Mathematical Programming 75 (1996) 241-256

Cut sharing for multistage stochastic linear programs with interstage dependency

Gerd Infanger [a,1,*], David P. Morton [b,2]

[a] Department of Operations Research, Stanford University, Stanford, CA 94305-4023, USA
[b] Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX 78712, USA

Received 1 September 1994; revised manuscript received 1 July 1995

Abstract

Multistage stochastic programs with interstage independent random parameters have recourse functions that do not depend on the state of the system. Decomposition-based algorithms can exploit this structure by sharing cuts (outer-linearizations of the recourse function) among different scenario subproblems at the same stage. The ability to share cuts is necessary in practical implementations of algorithms that incorporate Monte Carlo sampling within the decomposition scheme. In this paper, we provide methodology for sharing cuts in decomposition algorithms for stochastic programs that satisfy certain interstage dependency models. These techniques enable sampling-based algorithms to handle a richer class of multistage problems, and may also be used to accelerate the convergence of exact decomposition algorithms.

Keywords: Stochastic programming; Decomposition algorithms; Monte Carlo sampling; Interstage dependency

1. Introduction

An important class of practical planning problems involves sequential allocation of scarce resources among competing activities in the face of uncertainty with respect to future states of the system. Multistage stochastic programming with recourse provides an

* Corresponding author. e-mail: infanger@or.stanford.edu.

[1] Research leading to this work was partially supported by the Department of Energy Contract DE-FG03-92ER25116-A002; the Office of Naval Research Contract N00014-89-J-1659; the National Science Foundation Grants ECS-8906260, DMS-8913089; and the Electric Power Research Institute Contract RP 8010-09, CSA-4005335.

[2] This author's work was supported in part by the National Research Council under a Research Associateship at the Naval Postgraduate School, Monterey, California.

PII S0025-5610(96)00014-7

attractive modeling framework for many such problems. Van Slyke and Wets [24] first applied Benders decomposition to the two-stage stochastic linear program (SLP-2) via the L-shaped algorithm; Birge [1] extended their work to the multistage program (SLP-T). These algorithms decompose a problem by stage and scenario and iteratively improve upper and lower bounds on the optimal objective function by successively adding *cuts* to the subproblems of each stage; the cuts form an outer linearization of the future cost (recourse) functions. The algorithms terminate when the difference in objective bounds is sufficiently small; in this sense, the L-shaped method and its multistage counterpart are, within numerical tolerances, exact algorithms.

In many practical problems, the number of scenarios is so large that exact solution techniques are impractical. In this case, bounding and approximation schemes may prove useful; see, for example, Birge and Wets [4,5], Edirisinghe and Ziemba [9], Frauendorfer [11,12], and Kall et al. [17]. However, these schemes can be difficult to apply to problems with many random parameters due to the computational effort required to estimate high dimensional expectations. Monte Carlo sampling-based algorithms (suggested in 1961 by Dantzig and Madansky [8]) provide an attractive alternative for such problems. Stochastic quasigradient algorithms (see Ermoliev [10]) are sampling-based algorithms for stochastic programming. For two-stage models, Dantzig and Glynn [7], Infanger [16], and Higle and Sen [14] have proposed decomposition-based algorithms that incorporate Monte Carlo sampling at each iteration. Infanger [15] and Pereira and Pinto [19] have put forward decomposition and sampling-based algorithms for SLP-T.

In this paper, we are concerned with decomposition and sampling-based algorithms for multistage stochastic programs. In the multistage problem, if the stochastic parameters are interstage independent then the future cost functions do not depend on the current scenario, and hence, cuts generated for a particular scenario are also valid for any other scenario at the same stage. The ability to share cuts among different scenario subproblems at the same stage is critical for practical implementations of multistage sampling-based algorithms. Even if sufficient memory were available to store cuts separately at each node in the scenario tree, the frequency with which any particular node is revisited may be quite low (e.g., zero) in a sampling oriented algorithm. The purpose of this paper is to provide methodology for sharing cuts in multistage problems with stochastic parameters that exhibit certain types of *interstage dependency*. While the primary purpose of this paper is to enable sampling-based algorithms to handle a richer class of multistage models, the techniques discussed may also accelerate convergence of exact decomposition algorithms; see e.g., the implementations due to Gassmann [13] and Birge et al. [2]. Morton [18] provides empirical evidence that cut sharing can accelerate convergence of such algorithms.

This paper is organized as follows. In Section 2 we present a mathematical formulation of SLP-T. In Section 3 we state an exact Benders decomposition algorithm for SLP-T; this serves to review decomposition subproblems and optimality cut generation techniques that are required in the remainder of the paper. The cut sharing method for a linear lag-one interstage dependency model for right-hand-side vectors is detailed in

Section 4. Extensions of the cut sharing technique to higher order linear lag models and other more general interstage dependency models are described in Section 5 and Section 6, respectively. We present the cut sharing methodology under the assumption that cuts are computed exactly (i.e., they are calculated via population means) only in order to simplify the presentation. This is sufficient for application to the sampling-based algorithm of Pereira and Pinto [19]. The generalizations required for algorithms that use sample mean estimates for cutting planes are straightforward. See, for example, Infanger [15] for a discussion of issues associated with using sample mean estimates of cutting planes in a multistage framework. The paper is summarized in Section 7.

