. For

example, Latin hypercube sampling or randomized quasi-Monte Carlo sampling can also be applied, and the cut-sharing formulas we describe remain valid.

Henceforth, we assume that the output of the SSTIDM Algorithm defines the stochastic process which specifies model (1). Thus, stage t's sample space satisfies $|\Omega_t| = n_t$, the set of stage t descendants satisfies $|\Delta(\omega_t)| = n(t+1)$, $p_t^{\omega_t} = 1/n_t$, and $p_{t+1}^{\omega_{t+1}|\omega_t} = 1/n(t+1)$. Given the form of the dependency model (3), we introduce one additional construct. The stage t sample space may be expressed as $\Omega_t = \Sigma_2 \times \Sigma_3 \times \cdots \times \Sigma_t$, for $t = 2, \ldots, T$, where Σ_t is η_t's sample space and $|\Sigma_t| = n(t)$.

3. Sampling-based decomposition algorithm

This section states, in general form, the SDDP algorithm of Pereira and Pinto [15] applied to model (1). Section 4 then derives details of the cut computations when the stochastic parameters satisfy model (3). While we formalize the SDDP algorithm below, it is easy to visualize the algorithm's forward and backward passes, as shown in Fig. 1, although the practical scale of problems SDDP can address is much larger. For example, Maceira et al. [12] consider a sample scenario tree with $n(t) = 20$ scenarios per stage and $T = 120$ stages.

SDDP uses a master program at each stage, which accumulates cuts approximating the expected future cost function. Let $\vec{G}_t \in \mathbb{R}^{\ell_t \times d_t}$ and $\vec{g}_t \in \mathbb{R}^{\ell_t}$ denote a cut-gradient matrix and a cut-intercept vector. Each backward pass of SDDP along a sample path (there are three in Fig. 1) augments $\vec{G}_t$ and $\vec{g}_t$ with one additional row (incrementing ℓ_t), in a manner we make precise below. We define a stage t model that serves as a master program with respect to its stage $t+1$ descendants and as a subproblem with respect to its stage $t-1$ ancestor:

$$\min_{x_t, \theta_t} \quad c_t x_t + \theta_t \tag{4a}$$

$$\text{s.t.} \quad A_t x_t = B_t x_{t-1} + \rho_t b_t + k_t \quad : \pi_t \tag{4b}$$

$$-\vec{G}_t x_t + e\,\theta_t \geq \vec{g}_t \quad : \alpha_t \tag{4c}$$

$$x_t \geq 0. \tag{4d}$$

Here, θ_t in the objective function (4a), coupled with cut constraints (4c), forms an outer linearization of $\mathbb{E}_{b_{t+1}|b_t} h_{t+1}(x_t, b_{t+1})$. Constraint (4c)'s e is the ℓ_t-vector of all 1s. The structural and non-negativity constraints (4b) and (4d) repeat from model (2). Model (4) holds for $t = 1, \ldots, T$, except that for $t = T$ the cut constructs are absent. The π_t and α_t represent dual (row) vectors associated with constraints (4b) and (4c), respectively. It is model (4) that is solved at each node in the forward and backward paths in Fig. 1. When we let $b_t = b_t^{\omega_t}$ and $x_{t-1} = x_{t-1}^{a(\omega_t)}$, we refer to model (4) as sub(ω_t). When we specialize model (4) in this way, we similarly append ω_t superscripts to the primal and dual solutions, $(x_t^{\omega_t}, \theta_t^{\omega_t})$ and $(\pi_t^{\omega_t}, \alpha_t^{\omega_t})$. When $\{b_t\}_{t=1}^T$ is interstage independent then one set of cuts $(\vec{G}_t, \vec{g}_t)$ can be shared for sub(ω_t) for all $\omega_t \in \Omega_t$. However, under the interstage dependency model (3) cuts vary by scenario. As we see in Section 4, this dependence only appears in the cut intercepts and for this reason we label the cuts $(\vec{G}_t, \vec{g}_t^{\,\omega_t})$ when referring to sub(ω_t). We formalize the SDDP algorithm below.

SDDP algorithm:

Input: Instance of model (1) with scenario tree from the SSTIDM Algorithm.

