Decomposition Algorithms with Parametric Gomory Cuts for Two-Stage Stochastic Integer Programs

Dinakar Gade, Simge Küçükyavuz, Suvrajeet Sen
Integrated Systems Engineering
210 Baker Systems, 1971 Neil Avenue
The Ohio State University, Columbus OH 43210
{gade.6,kucukyavuz.2,sen.22}@osu.edu

August 15, 2012

Abstract

We consider a class of two-stage stochastic integer programs with binary variables in the first stage and general integer variables in the second stage. We develop decomposition algorithms akin to the L-shaped or Benders' methods by utilizing Gomory cuts to obtain iteratively tighter approximations of the second-stage integer programs. We show that the proposed methodology is flexible in that it allows several modes of implementation, all of which lead to finitely convergent algorithms. We illustrate our algorithms using examples from the literature. We report computational results using the stochastic server location problem instances which suggest that our decomposition-based approach scales better with increases in the number of scenarios than a state-of-the art solver which was used to solve the deterministic equivalent formulation.

Keywords: Two-stage stochastic integer programs, Gomory cuts, L-Shaped method, Benders' decomposition, Lexicographic dual simplex, finite convergence

1 Introduction

The Gomory fractional cutting plane algorithm [16] is one of the earliest cutting plane algorithms developed to solve pure integer programs. A few years after the introduction of the fractional cuts, Gomory introduced mixed integer cuts [17] which have been successfully incorporated into branch-and-bound methods to solve mixed integer programs [4, 8]. Gomory cuts and their extensions have also been a subject of numerous theoretical studies in the mixed integer programming area [14, 18, 10]. Similarly, the Benders' decomposition [5] and L-shaped methods [38] are amongst the earliest and most successful classes of algorithms for solving stochastic linear programs and stochastic integer programs with continuous variables in the second stage. Recent studies [36, 31] extend these algorithms to accommodate binary variables in the second stage. However, these methods do not accommodate models that require general integer variables in the

This work is supported in part by NSF-CMMI Grant 1100383.

second stage. For such instances, the introduction of Gomory cuts would provide significantly stronger approximations as has been demonstrated for deterministic MILP instances. Despite the fact that the crux of the methodology (i.e. Gomory cuts, Benders/L-shaped method) has been in existence for over four decades, the only study that has attempted to integrate these tools within one algorithm for stochastic integer programs (SIP) is the work of Carøe and Tind [12]. However, their framework creates first-stage approximations that are very challenging to optimize. In this paper, we provide a first bridge by devising a finite L-shaped decomposition algorithm that is driven by parametric Gomory cuts in the second stage. These Gomory cuts will be parametrized by the first-stage decisions, thus allowing us to adopt a decomposition framework. This algorithm can be used to solve two-stage stochastic integer programs with binary first-stage and pure integer second-stage variables. We show that the steps involved in the algorithm are not only computationally attractive, but also flexible in that they allow several implementations depending on the requirements of the specific instance under consideration.

Let $\tilde{\omega}$ be a random vector defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ and let $\mathbb{E}[\cdot]$ denote the usual mathematical expectation operator taken with respect to $\tilde{\omega}$. A standard mathematical formulation for two-stage SIPs is given in (1)-(7).

$$\min c^\top x + \mathbb{E}\left[f(x, \tilde{\omega})\right] \tag{1}$$
$$Ax \geq b \tag{2}$$
$$x \in \mathcal{X} \tag{3}$$

where for a particular realization ω of $\tilde{\omega}$, $f(x, \omega)$ is defined as

$$f(x, \omega) = \min y_0 \tag{4}$$
$$y_0 - g(\omega)^\top y = 0 \tag{5}$$
$$W(\omega) y = r(\omega) - T(\omega) x \tag{6}$$
$$y \in \mathcal{Y}. \tag{7}$$

The sets $\mathcal{X} \subseteq \mathbb{R}^{n_1}_+$ and $\mathcal{Y} \subseteq \mathbb{R} \times \mathbb{R}^{n_2}_+$ impose discrete, binary or continuous restrictions on the variables. Here, $c \in \mathbb{Q}^{n_1}, A \in \mathbb{Q}^{m_1 \times n_1}$, $b \in \mathbb{Q}^{m_1}$, $g(\omega) \in \mathbb{Q}^{n_2}$, $W(\omega) \in \mathbb{Q}^{m_2(\omega) \times n_2}$, $T(\omega) \in \mathbb{Q}^{m_2(\omega) \times n_1}$ and $r(\omega) \in \mathbb{Q}^{m_2(\omega)}$.

Two-stage stochastic programs fall under the large umbrella of optimization problems under uncertainty. In two-stage stochastic programs with recourse, the decision maker must choose a vector of current decisions x at a cost $c^\top x$ under the constraints (2)-(3) before the realization of the random vector $\tilde{\omega}$. Upon the realization of a particular ω of $\tilde{\omega}$ referred to as a *scenario*, the decision maker chooses a vector of decisions y in response to the realization ω. The second-stage decisions incur a cost $g(\omega)^\top y$ under the constraints (6)-(7). The second-stage matrix $W(\omega)$ is referred to as the *recourse matrix*. The decisions made in the first stage impact the second-stage constraints through the *technology matrix* $T(\omega)$. We let $X := \{x : (2) - (3)\}$ denote the set of first-stage solutions and $Y(x, \omega) := \{y : (5) - (7)\}$ denote the set of second-stage feasible solutions for a given $x \in X$ and ω. Typically, the first stage corresponds to strategic or capital investment decisions and the second stage corresponds to operational decisions made in response to uncertain events. The most commonly used objective function of two-stage recourse models is to minimize the first-stage costs and the expected future costs of the second stage. In this

paper, we assume that $\mathcal{X} = \{x \in \mathbb{B}^{n_1}\}$ and $\mathcal{Y} = \{y \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}\}$.

Stochastic programs of the type (1)-(7) are amongst the most challenging classes of optimization problems. In addition to the usual "curse of dimensionality" due to the multi-dimensional integral in the expectation calculation of (1), these models have to contend with difficulties arising from integer recourse problems of the second stage. In order to mitigate some of the difficulties, we make the following assumptions:

(A1) $\tilde{\omega}$ has a finite support on Ω.

(A2) All the data elements in the second stage $g(\omega), W(\omega), r(\omega), T(\omega)$ are integral.

(A3) The set X is non-empty.

(A4) $Y(x, \omega) \neq \emptyset$ for all $(x, \omega) \in X \times \Omega$.

(A5) $|E[f(x, \tilde{\omega})]| < +\infty$.

Because of assumption (A1), the expectation in (1) can be replaced by a weighted sum and one can write the so-called deterministic equivalent formulation (DEF), which may be viewed as a large-scale integer program:

$$\min c^\top x + \sum_{\omega \in \Omega} p_\omega y_0(\omega) \tag{8}$$

$$Ax \geq b \tag{9}$$

$$y_0(\omega) - g(\omega)^\top y(\omega) = 0, \qquad \omega \in \Omega \tag{10}$$

$$T(\omega)x + W(\omega)y(\omega) = r(\omega), \qquad \omega \in \Omega \tag{11}$$

$$x \in \mathcal{X}, y(\omega) \in \mathcal{Y}, \qquad \omega \in \Omega, \tag{12}$$

where $\mathcal{P}(\tilde{\omega} = \omega) =: p_\omega$. Note that in the decomposed formulation (1)-(7), the dependence of y-variables on ω is implicit. On the other hand, for the DEF, we express the dependence of y on the scenario explicitly using ω as an argument since we need to distinguish between the second-stage solutions for different ω. Throughout the paper, the arguments for y or lack thereof, will be clear from the context. Assumption (A2) is without loss of generality since the original rational data can be scaled by a suitable constant to obtain integer coefficients with the same representation. Assumption (A3) ensures the existence of at least one feasible first-stage solution within a compact subset of the $[0, 1]$ cube in $\mathbb{R}^{n_1}$. Assumption (A4) can be enforced for most practical problems by introducing artificial variables and penalizing the artificial variables in the second-stage objective. Note that Assumption (A4) implies $\mathbb{E}[f(x, \tilde{\omega})] < \infty$. Stochastic programs that satisfy this assumption are said to possess the *relatively complete recourse* property. We further assume in (A5) that the SIP has a finite optimum, i.e., $\mathbb{E}[f(x, \tilde{\omega})] > -\infty$ to ensure that the problem is well defined.

The DEF can be input to an off-the-shelf mixed integer programming (MIP) solver. However, as is often the case in stochastic programming, one encounters a large number of scenarios which imposes large memory and computational burden on the solver. One approach to deal with this difficulty is to work with smaller problems by decomposing the DEF into a first-stage problem and a collection of second-stage scenario problems. This is possible since for a given $x \in X$, the scenario problems can be solved independently. Benders' decomposition or the L-Shaped method exploit this decomposability and reformulate the problem in the space of the (x, η)-

variables as follows:

$$\min_{x \in X} \{c^\top x + \eta : \eta \geq \mathbb{E}[f(x, \tilde{\omega})]\}. \tag{13}$$

In an L-shaped algorithm, one constructs sequentially tighter approximations of $\mathbb{E}[f(x, \tilde{\omega})]$. When the second stage is a linear program, $f(\cdot, \omega)$ is a convex piecewise linear function and since $\mathbb{E}[f(\cdot, \tilde{\omega})]$ is a convex combination of such functions, it continues to satisfy convexity. Thus, it is possible to construct approximations of $\mathbb{E}[f(x, \tilde{\omega})]$ by using the so-called 'optimality cuts', which are affine. When the second stage is an integer program, $f(\cdot, \omega)$ is only lower semicontinuous and in general non-convex (see Blair and Jeroslow [9], Schultz [28]). Thus, approximating $\mathbb{E}[f(x, \tilde{\omega})]$ becomes a challenge when the second-stage variables are integers.