2. Problem statement

A T-stage stochastic linear program with recourse (SLP-T) may be formulated as follows.

$$\text{(SLP-T)} \quad \min_{x_1} \quad c_1 x_1 + E_{\xi_2} h_2(x_1, \xi_2)$$
$$\text{s.t.} \quad A_1 x_1 = b_1,$$
$$x_1 \geqslant 0,$$

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

$$h_t(x_{t-1}, \xi_t) = \min_{x_t} \quad c_t x_t + E_{\xi_{t+1} | \xi_2, \ldots, \xi_t} h_{t+1}(x_t, \xi_{t+1})$$
$$\text{s.t.} \quad A_t x_t = b_t + B_t x_{t-1},$$
$$x_t \geqslant 0,$$

and where $h_{T+1} \equiv 0$. $\xi_t = \text{vec}(b_t, c_t, B_t, A_t)$, $t = 2, \ldots, T$, denote random vectors; the vec operator transforms matrices into vectors by reading them columnwise. The sample space for stage t is denoted Ω_t and a sample point (scenario) in Ω_t is denoted ω_t. We use notation $\xi_t^{\omega_t}$, or alternatively $\xi_t(\omega_t)$, to represent a stage t realization. For notational convenience, we assume the existence of a first stage sample space; Ω_1 is a singleton set and $\xi_1^{\omega_1} = \text{vec}(b_1, c_1, A_1)$ represents the known state at the time decisions are made in the first stage; clearly, $p_1^{\omega_1}$ has value one. We will suppress first stage scenario indices, when convenient, to simplify notation. $A_t^{\omega_t}$, $t = 1, \ldots, T$ is an $m_t \times n_t$ matrix and the remaining matrices and vectors are dimensioned to conform. We assume that SLP-T has relatively complete recourse (all stage t subproblems are feasible for any feasible stage $t-1$ variable x_{t-1}, see, e.g., Wets [26]) and that each stage's feasible region is bounded; these requirements ensure that the subproblems formed in the course of the decomposition algorithm outlined in the next section have finite optimal solutions. While these assumptions are of little practical restriction, particularly in sampling-based algorithms, we note that infeasible or unbounded subproblems can be handled by the method of Van Slyke and Wets [24].

We assume the following special structure of the stage t sample space

$$\Omega_t = \Sigma_1 \times \cdots \times \Sigma_t \quad \text{for } t = 1, \ldots, T.$$

A stage t scenario, $\omega_t \in \Omega_t$, may be expressed $\omega_t = (\sigma_1, \ldots, \sigma_t)$ where $\sigma_i \in \Sigma_i$, $i = 1, \ldots, t$. The scenarios of SLP-T form a multistage scenario tree: each stage $t \geqslant 2$ scenario, ω_t, has a unique stage $t-1$ ancestor denoted $a(\omega_t)$, and a stage $t < T$ scenario has a set of descendent scenarios denoted $\Delta(\omega_t)$. We assume that ξ_t has finite support and a probability mass function given by $P\{\xi_t = \xi_t^{\omega_t}\} = p_t^{\omega_t}$. Conditional probability mass functions are written

$$P\{\xi_t = \xi_t^{\omega_t} \mid \xi_{t-1} = \xi_{t-1}^{\omega_{t-1}}\} = p_t^{\omega_t \mid \omega_{t-1}}.$$

In the special case of interstage independence we may write $\xi_t^{\omega_t} = \xi_t^{\sigma_t}$ and $p_t^{\omega_t \mid \omega_{t-1}} = p_t^{\sigma_t}$.

In SLP-T, decisions occur and uncertainties unfold in the following manner. A first stage decision, x_1, is made with knowledge of (b_1, c_1, A_1) and distributions on future data; next, an observation, $\xi_2^{\omega_2}$, is revealed and the second stage decision $x_2^{\omega_2}$ is made knowing this data, and the corresponding conditional distributions on future data, etc. The goal is to find a first stage decision, x_1, that minimizes the expected cost of operating the system modeled by SLP-T over T stages.

3. A decomposition algorithm for SLP-T

In this section, we review a decomposition algorithm that decomposes SLP-T into subproblems by stage and scenario. In the course of the algorithm, the subproblems' right-hand-sides and feasible regions continually change as information is passed between subproblems. A subproblem passes resources to its descendents in the form of the vector $B_{t+1}^{\omega_{t+1}} x_t^{a(\omega_{t+1})}$ and receives dual prices from its descendents to compute cuts that represent an outer linearization of the recourse function. The stage t $(1 \leqslant t < T)$ subproblem under scenario ω_t, denoted sub(ω_t), has the following form:

$$\min_{x_t, \theta_t} \quad z_t = c_t^{\omega_t} x_t + \theta_t \tag{1}$$

$$\text{s.t.} \quad A_t^{\omega_t} x_t = b_t^{\omega_t} x_t + B_t^{\omega_t} x_{t-1}^{a(\omega_t)},$$

$$-\vec{G}_t^{\omega_t} x_t + e\theta_t \geqslant \vec{g}_t^{\omega_t},$$

$$x_t \geqslant 0.$$

The rows of $\vec{G}_t^{\omega_t} \in \mathbb{R}^{I_t^{\omega_t} \times n_t}$ contain cut gradients; the elements of the vector $\vec{g}_t^{\omega_t}$ are cut intercepts, and e denotes the vector of all 1's. The number of cuts appended to sub(ω_t) at an arbitrary point in the algorithm is denoted $I_t^{\omega_t}$; as the decomposition algorithm proceeds, this value grows. The stage T subproblems are similar to (1) except that the cut constraints and scalar variable θ_t are absent.

The dual of sub(ω_t) may be written:

$$\begin{aligned} \max_{\pi_t, \alpha_t} \quad & z_t = \pi_t\left(b_t^{\omega_t} + B_t^{\omega_t} x_{t-1}^{a(\omega_t)}\right) + \alpha_t \vec{g}_t^{\omega_t} \\ \text{s.t.} \quad & \pi_t A_t^{\omega_t} - \alpha_t \vec{G}_t^{\omega_t} \leqslant c_t^{\omega_t}, \\ & e^T \alpha_t = 1, \\ & \alpha_t \geqslant 0, \end{aligned} \tag{2}$$

where if $t = T$, the cut gradient matrix $\vec{G}_t^{\omega_t}$, cut intercept vector $\vec{g}_t^{\omega_t}$, and corresponding dual vector α_t are absent. In this and subsequent sections, we require use of both the matrix of cut gradients, denoted $\vec{G}_t^{\omega_t}$, and a particular cut gradient, denoted $G_t^{\omega_t}$. We similarly distinguish between the vector of cut intercepts, $\vec{g}_t^{\omega_t}$ and a scalar cut intercept $g_t^{\omega_t}$. Let $(z_t^{\omega_t}, \pi_t^{\omega_t}, \alpha_t^{\omega_t})$ denote an optimal solution of (2). When the descendents of sub(ω_t) are solved at a particular stage t decision, say $x_t^{\omega_t}$, the cut gradient and intercept that may subsequently be appended to sub(ω_t) are found via