Output: Cut information that yields policy for model (1), first stage solution, x_1, and bounds $\underline{z}$ and $\bar{z}$.

0. let $k = 0$; append lower bounding cuts: $\theta_t \geq 0, t = 1, \ldots, T-1$;
1. solve the stage 1 master program, i.e., (4) with $t = 1$, and obtain (x_1^k, θ_1^k); let $\underline{z}^k = c_1 x_1^k + \theta_1^k$;
2. sample i.i.d. paths from Ω_T and index them by $\mathcal{S}^k$;
   do $\omega \in \mathcal{S}^k$
    do $t = 2$ to T
     form RHS of sub(ω_t): $B_t[x_{t-1}^{a(\omega_t)}]^k + \rho_t b_t^{\omega_t} + k_t$;
     form cut intercept vector for sub(ω_t): $\vec{g}_t^{\,\omega_t}$;
     solve and obtain $[x_t^{\omega_t}]^k$;
    enddo
   enddo
   let $\bar{z}^k = c_1 x_1^k + \frac{1}{|\mathcal{S}^k|} \sum_{\omega \in \mathcal{S}^k} \sum_{t=2}^T c_t [x_t^{\omega_t}]^k$;
3. if stopping criterion, given $\bar{z}^k$ and $\underline{z}^k$, is satisfied then stop and output: (i) cut information for each stage, (ii) first stage solution $x_1 = x_1^k$, and (iii) bounds $\underline{z} = \underline{z}^k$ and $\bar{z} = \bar{z}^k$;
4. do $t = T-1$ downto 1
    do $\omega \in \mathcal{S}^k$
     do $\omega_{t+1} \in \Delta(\omega_t)$
      form RHS of sub(ω_{t+1}): $B_t[x_{t-1}^{a(\omega_t)}]^k + \rho_t b_t^{\omega_t} + k_t$;
      form cut intercept vector for sub(ω_t): $\vec{g}_t^{\,\omega_t}$;
      solve to obtain dual vectors $\pi_{t+1}^{\omega_{t+1}}, \alpha_{t+1}^{\omega_{t+1}}$;
     enddo
     Form cut gradient G_t and information required to compute scenario dependent cut intercept $g_t^{\omega_t}$;
    enddo
   enddo
5. let $k = k+1$; goto step 1; □

Our statement of the SDDP algorithm assumes, in step 0, that $\mathbb{E}_{b_{t+1}|b_t} h_{t+1}(x_t, b_{t+1}) \geq 0$. When this is not the case, the initial cuts $\theta_t \geq 0$ should replace "0" with a valid lower bound. Steps 2 and 4 of the SDDP algorithm carry out the forward and backward passes depicted in Fig. 1. In step 2 at the k-th iteration, we select a set of random sample paths from the root node to the stage T leaf nodes denoted $\mathcal{S}^k$, and an element $\omega \in \mathcal{S}^k$ has the form $\omega = (\sigma_2, \ldots, \sigma_T)$ with $\omega_t = (\sigma_2, \ldots, \sigma_t)$. In step 4 we construct the cut gradient G_t and the scenario dependent cut intercept in a manner that Section 4 details.

As Section 1 indicates, multiple authors have proposed improvements to the basic SDDP algorithm we outline. We do not review those improvements here, except to say that the cut-sharing procedures we provide can also be used in those refined algorithms. Our statement of the SDDP algorithm is purposively vague on how to compute and store cuts. Specificity in this regard requires the development in the next section.

4. Sharing cuts under the dependency model

This section derives a closed-form expression for cut intercepts, which allows a cut computed for one stage t subproblem, sub(ω_t), to be adjusted so that the cut is valid for any other stage t subproblem, sub(ω_t'). The dual of model (4) is:

$$\max_{\pi_t, \alpha_t} \quad \pi_t\,[B_t x_{t-1} + \rho_t\,(R_{t-1} b_{t-1} + \eta_t) + k_t] + \alpha_t \vec{g}_t \tag{5a}$$

$$\text{s.t.} \quad \pi_t A_t - \alpha_t \vec{G}_t \leq c_t \quad : x_t \tag{5b}$$

$$e^\top \alpha_t = 1 \quad : \theta_t \tag{5c}$$

$$\alpha_t \geq 0, \tag{5d}$$

where Eq. (3a) substitutes for b_t in (5a). When $t = T$ constraints (5c) and the other terms with α_t are absent.