For SIP problems with pure integer recourse decisions, this paper addresses the following questions that have remained unresolved to date:

1. Can the expected recourse function $\mathbb{E}[f(x, \tilde{\omega})]$ be approximated by piecewise linear functions of the first-stage variables? Such approximations allow the master program to possess the same structure as in standard Benders' decomposition, which assumes linear programs in the second-stage.
2. Is it possible to avoid solving the scenario subproblems to IP optimality in each iteration?
3. Is it possible to design a finitely convergent decomposition algorithm using only cutting planes to approximate the feasible set of the second-stage?

Our paper resolves these questions with affirmative answers for problems with binary variables in the first stage and general integer variables in the second stage. As the paper unfolds, we will discuss the implications of these answers in greater technical depth.

The remainder of the paper is organized as follows: In Section 2, we discuss relevant literature. In Section 3, we develop a decomposition algorithm driven by parametric Gomory cuts for the second stage. Section 4 discusses alternative implementations of our main algorithm. Section 5 deals with the convergence of the algorithms presented in Sections 3 and 4. In Section 6, we illustrate our algorithms using examples from the literature. We report our preliminary computational experience in Section 7.

2 Literature

For a general introduction to stochastic programming, we refer the reader to Birge and Louveaux [7] and for surveys on stochastic mixed integer programming (SMIP), to Louveaux and Schultz [21] and Sen [30]. To aid the discussion of literature, we use a classification scheme to represent the various classes of two-stage stochastic mixed integer programs. This scheme is based on the variable restrictions represented by $\mathcal{X}$ and $\mathcal{Y}$. We use the sets B, C, D to denote the stages of the stochastic program that have binary, continuous and discrete variables, respectively. For example, the class of problems considered in this paper have $B = \{1, 2\}$, $C = \emptyset$, $D = \{2\}$, i.e., binary variables may appear in both stages, no continuous variables may appear in either stage and general integer variables may appear in the second stage.

Laporte and Louveaux's [20] L-shaped algorithm was one of the earliest decomposition methods for SMIPs. Their algorithm solves problems with $B = \{1, 2\}$, $C = \{2\}$ and $D = \{2\}$. They solve second-stage mixed integer programs to optimality at each iteration and construct first-stage cuts using the objective function values of the second-stage problems. The potential

drawback of their approach is the lack of scalability since each iteration requires the solution of multiple NP-hard mixed integer programs in the second stage.

Carøe and Tind [12] develop a conceptual generalization of the L-shaped method for problems with $B = \{1, 2\}, C = \{1\}, D = \{1, 2\}$ using integer programming duality. In their framework, one solves the scenario subproblems for a given $x \in X$ to optimality using a pure cutting plane algorithm with Gomory cuts. Then, one constructs the optimal subadditive dual function for each scenario, which is of the form $\mathcal{F}_\omega(d) = u^\top d + \sum_{i=1}^{t} (v^i)^\top F_i(d)$, where t denotes the number of Gomory cuts needed to solve the scenario subproblems and $u, v^i, i = 1, \ldots, t$ are rational vectors and $F_i, i = 1, \ldots, t$ are a series of nested Chvátal functions [9] (a Chvátal function is of the form $F_i(d) = q\lceil H_i \lceil H_{i-1} \cdots \lceil H_2 \lceil H_1 d \rceil \rceil \cdots \rceil \rceil$,, where q, H_j are rational matrices). The optimality cuts developed in [12] are of the form $\eta \geq \sum_{\omega \in \Omega} p_\omega \mathcal{F}_\omega(r(\omega) - T(\omega)x)$. Although the functions $\mathcal{F}_\omega$ can be represented as linear functions in integer variables, it is clear that this representation involves the introduction of a large number of integer variables for each scenario at every iteration. In addition, integer programs for each scenario need to be solved to optimality with a pure cutting plane algorithm at every iteration, thereby making the approach impractical for computational purposes.

To avoid the solution of multiple integer programs to completion in an iteration, cutting plane methods that partially approximate the second-stage problems within the L-shaped method have been proposed. In the literature, such methods for two-stage problems have been almost exclusively restricted to disjunctive cut-generation schemes. Carøe and Tind [11] use lift-and-project cutting planes [3] to solve problems with $B = \{1, 2\}, C = \{1, 2\}, D = \emptyset$. They generate cutting planes to separate points $(x, y(\omega))$ that do not satisfy the mixed integer requirement in the linear relaxation of the deterministic equivalent formulation. Sen and Higle [31] consider problems with $B = \{1, 2\}, C = \{2\}, D = \emptyset$ and in addition, assume that the recourse matrix is fixed, i.e, $W(\omega) = W$ for all $\omega \in \Omega$. They develop disjunctive cuts for the second stage that maintain the fixed-recourse property, i.e., they derive cuts that can be made valid across all scenarios by calculating an appropriate right-hand-side function. In addition, they sequentially approximate the value function using linear cutting planes in the first stage while avoiding the computation of the linear relaxation of the deterministic equivalent formulation before cut generation. Sherali and Fraticelli [36] develop a similar method in the framework of the reformulation linearization technique (RLT) (c.f. [35]). Sen and Sherali [34] develop an extension to the algorithm of Sen and Higle for problems with $B = \{1, 2\}, C = \{2\}, D = \{2\}$ by using disjunctive programming techniques and a partial branch-and-bound tree in the second stage. Ntaimo and Sen [26, 27], Yuan and Sen [39] report computational studies of the algorithms of [31] and [34]. Ntaimo [25] extends the algorithm of Sen and Higle to problems with random recourse matrix and a fixed technology matrix ($T(\omega) = T$ for all $\omega \in \Omega$) and ensures that the cuts for the second stage have common coefficients. Sherali and Zhu [37] develop an asymptotically convergent decomposition-based branch-and-bound algorithm that uses RLT cuts for problems with $B = \{1, 2\}, C = \{1, 2\}, D = \emptyset$.

Other approaches to solve stochastic integer programs include the algorithm by Schultz et al. [29] who develop an enumeration algorithm based on polynomial ideal theory (Gröbner bases) for problems with $B = \{2\}, C = \{1\}, D = \{2\}$. Unfortunately, Gröbner bases are notoriously difficult to compute (see Mayr [22]). Ahmed et al. [2] develop a finite branch-and-bound algorithm for problems with $B = \{1, 2\}, C = \{1\}, D = \{2\}$ and with a fixed technology

matrix. Kong et al. [19] study problems with $B = \{1, 2\}, C = \emptyset, D = \{1, 2\}$ and in addition assume that the technology, recourse matrices and the second-stage cost functions are fixed. Unlike the algorithms discussed in this paragraph, we make no assumptions on the elements of input data that are affected by the random variables.

Our overall approach is similar to the one taken in [31] in that we iteratively approximate the second-stage problems using cutting planes. Sen and Higle [31] use disjunctive cuts whereas we use Gomory cuts within our decomposition algorithm. In Section 6 we compare and contrast the features of their algorithm with ours using an example.

3 Parametric Gomory Cuts for Decomposition

In this section, we develop a decomposition algorithm driven by Gomory cutting planes to solve stochastic integer programs. Our approach does not attempt to solve multiple integer scenario sub-problems in a single iteration but instead, solves only linear programs in the second stage and generates violated cuts. In order to develop such an approach, we derive valid inequalities of the form $\pi(\omega)^\top y(\omega) \geq \pi_0(x, \omega)$. Because the right hand side π_0 is a function of x, we will refer to them as parametric Gomory cuts. Toward this end let $S(x, \omega) := \text{clconv}(\{(x, y_0(\omega), y(\omega)) \in \mathbb{B}^{n_1} \times \mathbb{Z} \times \mathbb{Z}_+^{n_2} : y_0(\omega) - g(\omega)^\top y = 0, T(\omega)x + W(\omega)y(\omega) = r(\omega)\})$. Let $T^c(\omega), W^c(\omega), r^c(\omega)$ be appropriately dimensioned rational matrices so that $S(x, \omega) = \{(x, y_0(\omega), y(\omega)) \in \mathbb{R}_+^{n_1} \times \mathbb{R} \times \mathbb{R}_+^{n_2} : T^c(\omega)x + W^c(\omega)(y_0(\omega), y(\omega)) \geq r^c(\omega)\}$. Let $\bar{x}$ denote an extreme point of $\text{conv}(X)$. Since the restriction $S(x, \omega) \cap \{(x, y_0(\omega), y(\omega)) : x = \bar{x}\}$ yields a face of $S(x, \omega)$, the extreme points of the set $Y^c(\bar{x}, \omega) := \{(y_0(\omega), y(\omega)) \in \mathbb{R} \times \mathbb{R}_+^{n_2} : W^c(\omega)(y_0(\omega), y(\omega)) \geq r^c(\omega) - T^c(\omega)\bar{x}\}$ are integral. We exploit this facial or extreme-point property of binary vectors to derive valid inequalities $\pi(\omega)^\top y(\omega) \geq \pi_0(x, \omega)$ with $\pi_0(\cdot, \omega)$ affine. Affine right hand side functions are clearly desirable since they are easy to evaluate and they result in first-stage approximations that are piecewise linear and convex.