$$G_t^{\omega_t} = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} \pi_{t+1}^{\omega_{t+1}} B_{t+1}^{\omega_{t+1}} \tag{3}$$

and

$$g_t^{\omega_t} = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} \pi_{t+1}^{\omega_{t+1}} b_{t+1}^{\omega_{t+1}} + \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} \alpha_{t+1}^{\omega_{t+1}} \vec{g}_{t+1}^{\omega_{t+1}} \tag{4}$$

where the second term in (4) is absent if $t = T - 1$. An equivalent expression for $g_t^{\omega_t}$, that does not require this caveat, is

$$g_t^{\omega_t} = \sum_{\omega_{t+1} \in \Delta(\omega_t)} p_{t+1}^{\omega_{t+1}|\omega_t} z_{t+1}^{\omega_{t+1}} - G_t^{\omega_t} x_t^{\omega_t}. \tag{5}$$

A decomposition algorithm for SLP-T is summarized in Fig. 1; in designing such an algorithm, we have considerable flexibility with respect to the order in which subproblems of the scenario tree are solved. The algorithm of Fig. 1 uses the *fastpass* tree traversing strategy; see Wittrock [27], Gassmann [13], and Morton [18] for further discussion of alternate tree traversing strategies. There are number of other enhancements and variants of this algorithm; for example, Birge and Louveaux [3] propose a multicut algorithm, Ruszczyński [21] uses multicuts and a quadratic proximal term, and Wets [25] describes a bunching technique designed to efficiently solve a collection of same-stage subproblems. However, we will not pursue these issues here because the primary purpose of this section is to provide necessary background with respect to basic cut generation techniques in decomposition algorithms for SLP-T. Note that, as SLP-T is a linear program, finite convergence of the decomposition algorithm is ensured; see, e.g., Birge [1].

We close this section by stating a result concerning valid cut generation. A *valid cut* is defined to be a hyperplane that lies below the recourse function. Proposition 1 states that dual feasible (but not necessarily optimal) price vectors of the descendent subproblems generate valid cuts; see [18] for a proof.

Proposition 1. *Consider* SLP-T *and in particular,* sub(ω_t); *let* $K_t = |\Delta(\omega_t)|$.

(i) *If* $t = T-1$ *and* $(\pi_T^1, \ldots, \pi_T^{K_{T-1}})$ *is a collection of dual feasible vectors for the descendents of* sub(ω_{T-1}), *then these dual vectors generate, via* (3) *and* (4), *a valid cut for* sub(ω_{T-1}).

(ii) *If* $t \leqslant T-2$, *the descendents of* sub(ω_t) *contain valid cuts, and* $[(\pi_{t+1}^1, \alpha_{t+1}^1), \ldots, (\pi_{t+1}^{K_t}, \alpha_{t+1}^{K_t})]$ *is a collection of dual feasible vectors for the descendents of* sub(ω_t), *then these dual vectors generate, via* (3) *and* (4), *a valid cut for* sub(ω_t).

Inductive application of Proposition 1 can be used to verify that cuts generated by the decomposition algorithm for SLP-T with stochastic parameters exhibiting interstage

step 0 define *toler* ≥ 0; let $\bar{z} = +\infty$; initialize sub(ω_t)'s set of cuts with $\theta_t \geq -M$, $\forall \omega_t \in \Omega_t$, $t = 1, \ldots, T-1$;

step 1 solve sub(ω_1) and obtain (x_1, θ_1);
let $\underline{z} = c_1 x_1 + \theta_1$;

step 2 **(forward pass:)**
do $t = 2$ to T
  do $\omega_t \in \Omega_t$
    form RHS of sub(ω_t): $B_t^{\omega_t} x_{t-1}^{a(\omega_t)} + b_t^{\omega_t}$;
    solve sub(ω_t) and obtain $x_t^{\omega_t}$;
    if $t = T$ also obtain $\pi_T^{\omega_T}$;
  enddo
enddo
let $\hat{z} = c_1 x_1 + \sum_{t=2}^{T} \sum_{\omega_t \in \Omega_t} p_t^{\omega_t} c_t^{\omega_t} x_t^{\omega_t}$;

step 3 **(stopping rule:)**
if $\hat{z} < \bar{z}$ then let $\bar{z} = \hat{z}$ and $x_1^* = x_1$;
if $\bar{z} - \underline{z} \leq \min(|\bar{z}|, |\underline{z}|) \cdot$ *toler* then stop: x_1^* is a solution yielding an objective function value within (100 · *toler*)% of optimal;

step 4 **(backward pass:)**
do $t = T-1$ downto 2
  do $\omega_t \in \Omega_t$
    augment sub(ω_t)'s set of cuts with $\theta_t - G_t^{\omega_t} x_t \geq g_t^{\omega_t}$;
    form RHS of sub(ω_t): $B_t^{\omega_t} x_{t-1}^{a(\omega_t)} + b_t^{\omega_t}$;
    solve sub(ω_t) and obtain $(\pi_t^{\omega_t}, \alpha_t^{\omega_t})$;
  enddo
enddo
augment sub(ω_1)'s set of cuts with $\theta_1 - G_1 x_1 \geq g_1$;
goto step 1;

Fig. 1. Decomposition algorithm for SLP-T.

independence, may be shared among subproblems at the same stage. Specifically, cuts generated for a particular stage t subproblem, say sub(ω_t), are valid for any other stage t subproblem, say sub(ω_t'), under interstage independence, because the descendents of sub(ω_t) and sub(ω_t') have identical dual feasible regions *and* the stochastic parameters in the cut gradient and intercept formulas (3) and (4) for the respective descendent problems are also identical. The base case of the inductive hypothesis (i.e., descendent subproblems contain valid cuts) is verified by applying part (i) of Proposition 1.