The dual variables for the descendants of sub(ω_t), denoted $(\pi_{t+1}^{\omega_{t+1}}, \alpha_{t+1}^{\omega_{t+1}})_{\omega_{t+1} \in \Delta(\omega_t)}$, are feasible to the dual of the descendants of any other stage t subproblem, say, sub(ω_t'); see [11, Proposition 1], and hence we can index them by $(\pi_{t+1}^{\sigma_{t+1}}, \alpha_{t+1}^{\sigma_{t+1}})_{\sigma_{t+1} \in \Sigma_{t+1}}$. From model (5), with t incremented to $t+1$, we therefore compute the cut gradient as:

$$G_t = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\,\omega_t} \pi_{t+1}^{\omega_t} B_{t+1} = \bar{\pi}_{t+1} B_{t+1}, \tag{6}$$

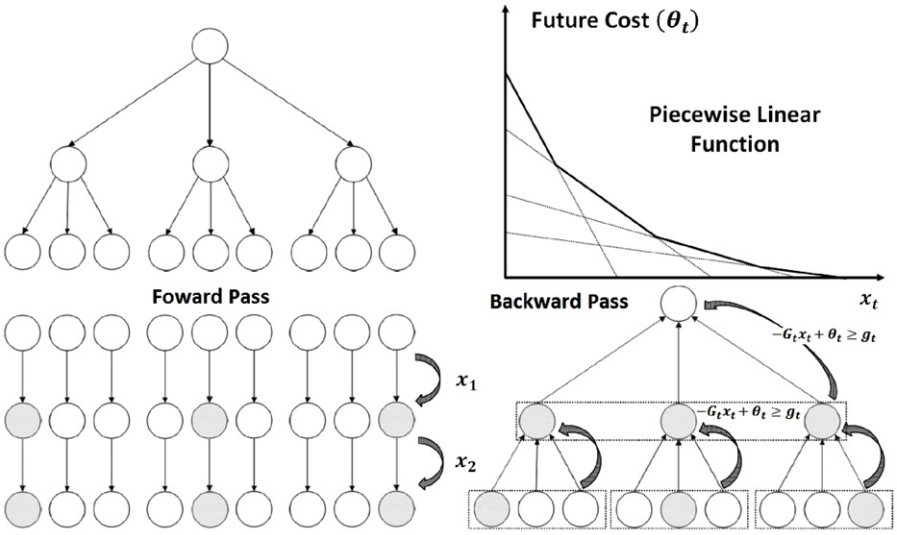

Fig. 1. Overview of SDDP scheme.

and this forms one row of $\vec{G}_t$. Note that $p_{t+1}^{\omega_{t+1}|\,\omega_t} = p_{t+1}^{\sigma_{t+1}}$ does not depend on ω_t. Thus the cut gradient for sub(ω_t) is identical to the cut gradient for sub(ω_t') when that cut is shared. In Eq. (6), we have $\bar{\pi}_{t+1} = \sum_{\sigma_{t+1}\in\Sigma_{t+1}} p_{t+1}^{\sigma_{t+1}} \pi_{t+1}^{\sigma_{t+1}}$, and in what follows we use $\mathbb{E}_{\sigma_t}[\cdot]$ to denote $\sum_{\sigma_t\in\Sigma_t} p_t^{\sigma_t}[\cdot]$.