In order to derive valid inequalities within a decomposition algorithm, suppose that a binary first-stage vector x' is given. Without loss of generality, we can derive the cut coefficients in a translated space for which the point x' is the origin. Once the cut coefficients are obtained in the translated space, it will be straightforward to transform these coefficients to be valid in the original space. Because x is binary, such translation is equivalent to replacing every non-zero element x_j by its complement, $1 - x_j$. Thus, without loss of generality we will derive the parametric Gomory cut for a generalized origin $\bar{x} = 0$. Let $(\bar{x}, \omega) \in X \times \Omega$ be given and let $B(\bar{x}, \omega)$ denote an optimal basis of the LP relaxation of the second stage problem. With this basis, we associate index sets $\mathcal{B}(\bar{x}, \omega)$ and $\mathcal{N}(\bar{x}, \omega)$ which correspond to basic and nonbasic variables respectively, and denote $N(\bar{x}, \omega)$ as the submatrix corresponding to non-basic columns. For ease of exposition, we drop the dependence on $(\bar{x}, \omega)$ and $y, B, N, \mathcal{B}, \mathcal{N}$ will be in reference to $(\bar{x}, \omega)$ while T, W, r will be in reference to ω. With a slight abuse of notation, the derivation here uses x to denote points in the translated space. Multiplying (5)-(6) by B^{-1} we obtain

$$y_{\mathcal{B}} + B^{-1}Ny_{\mathcal{N}} = B^{-1}\begin{pmatrix} 0 \\ r - Tx \end{pmatrix} =: h(x). \tag{14}$$

Let the components of $B^{-1}N$ be denoted by $\bar{w}_{ij}$. Also let

$$B^{-1}\begin{pmatrix} 0 \\ r \end{pmatrix} =: \rho,$$

$$B^{-1}\begin{pmatrix} \mathbf{0} \\ T \end{pmatrix} =: \Gamma$$

and let the components of Γ be denoted by γ_{ij}. Let $\mathcal{B}_i$ denote the ith basic variable. We pick a candidate source row corresponding to $y_{\mathcal{B}_i}$ for which $h_i(\bar{x}) \notin \mathbb{Z}$ for generating the Gomory cut. The corresponding row in (14) may be written by rearranging the terms as

$$y_{\mathcal{B}_i} + \sum_{j \in \mathcal{N}} \bar{w}_{ij} y_j + \sum_{j=1}^{n_1} \gamma_{ij} x_j = \rho_i. \tag{15}$$

Because we are deriving cuts in the translated space at $\bar{x} = 0$, we can treat these variables as 'non-basic' in the current solution. Letting $\phi(\beta) := \lceil \beta \rceil - \beta$, a Gomory fractional cut in the (x, y)-space can be written as

$$\sum_{j \in \mathcal{N}} \phi(\bar{w}_{ij}) y_j + \sum_{j=1}^{n_1} \phi(\gamma_{ij}) x_j \geq \phi(\rho_i).$$

or equivalently in the space of y-variables as a function of x as

$$\sum_{j \in \mathcal{N}} \phi(\bar{w}_{ij}) y_j \geq \phi(\rho_i) - \sum_{j=1}^{n_1} \phi(\gamma_{ij}) x_j. \tag{16}$$

Reintroducing the dependence on (x, ω), we note that inequality (16) has the desired form $\pi(\omega)^\top y(\omega) \geq \pi_0(x, \omega)$ and moreover it has an attractive property that the right hand side function π_0 is affine in x. When $x = \bar{x}$, inequality (16) is the usual Gomory fractional cut valid for $Y(\bar{x}, \omega)$. Moreover, inequality (16) is valid for $Y(x, \omega)$ for all $x \in X$. Thus, in our decomposition algorithm that iteratively tightens the approximations of $Y(x, \omega)$, at the beginning of iteration k, we can write a scenario approximation problem $\mathrm{SP}^{k-1}(x, \omega)$ as

$$f_\ell^{k-1}(x, \omega) = \min y_0 \tag{17}$$

$$y_0 - g(\omega)^\top y = 0 \tag{18}$$

$$W^{k-1}(\omega) y \geq r^{k-1}(\omega) - T^{k-1}(\omega) x \tag{19}$$

$$y \in \mathbb{R}^{n_2}_+, y_0 \in \mathbb{R}, \tag{20}$$

where the constraints $W^{k-1}(\omega)y \geq r^{k-1}(\omega) - T^{k-1}(\omega)x$ include the original constraints $W(\omega)y = r(\omega) - T(\omega)x$ and in addition, include parametric Gomory fractional cuts of the form (16) that may have been generated during iterations $1, \ldots, k-1$. For the first-stage solution x^k, we solve the approximation problem $\mathrm{SP}^{k-1}(x^k, \omega)$. If the solution to this problem is fractional, we add a parametric Gomory cut (16) and call this subproblem $\mathrm{SP}^k(x, \omega)$. If the solution to $\mathrm{SP}^{k-1}(x^k, \omega)$ is integral, we update just the iteration index and $\mathrm{SP}^k(x^k, \omega)$ is the same as $\mathrm{SP}^{k-1}(x^k, \omega)$. We solve $\mathrm{SP}^k(x^k, \omega)$ and let $u^k(\omega)$ denote the corresponding optimal dual multipliers. An affine

under-estimator of $\mathbb{E}[f(x,\tilde{\omega})]$, or an *optimality cut* can be generated as follows:

$$\eta - \sum_{\omega\in\Omega} p_\omega (u^k(\omega))^\top (r^k(\omega) - T^k(\omega)x) \geq 0. \tag{21}$$

Thus, a lower bounding approximation for (13) at iteration k, denoted by MP^k, can be written as follows:

$$\min c^\top x + \eta \tag{22}$$

$$Ax \geq b \tag{23}$$

$$\eta - \sum_{\omega\in\Omega} p_\omega (u^t(\omega))^\top (r^t(\omega) - T^t(\omega)x) \geq 0, \qquad t = 1,\ldots,k \tag{24}$$

$$x \in \mathbb{B}^{n_1}, \eta \in \mathbb{R}. \tag{25}$$

Remark 1. Note that unlike standard Benders' decomposition, the second-stage LP relaxation in this algorithm is updated from one iteration to the next, and moreover, these approximations include affine functions of first-stage variables. These lifted cuts are then used to obtain affine approximations of second-stage (IP) value functions. In this sense, for the class of problems studied here, the lifting allows us to overcome complicated sub-additive functions as proposed by Carøe and Tind [12].

Remark 2. The reader might recall that for mixed binary second-stage recourse problems, Sen and Higle [31] present a similar algorithm using disjunctive cuts. However, those cuts give rise to piecewise linear concave functions for $\pi_0(x,\omega)$ and as a result require further convexification for use within a Benders' type method.

Before stating the decomposition algorithm, we introduce the following additional notation. An iteration of the algorithm is denoted by k and LB^k, UB^k denote the lower and upper bounds on the optimal objective function value of SIP at iteration k. A decomposition algorithm with parametric Gomory cuts in the second stage is given in Algorithm 1.

The algorithm executes as follows: We initialize the master problem with no optimality cuts, and subproblems using their linear relaxations. We start with some $x^1 \in X$. In iteration k, using the first-stage solution x^k, we solve the scenario approximation problems $\text{SP}^{k-1}(x^k,\omega)$ for each $\omega \in \Omega$. If the second-stage solutions for each scenario are integral, it implies that we have found a feasible solution and we update the best upper bound UB^{k+1}. On the other hand, for each scenario ω for which the solution $y(\omega)$ is non-integral, we generate a Gomory cut (16) from the smallest index row whose right hand side is fractional. With the cut added to the current approximation, we obtain the matrices $W^k(\omega), T^k(\omega), r^k(\omega)$. We solve the resulting problem $\text{SP}^k(x^k,\omega)$ using the lexicographic dual simplex method and update the best upper bound if integer solutions are obtained for $\text{SP}^k(x^k,\omega)$ for all $\omega \in \Omega$. If no cut is generated for a scenario ω, the matrices $W^k(\omega), T^k(\omega), r^k(\omega)$ are set to $W^{k-1}(\omega), T^{k-1}(\omega), r^{k-1}(\omega)$, respectively. We then construct an optimality cut (21) using the optimal dual multipliers of the $\text{SP}^k(x^k,\omega)$. We solve the updated master problem MP^k using branch-and-bound, to obtain a first-stage solution x^{k+1} and a lower bound LB^{k+1} for the SIP and increment the iteration index. We repeat this process until the stopping condition, $UB^k = LB^k$ is met. In a computer implementation, alternate stopping criteria can be used, for example, for a pre-specified tolerance ε, we can terminate the algorithm when $UB^k - LB^k < \varepsilon$.

Algorithm 1 Decomposition with Parametric Gomory Cuts for 2-Stage SIP

Initialization: $k \leftarrow 1$, $LB^1 \leftarrow -\infty, UB^1 \leftarrow +\infty$, $W^0(\omega) \leftarrow W(\omega), T^0(\omega) \leftarrow T(\omega), r^0(\omega) \leftarrow r(\omega)$. Let $x^1 \in X$.
while $LB^k < UB^k$ **do**
  Solve $\text{SP}^{k-1}(x^k, \omega)$ for all $\omega \in \Omega$ using the lexicographic dual simplex method.
  if the solution to $\text{SP}^{k-1}(x^k, \omega)$, $y(\omega) \in \mathbb{Z} \times \mathbb{Z}^{n_2}$ for all $\omega \in \Omega$ **then**
    Put $W^k(\omega) \leftarrow W^{k-1}(\omega), T^k(\omega) \leftarrow T^{k-1}(\omega), r^k(\omega) \leftarrow r^{k-1}(\omega)$.
    Update $UB^{k+1} \leftarrow \min(UB^k, c^\top x^k + \sum_{\omega \in \Omega} p_\omega f_\ell^k(x^k, \omega))$.
  else
    For all $\omega \in \Omega$ s.t. $I_k(\omega) := \{i \in \{0, \ldots, n_2\} : y_i(\omega) \notin \mathbb{Z}\} \neq \emptyset$, choose the source row corresponding to the smallest index $i \in I_k(\omega)$ and generate Gomory fractional cut (16). Add the cut to $\text{SP}^{k-1}(x, \omega)$ to get $W^k(\omega), T^k(\omega), r^k(\omega)$. Put $W^k(\omega) \leftarrow W^{k-1}(\omega), T^k(\omega) \leftarrow T^{k-1}(\omega), r^k(\omega) \leftarrow r^{k-1}(\omega)$ for all ω with $I_k(\omega) = \emptyset$.
    Solve $\text{SP}^k(x^k, \omega)$ for all ω with $I_k(\omega) \neq \emptyset$ using the lexicographic dual simplex method.
    If the solution to $\text{SP}^k(x^k, \omega)$, $y(\omega) \in \mathbb{Z} \times \mathbb{Z}^{n_2}$, update $UB^{k+1} \leftarrow \min(UB^k, c^\top x^k + \sum_{\omega \in \Omega} p_\omega f_\ell^k(x^k, \omega))$.
  end if
  Add optimality cut (21) to MP^k and solve MP^k.
  Set x^{k+1}, LB^{k+1} to the optimal first-stage solution and objective of MP^k, respectively.
  $k \leftarrow k + 1$.
end while
return x^k, $c^\top x^k + \sum_{\omega \in \Omega} p_\omega f_\ell^k(x^k, \omega)$.