4. Cut sharing under linear lag-one RHS dependency

Pereira and Pinto [20] first suggested the possibility of incorporating autoregressive sequences for the right-hand-side vectors in a multistage decomposition and sampling-based algorithm. In this section, and Section 5 and Section 6, we show how the ability to share cuts can be extended from the interstage independence case (described above) to several different interstage dependency structures. We begin by assuming a simple linear lag-one dependency model of the right-hand-side vectors

$$b_t = R_{t-1} b_{t-1} + \eta_t \quad \text{for } t = 2, \dots, T, \tag{6}$$

where η_t, $t = 2, \dots, T$, are random m_t-vectors and the matrices R_t, $t = 1, \dots, T-1$, are known; without loss of generality, let $R_1 = 0$. We assume

$$\text{vec}(\eta_t, c_t, B_t, A_t), \quad t = 2, \dots, T \quad \text{are independent.} \tag{7}$$

This dependency model is a generalization of the well-known autoregressive lag-one model (AR1) in which $R_t = R$ does not depend on t. The random right-hand-side vectors of a multistage stochastic program may exhibit seasonal patterns. In such cases, the AR1 model can be used to separately analyze each period of the season; this can lead to a model of the form (6) in which $R_{t+p} = R_t$ for $t \geqslant 2$, where p denotes the length of the season. Box and Jenkins [6] and Tsay and Tiao [23] discuss parameter estimation and associated issues for univariate autoregressive models. See Tiao and Box [22], and references cited therein, for analyses and parameter estimation procedures for vector autoregressive models; these analyses typically assume that η_t is normally distributed. Note that while the AR1 model is a generalization of serial independence, it reflects a very specific interstage dependency structure. For example, the structure implies that the number of descendents and their corresponding probability mass functions are the same within each time stage.

4.1. Illustration: The SLP-3 case

In order to illustrate the basic idea behind the cut sharing methodology, we begin with the simple case of $T = 3$ and focus on cut calculations for the second stage subproblems as this is the only nontrivial case for SLP-3. The second stage cut gradient and intercept are given by

$$G_2^{\omega_2} = \sum_{\omega_3 \in \Delta(\omega_2)} p_3^{\omega_3 | \omega_2} \pi_3^{\omega_3} B_3^{\omega_3} \tag{8}$$

and

$$g_2^{\omega_2} = \sum_{\omega_3 \in \Delta(\omega_2)} p_3^{\omega_3 | \omega_2} \pi_3^{\omega_3} b_3^{\omega_3}. \tag{9}$$

Since c_3 and A_3 are independent of the second stage parameters, the collection of dual variables $\{\pi_3^{\omega_3} : \omega_3 \in \Delta(\omega_2)\}$ from the solution of one set of descendents is feasible for the descendents of any second stage scenario. Thus, by Proposition 1, part (i), these dual variables will generate a valid cut for any second stage subproblem, and we now show that these cuts may be generated and *recalled* in subsequent iterations in closed form. Due to the dual feasibility structure, we may label the dual variables with a σ_3 index. In this framework, the conditional probability mass function $p_3^{\omega_3 | \omega_2} = p_3^{\sigma_3}$ does not depend on ω_2. Furthermore, as B_3 is independent of the second stage random parameters, the cut gradient formula (8) is valid for all $\omega_2 \in \Omega_2$, and the ω_2 index on G_2 may be dropped. The cut intercept formula, however, involves b_3 which contains interstage dependency according to (6). Upon substitution of the lag-one model into (9) it is clear that

$$g_2^{\omega_2} = g_2^{\text{ind}} + g_2^{\text{dep}}(\omega_2), \tag{10}$$

where

$$g_2^{\text{ind}} = \sum_{\sigma_3 \in \Sigma_3} p_3^{\sigma_3} \pi_3^{\sigma_3} \eta_3^{\sigma_3}, \tag{11}$$

$$g_2^{\text{dep}}(\omega_2) = \underbrace{\left[\sum_{\sigma_3 \in \Sigma_3} p_3^{\sigma_3} \pi_3^{\sigma_3} \right]}_{\bar{\pi}_3} R_2 b_2^{\omega_2}. \tag{12}$$

In (10), the second stage cut intercept has been expressed as the sum of an ω_2-independent (ind) term and an ω_2-dependent (dep) term. The scenario dependent term given by (12) has a simple structure. Construction of $g_2^{\text{dep}}(\omega_2)$ for a particular second stage scenario requires knowledge of the second stage right-hand-side realization, $b_2^{\omega_2}$, and the expected value of the third stage dual variables, $\bar{\pi}_3$, used to generate the original cut. Thus, in a three-stage model the following information is stored for each cut: (i) cut gradient, G_2, (ii) scenario independent cut intercept term, g_2^{ind}, and (iii) expected dual vector, $\bar{\pi}_3$. Given a second stage scenario, ω_2, valid cuts may then be formed for sub(ω_2) from this information by computing the scenario dependent cut intercept (12) and then computing the cut intercept via (10). We regard (12) as a closed form, scenario dependent correction term for the second stage cut intercept.

4.2. The SLP-T case

SLP-3 does not reveal all of the complexities associated with the lag-one model for a general multistage problem. In a T-stage model, the analysis for the three-stage case is valid for cuts computed for stage $T-1$. However, cut intercepts for a general stage t are given by (4), and this formula requires the stage $t+1$ cut intercepts which in turn require the stage $t+2$ cut intercepts, etc. Computationally, it is clearly preferable to avoid the recursive calculation of these respective intercept terms through the exponentially growing scenario tree; in addition to being prohibitively expensive, such computation would require storing the set of dual variables $\{(\pi_{t+1}^{\omega_{t+1}}, \alpha_{t+1}^{\omega_{t+1}}): \omega_{t+1} \in \Delta(\omega_t)\}$ used to generate each cut. Thus, as in the three-stage case, we seek a closed form cut intercept correction term for SLP-T with the lag-one dependency model (6).