The cut intercept calculation is more subtle, and it is instructive to start with the simplest nontrivial case of SLP-3 for $t = 2$. The cut's gradient is given by (6) with $t = 2$, and the cut's intercept is given by:

$$\begin{aligned} g_2^{\omega_2} &= \sum_{\omega_3\in\Delta(\omega_2)} p_3^{\omega_3|\,\omega_2} \pi_3^{\omega_3} \left(\rho_3 b_3^{\omega_3} + k_3\right) \\ &= \mathbb{E}_{\sigma_3} \pi_3^{\sigma_3} \left(\rho_3 \left[R_2 b_2^{\omega_2} + \eta_3^{\sigma_3}\right] + k_3\right) \\ &= g_2^{ind} + g_2^{dep}(\omega_2), \end{aligned}$$

where

$$g_2^{ind} = \mathbb{E}_{\sigma_3} \pi_3^{\sigma_3} \rho_3 \eta_3^{\sigma_3} + \bar{\pi}_3 k_3 \tag{7a}$$

$$g_2^{dep}(\omega_2) = \bar{\pi}_3 \rho_3 R_2 b_2^{\omega_2}. \tag{7b}$$

This means that to run SDDP for SLP-3, for each stage $t = 2$ cut we must store: (i) the cut gradient, G_2, (ii) the independent component of the cuts intercept, g_2^{ind}, as given by (7a), and (iii) the m_3-dimensional row vector, $\bar{\pi}_3 \equiv \mathbb{E}_{\sigma_3}\pi_3^{\sigma_3}$. Then when we must "form cut intercept vector for sub(ω_2)" in steps 2 and 4 in the SDDP algorithm, we do so for each stage $t = 2$ cut by taking the inner product of $\bar{\pi}_3$ from (iii) with $\rho_3 R_2 b_2^{\omega_2}$ to form $g_2^{dep}(\omega_2)$ and then adding that to g_2^{ind} from (ii) to form $g_2^{\omega_2}$. This is done for each component of $\vec{g}_2^{\,\omega_2}$.

The SLP-3 case is simple because forming $g_2^{dep}(\omega_2)$ does not involve dual variables associated with the cuts in the descendant nodes because there are no cuts in the third stage. For SLP-T the general form of a cut intercept calculation for sub(ω_{t-1}) is:

$$\begin{aligned} g_{t-1}^{\omega_{t-1}} &= \sum_{\sigma_t\in\Sigma_t} p_t^{\sigma_t} \pi_t^{\sigma_t} \left(\rho_t \left[R_{t-1} b_{t-1}^{\omega_{t-1}} + \eta_t^{\sigma_t}\right] + k_t\right) \\ &\quad + \sum_{\sigma_t\in\Sigma_t} p_t^{\sigma_t} \alpha_t^{\sigma_t} \vec{g}_t^{\,(\omega_{t-1},\sigma_t)}. \end{aligned} \tag{8}$$

The computation for stage $T-1$ parallels that for the second stage in SLP-3, but the effect of the $\alpha_{t+1}^{\sigma_{t+1}} \vec{g}_{t+1}^{\,(\omega_t,\sigma_{t+1})}$ term in (8) recurses to earlier stages in a way we characterize in Theorem 1.

Theorem 1. *Consider model (1) under dependency model (3). Let $(\pi_{t+1}^{\sigma_{t+1}}, \alpha_{t+1}^{\sigma_{t+1}})_{\sigma_{t+1}\in\Sigma_{t+1}}$ denote the dual variables used to compute a stage t cut as in Eq. (8), $t = 1,\ldots,T-1$, with α_T absent under $t = T-1$. Let ℓ_t be the number of stage t cuts, and define $\mathcal{P}_{t+1}$ to be the $\ell_t \times m_{t+1}$ matrix whose rows contain $\bar{\pi}_{t+1} \equiv \mathbb{E}_{\sigma_{t+1}} \pi_{t+1}^{\sigma_{t+1}}$, and similarly define $\mathcal{A}_{t+1}$ to be the $\ell_t \times \ell_{t+1}$ matrix whose rows contain $\bar{\alpha}_t \equiv \mathbb{E}_{\sigma_{t+1}} \alpha_{t+1}^{\sigma_{t+1}}$.*