We conclude this section by noting that Algorithm 1 is also applicable when continuous and general integer variables are present in the first stage, but their coefficients in the technology matrix are zeros.

4 Alternative Implementations

In this section, we develop the following alternative implementations of Algorithm 1.

1. Multiple cuts in the first stage (multicut implementation)
2. A round of cuts in the second stage
3. Parametric Gomory mixed integer cuts in the second stage
4. Fixed recourse matrix
5. Partial branch-and-cut for binary second stage.

The first two implementations are relatively straightforward, while the last three require greater discussion and are described in subsections 4.1-4.3.

The *multicut* version [6] of this algorithm approximates the value function $f(x, \omega)$ for each $\omega \in \Omega$ by adding $|\Omega|$ optimality cuts in the first stage. Disaggregated cuts in the first stage may reduce the number of main iterations performed since more information is passed on to the first stage. A potential drawback is that if there are many scenarios, the large number of first-stage cuts may slow down the solution of the master problems.

An alternative implementation of our algorithm incorporates multiple violated Gomory cuts in the second stage per iteration, potentially as many as the number of fractional variables for each scenario. This approach of adding cutting planes from all rows corresponding to fractional

variables is referred to as adding a *round of cuts*. In the mixed integer programming literature, a number of authors demonstrate the computational superiority of adding a round of cuts over adding a single cut in each iteration (see for example [4]).

4.1 Parametric Gomory Mixed Integer Cuts

We show that Gomory mixed integer (GMI) cuts can also be used instead of fractional cuts in Algorithm 1. While GMI cuts were developed for mixed integer problems, they are also applicable for pure integer problems and give cuts that are at least as strong as the fractional cuts. We now derive GMI cuts in the context of our algorithm and continue with the assumption that the second-stage variables are pure-integer. Re-writing the source row for cut generation from Section 3, we have

$$y_{\mathcal{B}_i} + \sum_{j \in \mathcal{N}} \bar{w}_{ij} y_j + \sum_{j=1}^{n_1} \gamma_{ij} x_j = \rho_i.$$

A GMI cut from the above equation can be derived in the (x, y)-space as

$$\sum_{j \in \mathcal{N}} \min\left(\phi(\bar{w}_{ij}), \frac{\phi(\rho_i)(1-\phi(\bar{w}_{ij}))}{1-\phi(\rho_i)}\right) y_j + \sum_{j=1}^{n_1} \min\left(\phi(\gamma_{ij}), \frac{\phi(\rho_i)(1-\phi(\gamma_{ij}))}{1-\phi(\rho_i)}\right) x_j \geq \phi(\rho_i)$$

or equivalently in the space of y-variables as a function of x as

$$\sum_{j \in \mathcal{N}} \min\left(\phi(\bar{w}_{ij}), \frac{\phi(\rho_i)(1-\phi(\bar{w}_{ij}))}{1-\phi(\rho_i)}\right) y_j \geq \phi(\rho_i) - \sum_{j=1}^{n_1} \min\left(\phi(\gamma_{ij}), \frac{\phi(\rho_i)(1-\phi(\gamma_{ij}))}{1-\phi(\rho_i)}\right) x_j. \quad (26)$$

Similar to the case of fractional cuts, for a fixed $\bar{x} \in X$, inequality (26) yields the usual GMI cut for $Y(\bar{x}, \omega)$. Note that we express the GMI cuts in terms of $\phi(\cdot)$ rather than the commonly used $\delta(\beta) := \beta - \lfloor \beta \rfloor$ to conform to the direction of rounding of the fractional cut (16). This is done to ensure convergence, which is discussed in Section 5.

4.2 Fixed Recourse Matrix

For problems with a fixed recourse matrix, it may be desirable to maintain a fixed W^k matrix across all scenarios as the second-stage approximations are being constructed. This is one of the main considerations in the paper by Sen and Higle [31]. We show that it is possible to maintain a fixed recourse matrix when using Gomory cuts within a decomposition algorithm. Suppose that we start with a fixed recourse matrix $W(\omega) = W$ for all $\omega \in \Omega$, and in iteration k a Gomory fractional cut is generated for scenario $\bar{\omega} \in \Omega$. Suppose that $B(\bar{\omega})$ is the optimal basis for scenario problem $\bar{\omega}$ and row $i(\bar{\omega})$ is used for cut generation. Then, cut coefficients for the second-stage matrix that are common across scenarios can be obtained by multiplying the second-stage constraints for each $\omega \in \Omega$ by $B(\bar{\omega})^{-1}$ and deriving an inequality (16) from row $i(\bar{\omega})$. In addition to a fixed recourse matrix, if the technology matrix is also fixed, we obtain a series of approximations that maintains a fixed technology matrix. While common cut coefficients can be obtained using the method of Sen and Higle [31], a fixed technology matrix cannot be guaranteed during the construction of their approximations because the technology matrix updates involve the convexification of a reverse convex constraint.

4.3 Partial Branch-and-Cut for Binary Second Stage

In this subsection we assume that some or all of the second-stage variables are binary, i.e., $y_j \in \{0,1\}$ for $j \in J$ where $J \subseteq \{1,\dots,n_2\}$ and $J \neq \emptyset$. We show that strong approximations of the second-stage problems can be constructed using a partial branch-and-cut tree in which we branch on binary variables a few times. Tighter approximations may be of interest when the second-stage problems are difficult and have weak LP relaxation bounds. Our approach is somewhat similar to the algorithm in [34]. However, we do not solve a cut generating linear program to generate optimality cuts.

Let $(\bar{x},\omega) \in X \times \Omega$ be given. Without loss of generality, we will derive the parametric Gomory cuts in this setting for a generalized origin $\bar{x} = \mathbf{0}$. In the partial branch-and-cut approach, we solve the scenario problem $\text{SP}(\bar{x},\omega)$ and add a single Gomory cut or a round of Gomory cuts to the LP relaxation and re-solve the problem. We treat this step as generating cuts at the root node of a partial branch-and-cut tree. If the second-stage variables $y_j, j \in J$ are fractional, we branch on one of these fractional variables. Let σ denote a node of a partial branch-and-cut tree for the second-stage problems. Let $J_0^\sigma, J_1^\sigma \subseteq J$ denote the index set of the variables fixed at 0 and 1, respectively at node σ of the partial branch-and-cut tree. Further, we assume that the fixing of variables at the nodes is performed by adding the constraints $y_j \leq 0, j \in J_0^\sigma$ and $y_j \geq 1, j \in J_1^\sigma$ (see also Balas et al. [4] for a similar approach to lifting GMI cuts in a branch-and-cut scheme for deterministic MIPs). Since we can complement the binary second-stage variables $j \in J_1^\sigma$ as well, we can obtain additional cut(s) (16) using the basis for the nodal problem, which are globally valid not only for the scenario subproblem $\text{SP}(\bar{x},\omega)$, but also for $Y(x,\omega)$ for all $x \in X$. Therefore, we can potentially generate as many rounds of cuts as nodal problems in the partial branch-and-cut tree, all of which are globally valid for all $x \in X$. When the cuts generated at each node σ of the partial branch-and-cut tree are added to $\text{SP}(\bar{x},\omega)$ along with the root-node cuts, we get a bound that is at least as tight as the one obtained by adding a single round of cuts at the root node.

5 Finite Convergence

In this section, we prove the finiteness of Algorithm 1 and its variants. Throughout this section, we write the second-stage solutions as $y(x,\omega)$ to emphasize the convergence of the second-stage solutions for a particular $(x,\omega) \in X \times \Omega$. First, we state a few results needed for our convergence arguments.

Proposition 1 ([23]). *Let $y(x,\omega)$ denote an extreme point of $\text{conv}(Y(x,\omega))$ for some $(x,\omega) \in X \times \Omega$. Then there exists $M(x,\omega) < \infty$ such that $y_j(x,\omega) \leq M(x,\omega)$ with $M(x,\omega) \in \mathbb{Z}$ for $j = 1,\dots,n_2$.*

Since (x,ω) belong to a finite set $X \times \Omega$, there is no loss of generality in assuming a uniform upper bound $y_j(x,\omega) \leq M = \sup_{(x,\omega)}\{M(x,\omega)\}$ for all $j = 1,\dots,n_2, (x,\omega) \in X \times \Omega$. For the sake of completeness, we include the following definition.

Definition 1 (Lexicographic Order). Let $v^1, v^2 \in \mathbb{R}^n$. v^1 is said to be *lexicographically larger* than v^2 denoted as $v^1 \succ v^2$ if there exists a k such that $v^1_k > v^2_k$ and $v^1_j = v^2_j$ for all $j = 1,\dots,k-1$. We say that v^1 is lexicographically larger than or equal to v^2 denoted by $v^1 \succeq v^2$, if either $v^1 \succ v^2$ or $v^1 = v^2$.

Let $\beta^- := \min(\beta, 0)$ and $\beta^+ := \max(0, \beta)$. From Proposition 1, it follows that if $y(x,\omega) \in Y(x,\omega)$ for $(x,\omega) \in X \times \Omega$, then

$$\left(\sum_{j=1}^{n_2} g_j(\omega)^+ M, M, \ldots, M\right)^\top \succeq y(x,\omega) \succeq \left(\sum_{j=1}^{n_2} g_j(\omega)^- M, 0, \ldots, 0\right)^\top. \tag{27}$$

The proof of convergence of Gomory's fractional cutting plane algorithm relies on the lexicographic dual simplex method [24, 23]. In this version of the simplex method, one begins with and maintains simplex tableaus that are lexicographically dual feasible, i.e., tableaus that are dual feasible and have all the nonbasic columns lexicographically smaller than zero. This is accomplished by using a lexicographic pivoting rule. This approach ensures that the simplex method does not cycle, and furthermore, guarantees that the lexicographically smallest optimal solution to the linear program, if it exists, is found in finitely many steps.

The following is a key result that establishes a rounding property of the Gomory fractional cutting plane algorithm and is used to prove the finiteness of Gomory's fractional cutting plane algorithm.