Observe that the dual feasible region of a stage t subproblem (2) does not depend on the right-hand-side vectors. This is an important observation because dual feasibility ensures that valid cuts can be generated; again, see Proposition 1. This is to be contrasted with the case in which A_t, c_t, or B_t contain interstage dependencies; see subsequent Section 6.2. Recall that the stage t subproblem is assumed to have m_t rows (excluding cuts) and l_t cuts. We define $\overline{\mathscr{P}}_t$ to be the $l_{t-1} \times m_t$ matrix whose rows contain the expected value of the structural constraint dual variables, $\overline{\pi}_t = E_{\sigma_t} \pi_t^{\sigma_t}$. Similarly, $\overline{\mathscr{A}}_t$ is defined as the $l_{t-1} \times l_t$ matrix whose rows contain the expected value of the cut constraint dual variables, $\overline{\alpha}_t = E_{\sigma_t} \alpha^{\sigma_t}$. As can be seen from (4), these dual variables are used in the cut computation for stage $t-1$. The ith row of $\overline{\mathscr{P}}_t$ and $\overline{\mathscr{A}}_t$ contain the expected value of the dual variables used to generate the ith stage $t-1$ cut. In Theorem 2 we show

$$g_t^{\omega_t} = g_t^{\text{ind}} + g_t^{\text{dep}}(\omega_t), \tag{13}$$

where

$$g_t^{\text{dep}}(\omega_t) = \left[\overline{\pi}_{t+1} + \overline{\alpha}_{t+1} D_{t+1}\right] R_t b_t^{\omega_t}, \tag{14}$$

and the matrix D_t is defined recursively

$$D_t = \left[\overline{\mathscr{P}}_{t+1} + \overline{\mathscr{A}}_{t+1} D_{t+1}\right] R_t, \quad D_T = 0. \tag{15}$$

Note that an explicit formula for g_t^{ind} is not necessary; when we generate a cut for the first time (as opposed to subsequently recomputing it for another scenario) we may first compute $g_t^{\omega_t}$ from (5) and then subtract the dependent term calculated via (14) to obtain g_t^{ind}.

Theorem 2. *Assume the lag-one model* (6) *and* (7). *The cut intercepts for stage t, $t = 2, \ldots, T-1$, are given by* (13), (14) *and* (15).

Proof. We proceed by induction. The base case is $t = T-1$ and the expression for g_{T-1} is found by the method for the three-stage case; see Section 4.1. The inductive hypothesis is then (13) and (14) and we verify the same expressions with t decremented by one. It is convenient to adopt the vector analog of (14):

$$\vec{g}_t^{\,\mathrm{dep}}(\omega_t) = \left[\overline{\mathscr{P}}_{t+1} + \overline{\mathscr{A}}_{t+1} D_{t+1}\right] R_t b_t^{\omega_t}. \tag{16}$$

A stage $t-1$ intercept for scenario ω_{t-1} is defined as

$$g_{t-1}^{\omega_{t-1}} = \sum_{\omega_t \in \Delta(\omega_{t-1})} p_t^{\omega_t|\omega_{t-1}} \pi_t^{\omega_t} b_t^{\omega_t} + \sum_{\omega_t \in \Delta(\omega_{t-1})} p_t^{\omega_t|\omega_{t-1}} \alpha_t^{\omega_t} \vec{g}_t^{\,\omega_t}. \tag{17}$$

Substitution of the lag-one model (6) into the first term on the right-hand-side of (17) yields

$$\overline{\pi}_t R_{t-1} b_{t-1}^{\omega_{t-1}} + E_{\sigma_t} \pi_t^{\sigma_t} \eta_t^{\sigma_t}. \tag{18}$$

Substitution of the inductive hypothesis (13), (14) and the lag-one model (6) into the second term on the right-hand-side of (17) yields

$$\overline{\alpha}_t \vec{g}_t^{\,\mathrm{ind}} + \overline{\alpha}_t\left[\overline{\mathscr{P}}_{t+1} + \overline{\mathscr{A}}_{t+1} D_{t+1}\right] R_t R_{t-1} b_{t-1}^{\omega_{t-1}} + E_{\sigma_t} \alpha_t^{\sigma_t}\left[\overline{\mathscr{P}}_{t+1} + \overline{\mathscr{A}}_{t+1} D_{t+1}\right] R_t \eta_t^{\sigma_t}. \tag{19}$$

Using the definition of D_t from (15) and partitioning (18) and (19) into the scenario dependent and scenario independent parts yields the desired result. □

In SLP-T with lag-one model (6), the following information is stored for each stage t cut: (i) cut gradient, G_t, (ii) the scenario independent cut intercept term, g_t^{ind}, and (iii) the *cumulative expected dual vector* $[\overline{\pi}_{t+1} + \overline{\alpha}_{t+1} D_{t+1}]R_t$. Given a particular stage t scenario ω_t, valid cuts may be formed for sub(ω_t) from this information by computing the scenario dependent cut intercept term (14) and then computing the cut intercept via (13). When computing a cut for the first time, its associated cumulative expected dual vector can be generated from the set of cumulative expected dual vectors contained in the descendent scenarios. This follows from (14), (15), and the fact that the rows of the matrix D_t are the appropriate cumulative expected dual vectors. In other words, there is no need to explicitly store the matrices $\overline{\mathscr{P}}_{t+1}$ and $\overline{\mathscr{A}}_{t+1}$. Thus, the additional storage requirement, relative to the interstage independence case, involves saving the vector $[\overline{\pi}_{t+1} + \overline{\alpha}_{t+1} D_{t+1}]R_t \in \mathbb{R}^{m_t}$ for each cut.

5. Cut sharing under higher order linear lag models

In some settings, greater flexibility is necessary for modeling interstage dependency than is provided by the lag-one models described above. In this section, we describe a higher order lag model that is a generalization of vector ARMA (autoregressive moving average) models in the same way that the lag-one model of Section 4 is a generalization of the AR1 model. Again, see [6], [22], and [23] for further discussion of ARMA models and associated estimation and analysis issues. The higher order lag model we consider in this section is

$$b_t = \sum_{j=1}^{t-1} \left(R_j^t b_j + S_j^t \eta_j \right) + \eta_t \quad \text{for } t = 2, \ldots, T, \tag{20}$$

where once again, we assume the independence structure given by (7) and R_j^t and S_j^t are appropriately dimensioned deterministic matrices (some of which may be zero). In this case, we have the following theorem regarding scenario dependent cut correction terms.