Then the stage $t = 2,\ldots,T-1$ cut intercepts are given by:

$$g_t^{\omega_t} = g_t^{ind} + g_t^{dep}(\omega_t) \tag{9a}$$

$$\begin{aligned} g_t^{ind} &= \mathbb{E}_{\sigma_{t+1}} \pi_{t+1}^{\sigma_{t+1}} \rho_{t+1} \eta_{t+1}^{\sigma_{t+1}} + \overline{\pi}_{t+1} k_{t+1} + \overline{\alpha}_{t+1} \vec{g}_{t+1}^{\,ind} \\ &\quad + \mathbb{E}_{\sigma_{t+1}} \alpha_{t+1}^{\sigma_{t+1}} D_{t+1} \eta_{t+1}^{\sigma_{t+1}}, \end{aligned} \tag{9b}$$

$$g_t^{dep}(\omega_t) = [\overline{\pi}_{t+1}\rho_{t+1} + \overline{\alpha}_{t+1} D_{t+1}] R_t b_t^{\omega_t}, \tag{9c}$$

where for $t = 2,\ldots,T$ the matrix D_t is defined recursively as:

$$D_t = [\mathcal{P}_{t+1}\rho_{t+1} + \mathcal{A}_{t+1} D_{t+1}] R_t, \qquad D_T = 0. \tag{10}$$

Proof. We proceed by induction with base case $t = T-1$. The cut intercept $g_{T-1}^{\omega_{T-1}}$ can be computed using the method above for SLP-3, leading to Eq. (7) and in this case we have that (9) holds because $D_T = 0$ and $\bar{\alpha}_T = 0$.

The inductive hypothesis is (9). We verify the same expressions with t decremented by 1. The vector analog of (9c) is:

$$\vec{g}_t^{\,dep}(\omega_t) = [\mathcal{P}_{t+1}\rho_{t+1} + \mathcal{A}_{t+1} D_{t+1}] R_t b_t^{\omega_t} = D_t b_t^{\omega_t}.$$

A stage $t-1$ cut intercept is given by Eq. (8). The first term on the RHS of (8) is:

$$\overline{\pi}_t \rho_t R_{t-1} b_{t-1}^{\omega_{t-1}} + \mathbb{E}_{\sigma_t} \pi_t^{\sigma_t} \rho_t \eta_t^{\sigma_t} + \bar{\pi}_t k_t. \tag{11}$$

Applying the inductive hypothesis (9a) and (9c) and model (3) to the second term of (8), the second term becomes:

$$\begin{aligned} &\overline{\alpha}_t \vec{g}_t^{\,ind} + \overline{\alpha}_t [\mathcal{P}_{t+1}\rho_{t+1} + \mathcal{A}_{t+1} D_{t+1}] R_t R_{t-1} b_{t-1}^{\omega_{t-1}} \\ &\quad + \mathbb{E}_{\sigma_t} \alpha_t^{\sigma_t} [\mathcal{P}_{t+1}\rho_{t+1} + \mathcal{A}_{t+1} D_{t+1}] R_t \eta_t^{\sigma_t}. \end{aligned} \tag{12}$$

Using the definition of D_t from (10) and summing (11) and (12) we obtain:

$$\begin{aligned} g_{t-1}^{\omega_{t-1}} &= \overline{\pi}_t \rho_t R_{t-1} b_{t-1}^{\omega_{t-1}} + \mathbb{E}_{\sigma_t} \pi_t^{\sigma_t} \rho_t \eta_t^{\sigma_t} + \bar{\pi}_t k_t \\ &\quad + \overline{\alpha}_t \vec{g}_t^{\,ind} + \overline{\alpha}_t D_t R_{t-1} b_{t-1}^{\omega_{t-1}} + \mathbb{E}_{\sigma_t} \alpha_t^{\sigma_t} D_t \eta_t^{\sigma_t}. \end{aligned} \tag{13}$$

We partition expression (13) into its scenario dependent and independent parts to obtain (9), with t decremented to $t-1$. □

When running SDDP for model (1) under dependency model (3), it is necessary to store for each computed cut: (i) the cut gradient G_t, calculated using Eq. (6), (ii) the scenario independent cut intercept g_t^{ind} from Eq. (9b), and (iii) the cumulative expected dual vector $[\overline{\pi}_{t+1}\rho_{t+1} + \overline{\alpha}_{t+1}D_{t+1}]R_t$; i.e., one row of D_t from Eq. (10). At a particular stage t, we can form valid cuts for sub(ω_t) with the stored information by calculating the dependent part of the cut intercept using the closed-form scenario-dependent correction term (9c) and then the cut intercept using Eq. (9a). The first time we compute a cut, the cumulative expected dual vector associated with this cut can be created from the set of cumulative expected dual vectors from its descendant scenarios. In order to perform such computation we use (9c) and (10). Relative to the case of interstage independence we require additional storage of the q_t-vector $[\overline{\pi}_{t+1}\rho_{t+1} + \overline{\alpha}_{t+1}D_{t+1}]R_t$ for each cut.