Proposition 2 ([24]). *Let $(\bar{x},\omega) \in X \times \Omega$ be given. Let $y^t(\bar{x},\omega)$ denote the lexicographically smallest optimal solution to $\mathrm{SP}^t(\bar{x},\omega), t = k-1, k$, where the updated subproblem SP^k is formed by adding a Gomory cut (16) corresponding to the smallest index source row of the fractional solution $y^{k-1}(\bar{x},\omega)$. Let i_k denote the index of the variable used to form the Gomory cut, and define*

$$\alpha^k(\bar{x},\omega) := \left(y_0^{k-1}(\bar{x},\omega), y_1^{k-1}(\bar{x},\omega), \ldots, y_{i_k-1}^{k-1}(\bar{x},\omega), \lceil y_{i_k}^{k-1}(\bar{x},\omega)\rceil, 0, \ldots, 0\right)^\top. \tag{28}$$

Then,

$$y^k(\bar{x},\omega) \succeq \alpha^k(\bar{x},\omega) \succ y^{k-1}(\bar{x},\omega).$$

The following result shows that for a given $\bar{x} \in X$, we obtain a lexicographically increasing sequence of $\alpha^k(\bar{x},\omega)$'s.

Lemma 1. *Let $(\bar{x},\omega) \in X \times \Omega$ be given and let t, k denote two iterations such that $t > k$, and $x^k = x^t = \bar{x}$ and $y^j(\bar{x},\omega)$ are non-integral for $j = k, t$. Then $\alpha^t(\bar{x},\omega) \succ \alpha^k(\bar{x},\omega)$.*

Proof. It is sufficient to prove the result for the case in which t denotes the first iteration following k for which $x^t = x^k = \bar{x}$. For $x^k = \bar{x}$, we have

$$y^k(x^k,\omega) \succeq \alpha^k(x^k,\omega),$$

using Proposition 2. Since the inequalities added during iterations $k+1, \ldots, t-1$ are valid for $Y(x,\omega)$ for all $x \in X$ and $y^j(\bar{x},\omega)$ is the lexicographically smallest optimal solution for $\mathrm{SP}^j(\bar{x},\omega)$, $j = k, t-1$, we have for $x^t = \bar{x}$

$$y^{t-1}(x^t,\omega) \succeq y^k(x^k,\omega).$$

Furthermore, from (28) we have

$$\alpha^t(x^t,\omega) \succ y^{t-1}(x^t,\omega),$$

and the result follows. □

The next result shows that in the worst case, Algorithm 1 finds integral solutions to the second-stage problems for a given $(\bar{x}, \omega) \in X \times \Omega$. In the following, if no cuts are generated at iteration k for scenario ω, $y^{k-1}(x^k, \omega)$ and $y^k(x^k, \omega)$ will denote the same vector.

Lemma 2. *Let $(\bar{x}, \omega) \in X \times \Omega$ be given. Then there exists an integer $K(\bar{x}, \omega), 0 < K(\bar{x}, \omega) < \infty$ such that either the algorithm terminates in $k < K(\bar{x}, \omega)$ iterations or $y^k(\bar{x}, \omega) \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}$ and $f_\ell^k(\bar{x}, \omega) = f(\bar{x}, \omega)$ for all $k \geq K(\bar{x}, \omega)$.*

Proof. Let $\mathcal{K}(\bar{x}, \omega) := \{k_1, k_2, \ldots, k_s, \ldots\}$ denote a set of increasing iteration numbers such that $k \in \mathcal{K}(\bar{x}, \omega)$ if and only if $x^k = \bar{x}$. From Lemma 1, it follows that if $y^{k_i - 1}(x^{k_i}, \omega), k_i \in \mathcal{K}(\bar{x}, \omega)$ is fractional, then,

$$\alpha^{k_1}(x^{k_1}, \omega) \prec \alpha^{k_2}(x^{k_2}, \omega) \prec \cdots \prec \alpha^{k_s}(x^{k_s}, \omega) \prec \cdots .$$

Furthermore, from Proposition 2, we have $y^k(x^k, \omega) \succeq \alpha^k(x^k, \omega)$ for all k. Since there are only finitely many $y(\bar{x}, \omega)$ that satisfy (27), this implies that in the worst case, we obtain an integer optimal solution to $\mathrm{SP}^k(\bar{x}, \omega)$ at some $k = K(\bar{x}, \omega) \in \mathcal{K}(\bar{x}, \omega)$. Finally, since the cuts generated for $x \in X$ such that $x \neq \bar{x}$ during iterations $k \notin \mathcal{K}(\bar{x}, \omega)$ are also valid for $Y(\bar{x}, \omega)$, in any iteration $k \geq K(\bar{x}, \omega)$, we also have $y^k(\bar{x}, \omega) \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}$ and $f_\ell^k(\bar{x}, \omega) = f(\bar{x}, \omega)$. □

The next result shows that Algorithm 1 is finite.

Theorem 1. *Suppose that Assumptions (A1)-(A5) are satisfied. Then, Algorithm 1 finds an optimal solution to (1)-(7) with $\mathcal{X} = \{x \in \mathbb{B}^{n_1}\}$ and $\mathcal{Y} = \{y \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}\}$ in finitely many iterations.*

Proof. Let $K = \sup_{(x,\omega) \in X \times \Omega} K(x, \omega)$, where $K(x, \omega)$ is defined in Lemma 2. Since there are finitely many $(x, \omega) \in X \times \Omega$, Lemma 2 implies that $K < \infty$. It follows that in the worst case, for $k \geq K$, we obtain $y^k(x^k, \omega) \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}$ and $f_\ell^k(x^k, \omega) = f(x^k, \omega)$, for all $\omega \in \Omega$ and no second-stage cuts are generated after iteration $k > K$. Then, the convergence of Algorithm 1 follows from the convergence of the Benders' or L-shaped methods. □

Corollary 1. *Suppose that Assumptions (A1)-(A5) are satisfied.*

(a) The multicut, round-of-cuts, and the Gomory mixed integer-cut implementations find an optimal solution to (1)-(7) with $\mathcal{X} = \{x \in \mathbb{B}^{n_1}\}$ and $\mathcal{Y} = \{y \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}\}$ in finitely many iterations.

(b) Suppose that we begin with a fixed recourse matrix $W(\omega) = W$ for all $\omega \in \Omega$. Then the fixed recourse matrix implementation finds an optimal solution to (1)-(7) with $\mathcal{X} = \{x \in \mathbb{B}^{n_1}\}$ and $\mathcal{Y} = \{y \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}\}$ in finitely many iterations.

(c) The branch-and-cut implementation finds an optimal solution to (1)-(7) with $\mathcal{X} = \{x \in \mathbb{B}^{n_1}\}$ and $\mathcal{Y} = \{y \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}, 0 \leq y_j \leq 1, j \in J\}$, where $J \subseteq \{1, \ldots, n_2\}$, $J \neq \emptyset$, in finitely many iterations.

Proof. The convergence of all the alternative implementations follows automatically from the arguments given in Lemmas 1 and 2. Indeed, this is obvious for the multicut implementation. For the round-of-cuts and the partial branch-and-cut implementations, since all the cuts including

the one generated from the first fractional source row are present, these results hold. For the Gomory mixed integer cut implementation, it can be shown that Proposition 2 holds (c.f. [17]) and as a result, Lemmas 1 and 2 hold for this case as well. For the fixed recourse matrix implementation, although only one scenario is used to derive the common cuts, note that in the worst case, Gomory cuts for each scenario will be generated eventually. Thus, Lemma 1 holds and convergence follows. □

Several remarks are in order.

Remark 3. Note that in Line 3 of the algorithm, the lexicographic dual simplex method only ensures the convergence of the simplex method. However, the lexicographic dual simplex method in Line 9 is crucial to ensure the rounding property of Proposition 2 when Gomory cuts are added to the LP subproblems and re-optimized.

Remark 4. A number of finite cutting plane algorithms for deterministic MIP problems rely on 'memory' to prove finiteness. For example, in the case of disjunctive cutting plane methods for MIPs, one maintains a memory of row indices for binary MIPs [33, 3] or a tree for general bounded MIPs [13] to obtain sequential convexification. These ideas are carried over to the SMIP case to show convergence of the decomposition algorithms based on disjunctive methods [31]. On the other hand, in Gomory's algorithm one can implicitly associate a lexicographic enumeration tree [23] that automatically provides this memory. For the SIP case, we can associate a lexicographic enumeration tree for each $Y(x,\omega)$ where $(x,\omega) \in X \times \Omega$. Although a particular $\bar{x} \in X$ may reappear only after a few iterations, Lemma 1 shows that the trees corresponding to $Y(\bar{x},\omega), \omega \in \Omega$ are preserved and they get pruned whenever cuts are generated for these sets. In this sense, Lemma 1 offers a broader characterization of convergence of algorithms that use Gomory cuts.

Remark 5. In the algorithms for SMIPs based on the disjunctive decomposition schemes, higher dimensional cut generating linear programs (CGLPs), which themselves have the structure of two-stage stochastic linear programs, are solved to obtain cuts and the right-hand sides functions. On the other hand, the parametric Gomory cuts developed in our framework can be obtained by relatively simple operations. Indeed, the computationally expensive tasks for computing cuts in our algorithm involve lexicographic pivoting and computing $\Gamma(\omega)$ and $\rho(\omega)$. Zanette et al. [40] suggest guidelines for the implementation of a method that obtains the lexicographically smallest solution without performing lexicographic pivots.

6 Examples from the Literature

In this section we illustrate Algorithm 1 using examples from the literature. Our goal here is to illustrate the algorithm via a prototype and exclude any extraneous features. Therefore, we first implement Algorithm 1 in MATLAB (version 2011b). Our implementation includes a full-tableau lexicographic dual simplex method to solve the subproblems. We solve the master problems using pure branch-and-bound with CPLEX. The second-stage cuts are derived from a source row as Chvátal cuts [14], which are equivalent to the fractional cuts and ensure integer cut coefficients. Consider the following example that appears in Sen et al. [32], which is a variation of an example from Schultz et al. [29].