Theorem 3. *Assume the higher order lag model* (20) *and* (7). *The cut intercepts for stage* t, $t = 2, \ldots, T-1$, *are given by* $g_t^{\omega_t} = g_t^{\text{ind}} + g_t^{\text{dep}}(\omega_t)$ *and*

$$\begin{aligned} & g_t^{\text{dep}}(\omega_t) \\ & = \overline{\pi}_{t+1} \sum_{j=1}^{t} \left(R_j^{t+1} b_j^{\omega_j} + S_j^{t+1} \eta_j^{\sigma_j} \right) + \overline{\alpha}_{t+1} \sum_{i=t+1}^{T} D_{t+1}^i \sum_{j=1}^{t} \left(R_j^i b_j^{\omega_j} + S_j^i \eta_j^{\sigma_j} \right), \end{aligned} \tag{21}$$

where D_t^i, $i \geqslant t$, *is defined as*

$$\begin{cases} D_t^t = \overline{\mathscr{P}}_{t+1} R_t^{t+1} + \overline{\mathscr{A}}_{t+1} \sum_{i=t+1}^{T} D_{t+1}^i R_t^i, \\ D_t^{t+1} = \overline{\mathscr{P}}_{t+1} + \overline{\mathscr{A}}_{t+1} D_{t+1}^{t+1}, \\ D_t^i = \overline{\mathscr{A}}_{t+1} D_{t+1}^i, \quad i \geqslant t+2, \end{cases} \tag{22}$$

with $D_T^T = 0$ *and* $\overline{\mathscr{A}}_T = 0$.

Proof. We proceed by induction. The base case is $t = T-1$

$$\begin{aligned} g_{T-1}^{\omega_{T-1}} &= E_{\omega_T \mid \omega_{T-1}} \pi_T^{\omega_T} \sum_{j=1}^{T-1} \left(R_j^T b_j^{\omega_j} + S_j^T \eta_j^{\sigma_j} \right) + \text{ind. terms} \\ &= \overline{\pi}_T \sum_{j=1}^{T-1} \left(R_j^T b_j^{\omega_j} + S_j^T \eta_j^{\sigma_j} \right) + \text{ind. terms}. \end{aligned}$$

Verification of the base case is complete since the second term in (21) is absent for $t = T-1$. Throughout this proof, we use ''ind. terms'' to denote terms that contribute to the scenario independent cut intercept term; e.g., g_{T-1}^{ind}.

Now

$$\begin{aligned} g_{t-1}^{\omega_{t-1}} &= E_{\omega_t \mid \omega_{t-1}} \pi_t^{\omega_t} \sum_{j=1}^{t-1} \left(R_j^t b_j^{\omega_j} + S_j^t \eta_j^{\sigma_j} + \eta_t^{\sigma_t} \right) + E_{\omega_t \mid \omega_{t-1}} \alpha_t^{\omega_t} \left(\vec{g}_t^{\text{ind}} + \vec{g}_t^{\text{dep}}(\omega_t) \right) \\ &= \overline{\pi}_t \sum_{j=1}^{t-1} \left(R_j^t b_j^{\omega_j} + S_j^t \eta_j^{\sigma_j} \right) + E_{\omega_t \mid \omega_{t-1}} \alpha_t^{\omega_t} \vec{g}_t^{\text{dep}}(\omega_t) + \text{ind. terms}. \end{aligned}$$

By the inductive hypothesis, i.e., the vector analog of (21), we have

$$E_{\omega_t|\omega_{t-1}}\alpha_t^{\omega_t}\vec{g}_t^{\text{dep}}(\omega_t)=E_{\omega_t|\omega_{t-1}}\alpha_t^{\omega_t}\left[\overline{\mathscr{P}}_{t+1}\sum_{j=1}^{t}\left(R_j^{t+1}b_j^{\omega_j}+S_j^{t+1}\eta_j^{\sigma_j}\right)\right.$$
$$\left.+\overline{\mathscr{A}}_{t+1}\sum_{i=t+1}^{T}D_{t+1}^{i}\sum_{j=1}^{t}\left(R_j^{i}b_j^{\omega_j}+S_j^{i}\eta_j^{\sigma_j}\right)\right].$$

We separate the terms of this expression into an ω_t-dependent term and an $(\omega_2,\dots,\omega_{t-1})$-dependent term. After substitution of the dependency model (20) for $b_t^{\omega_t}$ we achieve

$$\begin{aligned}
&E_{\omega_t|\omega_{t-1}}\alpha_t^{\omega_t}\vec{g}_t^{\text{dep}}(\omega_t)\\
&=\overline{\alpha}_t\left[\overline{\mathscr{P}}_{t+1}\sum_{j=1}^{t-1}\left(R_j^{t+1}b_j^{\omega_j}+S_j^{t+1}\eta_j^{\sigma_j}\right)+\overline{\mathscr{A}}_{t+1}\sum_{i=t+1}^{T}D_{t+1}^{i}\sum_{j=1}^{t-1}\left(R_j^{i}b_j^{\omega_j}+S_j^{i}\eta_j^{\sigma_j}\right)\right.\\
&\quad+\overline{\mathscr{P}}_{t+1}R_t^{t+1}\sum_{j=1}^{t-1}\left(R_j^{t}b_j^{\omega_j}+S_j^{t}\eta_j^{\sigma_j}\right)\\
&\quad\left.+\overline{\mathscr{A}}_{t+1}\sum_{i=t+1}^{T}D_{t+1}^{i}R_t^{i}\sum_{j=1}^{t-1}\left(R_j^{t}b_j^{\omega_j}+S_j^{t}\eta_j^{\sigma_j}\right)\right]\\
&\quad+\text{ind. terms.}
\end{aligned}\tag{23}$$

By rearranging the bracketed term of (23) we find that the coefficients of $\sum_{j=1}^{t-1}(R_j^{t}b_j^{\omega_j}+S_j^{t}\eta_j^{\sigma_j})$ are D_t^i, $i=t,\dots,T$, as defined by (22) and this completes the proof. □

6. Cut sharing under more general dependency models

In this section, we extend the results of the previous two sections in two directions. First, we describe a more general RHS lag-one model. Second, we consider interstage dependency of stochastic parameters in the transition matrices B_t.