5. Expanding the state

Another way to handle dependency model (3) is to reformulate model (1) so it exhibits interstage independence. This can be done by increasing the dimension of the decision vectors to capture the history of the stochastic process. Let y_t be an auxiliary decision vector with $y_t = R_{t-1}b_{t-1} + \eta_t$. Then we have:

$$\begin{aligned} \min_{x_1, y_1} \quad & c_1x_1 + \mathbb{E}_{\eta_2} h_2(x_1, y_1, \eta_2) \\ \text{s.t.} \quad & A_1x_1 - \rho_1 y_1 = B_1x_0 + k_1 \\ & y_1 = b_1 \\ & x_1 \geq 0, \end{aligned} \tag{14}$$

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

$$\begin{aligned} h_t(x_{t-1}, y_{t-1}, \eta_t) = \min_{x_t, y_t} \quad & c_tx_t + \mathbb{E}_{\eta_{t+1}} h_{t+1}(x_t, y_t, \eta_{t+1}) \\ \text{s.t.} \quad & A_tx_t - \rho_t y_t = B_tx_{t-1} + k_t \\ & y_t = R_{t-1}y_{t-1} + \eta_t \\ & x_t \geq 0, \end{aligned}$$

and where $h_{T+1} \equiv 0$.

Applying the decomposition algorithm to model (14) we arrive at the master program:

$$\min_{x_t, y_t, \theta_t} \quad c_tx_t + \theta_t \tag{15a}$$

$$\text{s.t.} \quad A_tx_t - \rho_t y_t = B_tx_{t-1} + k_t \quad : \pi_t \tag{15b}$$

$$-\vec{G}_t^x x_t - \vec{G}_t^y y_t + e\,\theta_t \geq \vec{g}_t \quad : \alpha_t \tag{15c}$$

$$y_t = R_{t-1}y_{t-1} + \eta_t \quad : \gamma_t \tag{15d}$$

$$x_t \geq 0, \tag{15e}$$

where $\vec{G}_t^x$ and $\vec{G}_t^y$ are the cut-gradient matrices related to x_t and y_t, respectively.

The dual of (15) is:

$$\max_{\pi_t, \alpha_t, \gamma_t} \quad \pi_t\,(B_tx_{t-1} + k_t) + \alpha_t\vec{g}_t + \gamma_t\,(R_{t-1}y_{t-1} + \eta_t) \tag{16a}$$

$$\text{s.t.} \quad \pi_tA_t - \alpha_t\vec{G}_t^x \leq c_t \quad : x_t \tag{16b}$$

$$-\pi_t\rho_t - \alpha_t\vec{G}_t^y + \gamma_t = 0 \quad : y_t \tag{16c}$$

$$e^\top\alpha_t = 1 \quad : \theta_t \tag{16d}$$

$$\alpha_t \geq 0. \tag{16e}$$

Thus the cut gradients for x_t and y_t are:

$$G_t^x = \mathbb{E}_{\sigma_{t+1}} \pi_{t+1}^{\sigma_{t+1}} B_{t+1}, \tag{17a}$$

$$G_t^y = \mathbb{E}_{\sigma_{t+1}} \gamma_{t+1}^{\sigma_{t+1}} R_t. \tag{17b}$$

The cut intercept for the new formulation is computed using:

$$g_t = \mathbb{E}_{\sigma_{t+1}} \gamma_{t+1}^{\sigma_{t+1}} \eta_{t+1}^{\sigma_{t+1}} + \bar{\pi}_{t+1}k_{t+1} + \bar{\alpha}_{t+1}\vec{g}_{t+1}. \tag{18}$$

With these constructs we can apply SDDP without requiring the cut correction terms of Section 4. Rather, for each cut we store: (i) the cut gradients, G_t^x and G_t^y using Eqs. (17) and (ii) the cut intercept term, g_t, using Eq. (18). (Because of interstage independence, we need not index these cut coefficients by ω_t.) That said, we do not see this as a good idea, recalling that for the Brazilian hydro-thermal system the number of aggregate reservoirs is 4 and the number of hydro plants is about 150. So we have dramatically increased the size of the linear programs by introducing decision variables y_t, and the associated constraints.