Example 1.

$$\min -1.5x_1 - 4x_2 + \mathbb{E}[f(x, \tilde{\omega})]$$
$$x \in \{0,1\}^2$$

where,

$$f(x,\omega) = \min y_0$$
$$y_0 + 16y_1 + 19y_2 + 23y_3 + 28y_4 - 100R = 0$$
$$2y_1 + 3y_2 + 4y_3 + 5y_4 - R \leq r_1(\omega) - x_1 \tag{29}$$
$$6y_1 + 1y_2 + 3y_3 + 2y_4 - R \leq r_2(\omega) - x_2 \tag{30}$$
$$y_0 \in \mathbb{Z}, y_i \in \{0,1\}, i = 1,\ldots,4, R \in \mathbb{Z}_+, \tag{31}$$

Here $\Omega = \{\omega_1, \omega_2\}$, $p_{\omega_1} = p_{\omega_2} = 0.5$, $(r_1(\omega_1), r_2(\omega_1)) = (5,2)$ and $(r_1(\omega_2), r_2(\omega_2)) = (10,3)$. The variable R is an artificial variable that ensures the relatively complete recourse property. In [32], the authors assume that R is continuous. The following steps illustrate the execution of Algorithm 1 on Example 1. In the interest of brevity, we only give partial details of the first iteration. We also give a few details of iteration 2, which is used later in this section to illustrate the partial branch-and-cut implementation.

Initialization. We put $LB^1 = -\infty, UB^1 = +\infty$, $T^0(\omega_1) = T^0(\omega_2) = T$, $W^0(\omega_1) = W^0(\omega_2) = (W\ I)$ and $r^0(\omega_i) = r(\omega_i), i = 1,2$, where I is a 6×6 identity matrix corresponding to the slack variables. We start with an initial solution $x^1 = (1,1)$.

Iteration 1. Upon solving the subproblems with x^1, we obtain an integral solution to $\mathrm{SP}^0(x^1, \omega_1)$ and a non-integral solution $y(\omega_2) = (-33, 0, 1, 0, 0.5, 0)^\top$ to $\mathrm{SP}^0(x^1, \omega_2)$. For ease of exposition, in the following cut derivation, we do not include the dependence of y on ω_2 explicitly and represent the final cuts in terms of structural variables $y_1, \ldots, y_4, R$. We generate a cut from the source row corresponding to y_4 for ω_2 as follows: By multiplying with the basis inverse corresponding to the LP optimal solution of $\mathrm{SP}^0(x^1, \omega_2)$, we get the source row as

$$3y_1 + 1.5y_3 + y_4 - 0.5R + 0.5s_2 - 0.5s_4 = 1 - 0.5x_2,$$

where s_2 and s_4 are the slacks corresponding to (30) and $y_2 \leq 1$, respectively. Note here that $\rho_i = 1$ and $\Gamma_i = (\ 0 \quad -0.5\ )$, and $\mathcal{B}_i = 4$. Since $x_2 = 1$ in the current first-stage solution, we complement x_2 using $x_2' = 1 - x_2$. Re-writing the source row by rearranging the terms, we get

$$-0.5x_2' + 3y_1 + 1.5y_3 + y_4 - 0.5R + 0.5s_2 - 0.5s_4 = 0.5.$$

The corresponding Chvátal cut obtained by substituting out the slack variables is

$$3y_1 + y_2 + y_3 + y_4 - R \leq 2 - x_2. \tag{32}$$

Note that if we derive the Gomory fractional cut instead of the Chvátal cut, after simplification, we obtain the same inequality (32).

We then obtain $r^1(\omega_2)$, $W^1(\omega_2)$ and $T^1(\omega_2)$ by adding the cut coefficients of (32) to $r^0(\omega_2)$, $W^0(\omega_2)$ and $T^0(\omega_2)$, respectively. With the updated matrices, we optimize $\mathrm{SP}^1(x^1, \omega_2)$ using the lexicographic dual simplex method and obtain the solution $y(\omega_2) = (-28, 0, 0, 0, 1, 0)^\top$. Since

Table 1: Decomposition with Gomory cuts for Example 1

k	x	$f_\ell^{k-1}(x^k,\omega_1)$	$f_\ell^{k-1}(x^k,\omega_2)$	$f_\ell^{k}(x^k,\omega_1)$	$f_\ell^{k}(x^k,\omega_2)$	Cuts	LB^{k+1}	UB^{k+1}
1	(1,1)	-19	-33	-	-28	1	-41.5	-29
2	(1,0)	-25.14	-47	-25	-	1	-39	-29
3	(0,0)	-30.38	-47	-28.33	-	1	-37.67	-29
4	(0,0)	-28.33	-47	-28	-	1	-37.5	-37.5

we have obtained a feasible solution, we update the upper bound $UB^2 = -29$. Using the dual multipliers for the scenario subproblems, we construct an optimality cut (21). The updated master problem MP^1 is

$$\begin{aligned} &\min -1.5x_1 - 4x_2 + \eta \\ &\eta \geq 16.5x_2 - 40 \\ &x \in \{0,1\}^2, \eta \in \mathbb{R}. \end{aligned}$$

The solution of this master problem yields a lower bound $LB^2 = -41.5$ with $x^2 = (1,0)$.

Iteration 2. The scenario solution with $x^2 = (1,0)$ is $y(\omega_1) = (-25.1538, 0.1154, 1, 0, 0.1538, 0)^\top$ for ω_1 and $y(\omega_2)$ is integral. The cut corresponding to y_0 for ω_1 is

$$6y_1 + 2y_2 + 3y_3 + 3y_4 - 2R \leq 4 - x_1. \tag{33}$$

Note that we substituted out y_0 from the cut and expressed it in terms of the structural variables. Updating the matrices for ω_1 using this cut and optimizing $\text{SP}^2(x^2,\omega_1)$, we get a new solution $y(\omega_1) = (-25, 0.0556, 1, 0.2222, 0, 0)^\top$. Since we have not found a feasible solution, we set $UB^3 = UB^2 = 29$. We add the optimality cut $\eta \geq 3.167x_1 + 4.5x_2 - 39$ and when we solve the updated master problem, we a new lower bound lower bound $LB^3 = -39$ and $x^3 = (0,0)$.

In Tables 1 and 2, we provide a summary of iterations of our algorithm and the D^2 algorithm of Sen and Higle [31], respectively on Example 1. In these tables, k indicates the iteration number, $f_\ell^{k-1}(x^k,\omega_s), s = 1,2$ denote the objective function values of the LP relaxation of the scenario sub-problems and $f_\ell^{k}(x^k,\omega_s), s = 1,2$, denote the objective function values after Gomory cut addition at iteration k. If no cut is generated for a scenario, it is indicated by '-'. Finally, Cuts, LB^{k+1} and UB^{k+1} denote the number of Gomory cuts added in the second stage, the lower and upper bounds, respectively at the end of the iteration. From these tables, we observe that using Algorithm 1, we obtain the optimal solution with one additional iteration than number of iterations used by the D^2 algorithm. However, as discussed earlier, cut generation in the D^2 algorithm involves the solution of higher dimensional CGLP and right-hand-side convexification linear programs. On the other hand, cut generation in Algorithm 1 is relatively inexpensive. Another advantage of our algorithm is that we can warm-start the dual simplex method for the scenario subproblems using the basis generated in the previous iteration. In total, only 13 lexicographic dual simplex pivots are performed in the second stage during the execution of Algorithm 1 in the MATLAB implementation.

Next, we illustrate how globally valid cuts can be generated when branch-and-cut is incorporated into the second-stage problems using Example 1. We focus our attention on it-

Table 2: Disjunctive Decomposition for Example 1

k	x	$f_\ell^{k-1}(x^k, \omega_1)$	$f_\ell^{k-1}(x^k, \omega_2)$	$f_\ell^k(x^k, \omega_1)$	$f_\ell^k(x^k, \omega_2)$	Cuts	LB^{k+1}	UB^{k+1}
1	(1,1)	-19	-33	-	-29.2	1	-41.5	$+\infty$
2	(1,0)	-25.13	-47	-25.0	-	1	-38.5	$+\infty$
3	(0,0)	-28	-47	-	-	0	-37.5	-37.5

eration 2 and ω_1 and drop the dependence of y on ω_1 in the following discussion. Recall that in this iteration, we add the cut (33) to $\text{SP}^1(x^2, \omega_1)$ to obtain a fractional solution $y = (-25, 0.0556, 1, 0.2222, 0, 0)^\top$. We may treat this as adding a cut at the root node of a partial branch-and-cut tree for the scenario subproblem. We then create two new branches with $y_1 \leq 0$ and $y_1 \geq 1$. In the following, we discuss cut generation in this partial branch-and-cut tree for the first child node $y_1 \leq 0$. Solving the linear relaxation of the first node, we obtain a fractional solution $y = (-24.75, 0, 1, 0.25, 0, 0)^\top$. The source row corresponding to y_0 is

$$y_0 - 0.75y_4 - 94.25R - 5.75s_1 - 1.75s_4 - 4.5s_8 = -30.5 + 5.75(1 - x_1'),$$

where s_1, s_4, s_8 are slacks corresponding to (29), $y_2 \leq 1$ and $y_1 \leq 0$, respectively. The Chvátal cut expressed in terms of the structural variables is

$$2y_1 + 3y_2 + 3y_3 + 3y_4 - R \leq 4 - x_1$$

Note that this cut is globally valid for $Y(x, \omega_1)$ for all $x \in X$. Adding this cut to the root node LP with $x^2 = (1, 0)$ and re-optimizing, we obtain a solution $(-23.81, 0.09, 0.44, 0, 0.5, 0)^\top$. Adding an optimality cut using the dual multipliers from this solution and solving the updated master problem, we obtain a lower bound $LB^3 = -38.47$. We observe that this lower bound (using branch-and-cut) is an improvement on the previously obtained lower bound of -39, which was produced with a pure cutting plane implementation, at iteration 2..