6.1. More general lag-one models for the RHS

There are many generalizations of the lag-one model (6), but not all such models are tractable with respect to deriving closed form cut intercept correction terms. The analysis of Section 4 clearly shows the importance of additivity; a general lag-one process $b_t=f_t(b_{t-1},\eta_t)$ does not yield closed form formulae for the intercept correction terms. Moreover, an additive model of the following form:

$$b_t=f_t(b_{t-1})+\eta_t\quad\text{for } t=3,\dots,T,$$

where $f_t : \mathbb{R}^{m_{t-1}} \to \mathbb{R}^{m_t}$ is tractable for a three-stage model, but breaks down in the general case. Thus, in addition to additivity, the linearity of $R_{t-1} b_{t-1}$ plays an important role in deriving the correction term; see the proof of Theorem 2. In Corollary 4 below, we essentially recover the result of Theorem 2 under the generalized lag-one dependency model

$$b_t = R_{t-1}\left[f_{t-1}(v_{t-1}) + b_{t-1}\right] + \eta_t \quad \text{for } t = 2, \ldots, T, \tag{24}$$

where $v_t = \text{vec}(\eta_t, c_t, B_t, A_t)$, $f_t : \mathbb{R}^{N_t} \to \mathbb{R}^{m_{t-1}}$, and $N_t = m_t + n_t + m_t \cdot n_{t-1} + m_t \cdot n_t$. The proof of Corollary 4 is similar to that of Theorem 2 and is omitted.

Corollary 4. *Assume the lag-one model* (24) *and* (7). *The cut intercepts for stage t, $t = 2, \ldots, T-1$, are given by $g_t^{\omega_t} = g_t^{\text{ind}} + g_t^{\text{dep}}(\omega_t)$ and*

$$g_t^{\text{dep}}(\omega_t) = \left[\bar{\pi}_{t+1} + \bar{\alpha}_{t+1} D_{t+1}\right] R_t\left[f_t(v_t^{\sigma_t}) + b_t^{\omega_t}\right], \tag{25}$$

where D_t is defined by (15).

The lag-one model (24) may be useful, for example, when the random demand for a finished product in one time period is correlated with, say, the price of raw materials and/or demand in the previous period.

In similar fashion to (24) the higher order lag model may be generalized as

$$b_t = \sum_{j=1}^{t-1} \left(R_j^t b_j + f_j^t(v_j)\right) + \eta_t \quad \text{for } t = 2, \ldots, T, \tag{26}$$

where $f_j^t : \mathbb{R}^{N_t} \to \mathbb{R}^{m_t}$. We now have the following corollary to Theorem 3.

Corollary 5. *Assume the higher order lag-one model* (26) *and* (7). *The cut intercepts for stage t, $t = 2, \ldots, T-1$, are given by $g_t^{\omega_t} = g_t^{\text{ind}} + g_t^{\text{dep}}(\omega_t)$ and*

$$g_t^{\text{dep}}(\omega_t) = \bar{\pi}_{t+1} \sum_{j=1}^{t} \left(R_j^{t+1} b_j^{\omega_j} + f_j^{t+1}(v_j^{\sigma_j})\right) + \bar{\alpha}_{t+1} \sum_{i=t+1}^{T} D_{t+1}^i \sum_{j=1}^{t} \left(R_j^i b_j^{\omega_j} + f_j^i(v_j^{\sigma_j})\right), \tag{27}$$

where D_t^i, $i \geqslant t$, is defined by (22) *with $D_T^T = 0$ and $\bar{\mathscr{A}}_T = 0$.*

6.2. Interstage dependency of B_t

Interstage dependency of the objective function coefficients, c_t, and the structural matrix, A_t, directly affect dual feasibility and hence create significant difficulties with respect to sharing cuts. Interstage dependency of the transition matrices, B_t, also affect dual feasibility via cut gradients; see (3). In general, this also poses significant difficulties; there is, however, one case that can be easily handled. In particular, consider the case in which the right-hand-side satisfies (6) and

$$B_3 = \tilde{R}_2 B_2 + \tilde{\eta}_3, \tag{28}$$

where $\tilde{\eta}_3$ is a random $m_3 \times n_2$-matrix, and $\tilde{R}_2$ is known. In this case, we assume

$$\text{vec}(\eta_2, c_2, B_2, A_2), \text{vec}(\eta_3, c_3, \tilde{\eta}_3, A_3), \text{vec}(\eta_t, c_t, B_t, A_t), \ t = 4, \ldots, T,$$

are independent

Under this dependency model, the cut intercept is handled in identical fashion to that of Theorem 2 and cut gradients calculated for stages $3, \ldots, T-1$ via (3) remain unchanged. The cut gradient for the second stage, however, requires a scenario dependent correction term; in analogous fashion to the three-stage cut intercept analysis of Section 4.1 we obtain

$$G_2^{\omega_2} = G_2^{\text{ind}} + G_2^{\text{dep}}(\omega_2), \tag{29}$$

where

$$G_2^{\text{ind}} = \sum_{\sigma_3 \in \Sigma_3} p_3^{\sigma_3} \pi_3^{\sigma_3} \tilde{\eta}_3(\sigma_3), \tag{30}$$

$$G_2^{\text{dep}}(\omega_2) = \underbrace{\left[\sum_{\sigma_3 \in \Sigma_3} p_3^{\sigma_3} \pi_3^{\sigma_3} \right]}_{\overline{\pi}_3} \tilde{R}_2 B_2^{\omega_2}. \tag{31}$$

In summary, we have obtained closed form scenario correction terms for cut formulas when the right-hand-side satisfies a lag-one model for all stages and the transition matrices satisfy a lag-one model only through the third stage.