The above development has value because it provides a simple, and more generally applicable, means by which we can derive the scenario-dependent cut correction terms of Theorem 1. This is made precise in Theorem 2.

Theorem 2. *Applying the SDDP algorithm to model (1) under the scenario-dependent cut intercept correction terms of Theorem 1 is equivalent to applying the SDDP algorithm to model (14) under interstage independence. The mapping between the two is as follows: Let (π_t, α_t) solve model (5), let $\vec{G}_t$ be the matrix version of Eq. (6) in primal–dual pair (4)–(5), and let $\vec{g}_t^{\,ind}$ and $\vec{g}_t^{\,dep}(\omega_t)$ be defined via (9b)–(9c). Then $\vec{G}_t^x$, $\vec{G}_t^y$, and $\vec{g}_t$ from (17) and (18) in primal–dual pair (15)–(16) satisfy*

$$\vec{G}_t^x = \vec{G}_t \tag{19a}$$

$$\vec{G}_t^y = D_t \tag{19b}$$

$$\vec{g}_t = \vec{g}_t^{\,ind} \tag{19c}$$

$$\vec{G}_t^y b_t^{\omega_t} = \vec{g}_t^{\,dep}(\omega_t), \tag{19d}$$

and

$$(\pi_t, \alpha_t, \pi_t\rho_t + \alpha_tD_t) \tag{20}$$

solves model (16).

Proof. Consider primal–dual pairs (4)–(5) and (15)–(16) for $t = T$ with the usual caveat that the cut constructs are absent. If x_T and π_T solve (4)–(5) then it is straightforward to show, using linear programming optimality conditions, that (x_T, b_T) and $(\pi_T, \pi_T\rho_T)$ solve (15)–(16). This establishes (20) for $t = T$, and hence (19) for $t = T - 1$. This provides the base case of an inductive argument.

Now consider primal–dual pairs (4)–(5) and (15)–(16) for t and assume (19a)–(19c) hold in those models. If (x_t, θ_t) and (π_t, α_t) solve (4)–(5) then it is again straightforward to show that (x_t, b_t, θ_t) and $(\pi_t, \alpha_t, \pi_t\rho_t + \alpha_tD_t)$ solve (15)–(16). With this mapping of the dual variables we compute cuts for stage $t - 1$. Using Eq. (17a), we have that Eq. (17b) follows, where both have t decremented by 1. Similarly, using Eq. (17b) we have $G_{t-1}^y = \mathbb{E}_{\sigma_t}\left[\pi_t^{\sigma_t}\rho_t + \alpha_t^{\sigma_t}D_t\right]R_{t-1}$, which in vector form yields $\vec{G}_{t-1}^y = [\mathcal{P}_t\rho_t + \mathcal{A}_tD_t]R_{t-1} = D_{t-1}$. Eq. (19c) holds upon substituting $\pi_t^{\sigma_t}\rho_t + \alpha_t^{\sigma_t}D_t$ into (18). Given (19b), (9c) in vector form, and Eq. (10), we have that Eq. (19d) is immediate. □

The value of formulation (14) is two-fold. First, this reformulation clarifies what interstage dependency models are tractable using cut-sharing formulas. Second, Theorem 2 provides a means to derive the type of cut-sharing formulas we obtain in Theorem 1 under different interstage dependency models. In particular, we see that Eq. (19d) provides the correction term for the cut intercept. The cut-gradient matrix satisfies Eq. (19b). So the appropriate D_t-matrix for a different interstage dependency model can be found by unfolding the recursion specified by $\gamma_t = \pi_t\rho_t + \alpha_tD_t$ and Eq. (19b), which we can see follows from constraint (16c) in model (16).