Example 2. In this example, we make all the second-stage variables from Example 1 general integer, i.e., replace (31) by the following:

$$y_0 \in \mathbb{Z}, y_i \in \{0, 1, \ldots, 5\}, i = 1, \ldots, 4, R \in \mathbb{Z}_+$$

and in addition, assume that $(r_1(\omega_1), r_2(\omega_1)) = (10, 4)$ and $(r_1(\omega_2), r_2(\omega_2)) = (13, 8)$. Table 3 gives the sequence of iterations for our algorithm on Example 2 using the single-cut implementation. Note that the D^2 algorithm of Sen and Higle [31] is not designed for problems with general integer variables in the second stage. Figure 1 shows the manner in which approximations in the (x_1, x_2, z)-space are constructed by our algorithm for Example 2 using the single-cut implementation, where $z = c^\top x + \eta$. In 1(a) we plot the function $z_\ell(x) := c^\top x + \mathbb{E}[f_\ell(x, \tilde{\omega})]$, where $f_\ell(x, \tilde{\omega}) := f_\ell^0(x, \tilde{\omega})$ is the objective value of the initial linear relaxation of the second-stage problems, which provides the best approximation available at the beginning of the algorithm. In addition, the four black circles represent the points $c^\top x + \mathbb{E}[f(x, \tilde{\omega})]$ evaluated at the extreme

Table 3: Algorithm 1 (single cut) for Example 2

k	x	$f_\ell^{k-1}(x^k,\omega_1)$	$f_\ell^{k-1}(x^k,\omega_2)$	$f_\ell^{k}(x^k,\omega_1)$	$f_\ell^{k}(x^k,\omega_2)$	Cuts	LB^{k+1}	UB^{k+1}
1	(1,1)	-57	-77.88	-	-77	1	-77.56	$+\infty$
2	(0,1)	-57	-84	-	-83.29	1	-74.38	$+\infty$
3	(0,0)	-63.75	-84.63	-61.5	-84	2	-74.14	$+\infty$
4	(0,1)	-57	-83.29	-	-82.67	1	-73.83	$+\infty$
5	(0,1)	-57	-82.67	-	-80.5	1	-72.75	$+\infty$
6	(0,0)	-61.5	-84	-57	-81.45	2	-72.75	$+\infty$
7	(0,1)	-57	-80.5	-	-80	1	-72.5	-72.5

Table 4: Algorithm 1 (round of cuts) for Example 2

k	x	$f_\ell^{k-1}(x^k,\omega_1)$	$f_\ell^{k-1}(x^k,\omega_2)$	$f_\ell^{k}(x^k,\omega_1)$	$f_\ell^{k}(x^k,\omega_2)$	Cuts	LB^{k+1}	UB^{k+1}
1	(1,1)	-57	-77.88	-	-76.25	3	-77.56	$+\infty$
2	(0,1)	-57	-84	-	-83.44	1	-75.88	$+\infty$
3	(0,0)	-63.75	-84.63	-58	-84	6	-74.22	$+\infty$
4	(0,1)	-57	-83.44	-	-80	3	-72.5	-72.5

points of the $[0,1]$-box in $\mathbb{R}^2$. Let

$$z^k(x) := c^\top x + \max_{t=1,\dots,k} \left\{ \sum_{\omega\in\Omega} p_\omega (u^t(\omega))^\top (r^t(\omega) - T^t(\omega)x) \right\}.$$

In 1(b), 1(c) and 1(d), in addition to the linear relaxation and the four feasible points for the SIP, we plot the functions $z^3(x), z^5(x)$ and $z^7(x)$, respectively. Figure 1(d) shows that after 7 optimality cuts are added, the approximation intersects the optimal solution $(0,1,-72.5)$ and the algorithm terminates. Figure 1 highlights how the lifting of Gomory cuts allows us to avoid representing sub-additive functions in the master problem and instead, construct affine approximations of the second-stage value functions.

In Table 4 we show the iterations performed when a round of cuts is added in each iteration for Example 2. Note that 13 Gomory cuts are generated in the second stage with this approach whereas 9 were generated for the single-cut version (see Table 3). However, the round-of-cuts approach takes only four iterations to terminate with an optimal solution.

To be able to compare the performance of our algorithm against the deterministic equivalent formulation, we generate four larger instances from a variation of Example 2. The goal of this test is not to compare the performance of our algorithm against commercial MIP solvers, but to test how our decomposition algorithm scales when compared to solving the DEF directly without additional features. The right hand sides for the scenarios (r_1, r_2) in these instances are generated from $\{5, 5+\theta, 5+2\theta, \dots, 15\} \times \{5, 5+\theta, 5+2\theta, \dots, 15\}$ with $\theta \in \{1, 2, 5, 10\}$ giving four instances with 4,9,36 and 121 scenarios. Ahmed et al. [2] use a similar set of instances as one of their test sets. Table 5 shows the results of the comparison of our algorithm against two different settings of CPLEX 12.3 (running on a Mac OSX laptop with a 2.4GHz processor) on the deterministic equivalent formulations. In this table, the column Scen shows the number of scenarios in the instance. Obj indicates the optimal objective function value of the SIP. The

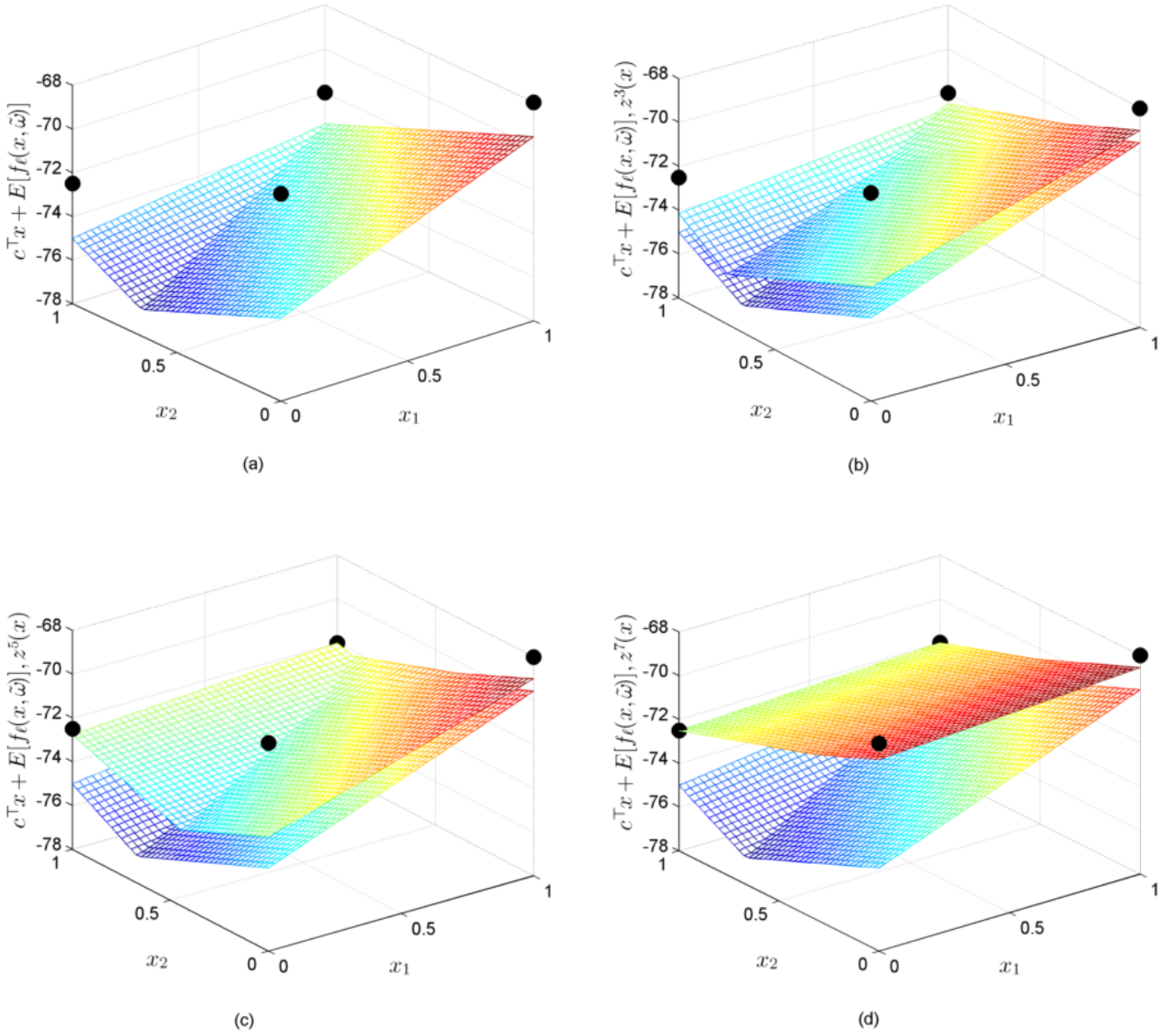

Figure 1: Approximations constructed by Algorithm 1 for Example 2

Table 5: Comparison of Algorithm 1 and the DEF

Scen	Obj	Vars	Cons	GDD-S	GDD-R	B&B	B&B+Gom
4	−63.50	22	24	7 (0, 13)	7 (0, 32)	54	2 (6)
9	−65.67	47	54	8(0, 43)	6 (0, 76)	324	19 (17)
36	−66.83	182	216	10(1, 183)	6 (0, 384)	1.53E7*	206 (50)
121	−67.17	607	726	10(1, 634)	6 (0, 1302)	7.52E6*	11359 (167)

columns Vars and Cons indicate the number of columns and rows in the deterministic equivalent formulation, respectively. For entries under the columns GDD-S (single-cut) and GDD-R (round-of-cuts), we report the number of iterations of the decomposition algorithm with Gomory cuts, followed by two numbers within parentheses: the first number denotes the total number of non-root nodes over all iterations explored by the CPLEX's branch-and-bound tree for the master program (stage 1) with cuts, presolve and heuristics turned off. The second number denotes the total number of Gomory cuts added in the second stage over all iterations. Since our algorithmic design includes only pure branch-and-bound (first-stage) and pure cutting planes in the second-stage, our comparisons with CPLEX in this section is also done with comparable settings to exclude any extraneous features. Thus, we compare two sets of runs using CPLEX: the column B&B indicates the number of pure branch-and-bounds nodes explored by CPLEX on the DEF. The column B&B + Gom shows the number of nodes explored by CPLEX along with the number of cuts added in the parenthesis when we use pure branch-and-bound with Gomory cuts. The solution times for all entries in the columns GDD-S, GDD-R, B&B, B&B + Cuts are less than 5 seconds, except when indicated by an asterisk. An asterisk next to an entry implies that the stipulated time limit of 15 minutes was exceeded. It is important to note that these times are obtained using a prototyping script (i.e. MATLAB), which is a far-cry from the technology underlying CPLEX based on a C/C++ platform that has been developed over the past two decades. In Section 7 we present an initial implementation of Algorithm 1 utilizing such a platform.