7. Summary

In many applications modeled by multistage stochastic linear programming, the number of scenarios is so large that exact solution techniques are not computationally practical. It has long been recognized that incorporating Monte Carlo sampling *within* a decomposition scheme might provide an attractive approach for solving problems with many scenarios, and recently such algorithms have been proposed for both two-stage and multistage problems. One of the (rather demanding) assumptions typically made in multistage models solved by decomposition and sampling-based algorithms is interstage independence of the stochastic parameters. We have shown that certain types of interstage dependency structures may be incorporated with relative ease in such algorithms. In addition, the methodology we have presented may also be useful in accelerating convergence (particularly in the early iterations) of exact decomposition algorithms for this class of dependency models.

Acknowledgements

We thank two anonymous referees for their valuable comments.

References

[1] J.R. Birge, "Decomposition and partitioning methods for multistage stochastic linear programs," *Operations Research* 33 (1985) 989–1007.

[2] J.R. Birge, C.J. Donohue, D.F. Holmes and O.G. Svintsitski, "A parallel implementation of the nested decomposition algorithm for multistage stochastic linear programs," Technical Report 94-1, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI (1994).

[3] J.R. Birge and F.V. Louveaux, "A multicut algorithm for two-stage stochastic linear programs," *European Journal of Operational Research* 34 (1988) 384–392.

[4] J.R. Birge and R.J.-B. Wets, "Sublinear upper bounds for stochastic programs with recourse," *Mathematical Programming* 43 (1989) 131–149.

[5] J.R. Birge and R.J.-B. Wets, "Designing approximation schemes for stochastic optimization problems, in particular, for stochastic programs with recourse," *Mathematical Programming Study* 27 (1986) 54–102.

[6] G.E.P. Box and G.M. Jenkins, *Time Series Analysis: Forecasting and Control* (Holden-Day, San Francisco, CA, revised ed., 1976).

[7] G.B. Dantzig and P.W. Glynn, "Parallel processors for planning under uncertainty," *Annals of Operations Research* 22 (1990) 1–21.

[8] G.B. Dantzig and M. Madansky, "On the solution of two-staged linear programs under uncertainty," in: J. Neyman, ed., *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability I* (University of California Press, Berkeley, CA, 1961) pp. 165–176.

[9] N.C.P. Edirisinghe and W.T. Ziemba, "Tight bounds for stochastic convex programs," *Operations Research* 40 (1992) 660–677.

[10] Y. Ermoliev, "Stochastic quasigradient methods," in: Y. Ermoliev and R.J.-B. Wets, eds., *Numerical Techniques for Stochastic Optimization* (Springer, Berlin, 1988).

[11] K. Frauendorfer, *Stochastic Two-Stage Programming*, Lecture Notes in Economics and Mathematical Systems, Vol. 392 (Springer, Berlin, 1992).

[12] K. Frauendorfer, "Solving SLP recourse problems with arbitrary multivariate distributions – The dependent case," *Mathematics of Operations Research* 13 (1988) 377–394.

[13] H.I. Gassmann, "MSLiP: A computer code for the multistage stochastic linear programming problem," *Mathematical Programming* 47 (1990) 407–423.

[14] J.L. Higle and S. Sen, "Stochastic decomposition: An algorithm for two-stage linear programs with recourse," *Mathematics of Operations Research* 16 (1991) 650–669.

[15] G. Infanger, *Planning under Uncertainty: Solving Large-scale Stochastic Linear Programs* (The Scientific Press Series, Boyd & Fraser, New York, 1994).

[16] G. Infanger, "Monte Carlo (importance) sampling within a Benders decomposition algorithm for stochastic linear programs," *Annals of Operations Research* 39 (1992) 69–95.

[17] P. Kall, A. Ruszczyński, and K. Frauendorfer, "Approximation techniques in stochastic programming," in: Y. Ermoliev and R.J.-B. Wets, eds., *Numerical Techniques for Stochastic Optimization* (Springer, Berlin, 1988).

[18] D.P. Morton, "An enhanced decomposition algorithm for multistage stochastic hydroelectric scheduling," *Annals of Operations Research* 64 (1996) 211–235.

[19] M.V.F. Pereira and L.M.V.G. Pinto, "Multi-stage stochastic optimization applied to energy planning," *Mathematical Programming* 52 (1991) 359–375.

[20] M.V.F. Pereira and L.M.V.G. Pinto, "Stochastic optimization of a multi-reservoir hydroelectric System – A decomposition approach," *Water Resources Research* 21 (1989) 779–792.

[21] A. Ruszczyński, "A regularized decomposition method for minimizing a sum of polyhedral functions," *Mathematical Programming* 35 (1986) 309–333.

[22] G.C. Tiao and G.E.P. Box, "Modeling multiple time series with applications," *Journal of the American Statistical Association* 76 (1981) 802–816.

[23] R.S. Tsay and G.C. Tiao, "Consistent estimates of autoregressive parameters and extended sample autocorrelation function for stationary and nonstationary ARMA models," *Journal of the American Statistical Association* 79 (1984) 84–96.

[24] R.M. Van Slyke and R.J.-B. Wets, "L-shaped linear programs with applications to optimal control and stochastic programming," *SIAM Journal on Applied Mathematics* 17 (1969) 638–663.

[25] R.J.-B. Wets, "Large scale linear programming techniques," in: Y. Ermoliev and R.J.-B. Wets, eds., *Numerical Techniques for Stochastic Optimization* (Springer, Berlin, 1988).

[26] R.J.-B. Wets, "Stochastic programs with fixed recourse: The equivalent deterministic program," *SIAM Review* 16 (1974) 309–339.

[27] R.J. Wittrock, "Advances in a nested decomposition algorithm for solving staircase linear programs," Technical Report SOL 83-2, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA (1983).