Acknowledgments

We thank Erlon Finardi and Vitor de Matos for valuable discussions regarding the ideas of Section 5, and we thank Area Editor Rüdiger Schultz and an anonymous referee for their comments on our initial manuscript. This paper is part of the first author's dissertation research, which was funded by the Capes/Fulbright Program through process 2800-06-3/2007 and grant 15073707. The second author's research was partially supported by the Defense Threat Reduction Agency through grant HDTRA1-08-1-0029.

References

[1] N.V. Arvanitidis, J. Rosing, Composite representation of a multireservoir hydroelectric power system, IEEE Transactions on Power Apparatus and Systems 89 (1970) 319–329.
[2] J. Birge, Decomposition and partitioning methods for multistage stochastic linear programs, Operations Research 33 (1985) 989–1007.
[3] Z. Chen, W. Powell, Convergent cutting-plane and partial-sampling algorithm for multistage stochastic linear programs with recourse, Journal of Optimization Theory and Applications 102 (1999) 497–524.
[4] G.B. Dantzig, P.W. Glynn, Parallel processors for planning under uncertainty, Annals of Operations Research 22 (1990) 1–21.
[5] G.B. Dantzig, A. Madansky, On the solution of two-stage linear programs under uncertainty, in: J. Neyman (Ed.), Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 1961, pp. 165–176.
[6] V.L. de Matos, E.C. Finardi, E.L. da Silva, Comparison between the energy equivalent reservoir per subsystem and per cascade in the long-term operation planning in Brazil, in: EngOpt 2008—International Conference on Engineering Optimization, Rio de Janeiro, Brazil, 2008.
[7] C.J. Donohue, J.R. Birge, The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse, Algorithmic Operations Research 1 (2006) 20–30.
[8] J.L. Higle, S. Sen, Stochastic decomposition: an algorithm for two-stage linear programs with recourse, Mathematics of Operations Research 16 (1991) 650–669.
[9] T. Homem-de-Mello, V. de Matos, E. Finardi, Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling, Energy Systems 2 (2011) 1–31.
[10] G. Infanger, Monte Carlo (importance) sampling within a Benders decomposition algorithm for stochastic linear programs, Annals of Operations Research 39 (1992) 69–95.
[11] G. Infanger, D.P. Morton, Cut sharing for multistage stochastic linear programs with interstage dependency, Mathematical Programming 75 (1996) 241–256.
[12] M.E.P. Maceira, V.S. Duarte, D.D.J. Penna, L.A.M. Moraes, A.C.G. Melo, Ten years of application of stochastic dual dynamic programming in official and agent studies in Brazil: description of the NEWAVE program, in: 16th Power Systems Computation Conference, Glasgow, 2008.
[13] M.E.P. Maceira, V.S. Duarte, D.D.J. Penna, M.P. Tcheou, An approach to consider hydraulic coupled systems in the construction of equivalent reservoir model in hydrothermal operation planning, in: 17th Power Systems Computation Conference, Stockholm, 2011.
[14] M. Pereira, Optimal stochastic operations scheduling of large hydroelectric systems, International Journal of Electrical Power and Energy Systems 11 (1989) 161–169.
[15] M. Pereira, L. Pinto, Multi-stage stochastic optimization applied to energy planning, Mathematical Programming 52 (1991) 359–375.
[16] A. Philpott, V. de Matos, Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion, Technical Report Electric Power Optimization Centre, University of Auckland, 2010.
[17] A. Philpott, Z. Guan, On the convergence of stochastic dual dynamic programming and related methods, Operations Research Letters 36 (2008) 450–455.
[18] S. Rebennack, A unified state-space and scenario tree framework for multi-stage stochastic optimization: an application to emission-constrained hydrothermal scheduling, Ph.D. Thesis, University of Florida, 2010.
[19] A. Shapiro, Analysis of stochastic dual dynamic programming method, European Journal of Operational Research 209 (2011) 63–72.
[20] R.M. Van Slyke, R. Wets, *L*-shaped linear programs with applications to optimal control and stochastic programming, SIAM Journal on Applied Mathematics 17 (1969) 638–663.