From the columns GDD-S and GDD-R in Table 5, we see that the number of master problems solved remains relatively stable as the number of scenarios increases. In addition, the master programs are far smaller in size than the DEF, and as a result the total number of nodes explored for the master problems is negligible for these instances. Although a larger number of cuts are generated in the round-of-cuts version than the single-cut version, the additional computational effort spent in deriving these cuts is trivial and moreover, the number of master programs solved reduces significantly for the last three instances. When CPLEX branch-and-bound is run on the DEF with cuts, presolve and heuristics turned off, the performance is quite poor for the last two instances. In fact, CPLEX exceeds the time limit of 15 minutes after solving millions of nodes for the last two instances. When Gomory cuts are turned on, CPLEX's performance improves with fewer nodes explored. However, even on these relatively small problems the lack of scalability of the DEF is evident, as for the last instance, over eleven thousand branch-and-cut nodes are explored by CPLEX. Based on the instances tested here we see that the number of iterations of the decomposition-based cutting plane procedure is quite stable, relative to CPLEX branch-and-bound or branch-and-cut on the DEF.

7 Preliminary Computational Experience

In this section, we describe a preliminary implementation of the decomposition algorithm with Gomory cuts using the single-cut implementation. We implement the algorithm in C integrated with IBM ILOG CPLEX 12.4 using the C callable library API where CPLEX is used to solve the master problem and subproblems. We conduct our tests on a Windows XP operating system running on a 2.66 GHz Intel CoreQuad processor with 4GB RAM. We compile our code using Visual Studio 2008.

We test our implementation on instances of the stochastic server location problem (SSLP) [26] with at least 50 scenarios. These instances are available online as a part of the stochastic integer programming test problem library (SIPLIB). The original instances have $B = \{1, 2\}$, $C = \{2\}$ and $D = \emptyset$. We convert these instances to pure integer second stage by changing the declaration of the continuous variables to integers. Our implementation also includes a version of the lexicographic dual simplex described in [40, 1]. Although the lexicographic dual simplex is needed for theoretical convergence, we do not observe a significant impact on our computational tests on the SSLP test instances (see [15] for details). Hence, we report our tests with the regular dual simplex method.

Table 6 shows the size of the deterministic equivalent formulation (DEF) and the sizes of the subproblems (Sub) of the test instances. In this table, `Cons`, `Bin` and `Int` denote the number of constraints, binary variables and integer variables, respectively. In Table 7, we report the performance of CPLEX 12.4 with default settings on the deterministic equivalent problems. We impose a time limit of 1 hour. In this table, we also report the total number of cuts generated by CPLEX, the number of branch-and-cut nodes explored by CPLEX and the end gap at termination. We note that out of the 11 instances tested, CPLEX solves 6 instances of the DEFs. Since the default optimality tolerance for CPLEX is 0.01%, instances with end gap of 0.01% are considered optimal.

Table 6: SSLP Instances Description

	DEF			Sub		
Instance	`Cons`	`Bin`	`Int`	`Cons`	`Bin`	`Int`
SSLP_5_25_50	1,501	6,255	250	30	130	5
SSLP_5_25_100	3,001	12,505	500	30	130	5
SSLP_5_50_50	2,751	12,505	250	55	255	5
SSLP_5_50_100	5,501	25,005	500	55	255	5
SSLP_5_50_1000	55,001	250,005	6000	55	255	5
SSLP_5_50_2000	110001	500,005	12,000	55	255	5
SSLP_10_50_50	3,001	25,010	500	60	510	10
SSLP_10_50_100	6,001	50,010	1,000	60	510	10
SSLP_10_50_500	30,001	250,010	5,000	60	510	10
SSLP_10_50_1000	60,001	500,010	10,000	60	510	10
SSLP_10_50_2000	120,001	1,000,010	20,000	60	510	10

In Table 8, we report the results using the decomposition algorithm with the single-cut implementation. The columns `Iter`, `M-Cuts`, `M-Nodes`, `S-Cuts`, `Time`, `Egap` indicate the number of iterations, the total number of cuts generated by CPLEX over all iterations for solving the master problem, the total number of branch-and-cut nodes over all iterations used by CPLEX for solving the master problem, the total number of Gomory cuts generated in the second stage,

Table 7: Deterministic Equivalent Solution by CPLEX

Instance	Cuts	Nodes	Time (s)	Egap
SSLP_5_25_50	1032	0	2.03	0.00%
SSLP_5_25_100	1524	0	1.72	0.00%
SSLP_5_50_50	1216	0	1.06	0.00%
SSLP_5_50_100	2137	0	3.56	0.00%
SSLP_5_50_1000	18351	0	212.64	0.00%
SSLP_5_50_2000	30987	0	1020.54	0.00%
SSLP_10_50_50	3393	189231	801.49	0.01%
SSLP_10_50_100	5473	325739	3667.22	0.10%
SSLP_10_50_500	27837	1307	3601.32	0.38%
SSLP_10_50_1000	57215	0	3610.06	3.56%
SSLP_10_50_2000	46810	0	3601.55	18.59%

the solution time and the end gap at termination, respectively. If no significant improvement is made in the lower bound or percentage gap in 50 iterations, we perform one upper bounding step by solving the scenario subproblems as integer programs and terminate the algorithm. We set the tolerance to check integrality for cut generation to 10^{-4}. In our experiments, we find that a smaller integrality tolerance gives rise to invalid cuts. From these results, we observe that our algorithm performs better than solving the DEF directly on all instances tested. For the largest instances SSLP_10_50_1000 and SSLP_10_50_2000, the deterministic equivalent is very large and we observe that CPLEX does not perform branch-and-cut and only uses cutting planes. The end gaps for the DEF for these instances are relatively large at 3.56% and 18.59%, respectively, whereas the decomposition algorithm terminates on both instances with small end gaps in less than an hour. Moreover, we observe that the decomposition algorithm scales well with respect to the growth in the number of scenarios.

Table 8: Results for the Decomposition Algorithm with Gomory Cuts

Instance	Iter	M-Cuts	M-Nodes	S-Cuts	Time(s)	Egap
SSLP_5_25_50	17	51	0	118	0.18	0.00%
SSLP_5_25_100	17	52	0	212	0.22	0.00%
SSLP_5_50_50	28	90	0	38	0.27	0.00%
SSLP_5_50_100	30	99	0	154	0.48	0.00%
SSLP_5_50_1000	31	98	0	1292	2.88	0.00%
SSLP_5_50_2000	32	104	2	2529	5.73	0.00%
SSLP_10_50_50	419	6913	119660	8022	109.2	0.02%
SSLP_10_50_100	454	6905	96523	18172	218.42	0.02%
SSLP_10_50_500	390	7402	113265	77462	740.38	0.03%
SSLP_10_50_1000	411	7326	112761	157374	1615.42	0.02%
SSLP_10_50_2000	421	6095	77826	314627	2729.61	0.02%

8 Conclusions

We develop a decomposition algorithm that integrates Gomory cuts with the L-shaped method to solve stochastic programs with binary first-stage variables and general integer second-stage variables. We make no assumptions on the elements of the second-stage costs, recourse and technology matrices and right-hand-sides that are affected by the random variables. First, we

address the question whether the expected recourse function $\mathbb{E}[f(x, \tilde{\omega})]$ can be approximated by a lower bounding problem that has the same properties as a Benders' master problem. We show that the algorithm generates piecewise linear convex approximations of the expected recourse function. Next, we address the question on devising an algorithm that avoids solving the scenario subproblems to integer optimality in every iteration. Our algorithm iteratively constructs LP-based approximations of the integer second-stage subproblems. Finally, we address the issue of finite convergence of a decomposition algorithm that uses only cutting planes to approximate the second-stage subproblems by developing parameterized Gomory cuts that are constructed using relatively simple operations. Although Gomory cuts for SIP were first introduced in a conceptual framework in Carøe and Tind [12], we believe that our algorithm is the first incorporation of parametric Gomory cuts within the L-shaped method that creates computationally amenable first-stage approximations for two-stage SIP. Further, we show that the nature of convergence of our algorithm has a robust characterization in that it can be extended to numerous settings and allowing alternative implementations without the need to modify the convergence proofs.

As a result of our development, we can now allow both parametric disjunctive as well as parametric Gomory cuts to be included within finitely convergent decomposition algorithms for SIP. Such disparate collections of cuts have proven to be indispensable for the success of branch-and-cut algorithms in deterministic MIP, and we are hopeful that the introduction of Gomory cuts within a decomposition algorithm will be just as valuable for SIP. Our preliminary computational results are promising and demonstrate the scalability of our algorithm with respect to the growth in the number of scenarios. A detailed computational study of the algorithms developed here with the lexicographic dual simplex method and other extensions will be reported in a subsequent paper.

Acknowledgment

We thank the two referees for their suggestions that improved the paper.

References

1. Achterberg, T.: SCIP: solving constraint integer programs. Mathematical Programming Computation **1**(1), 1–41 (2009)

2. Ahmed, S., Tawarmalani, M., Sahinidis, N.: A finite branch-and-bound algorithm for two-stage stochastic integer programs. Mathematical Programming **100**(2), 355–377 (2004)

3. Balas, E., Ceria, S., Cornuéjols, G.: A lift-and-project cutting plane algorithm for mixed 0–1 programs. Mathematical Programming **58**(1), 295–324 (1993)

4. Balas, E., Ceria, S., Cornuéjols, G., Natraj, N.: Gomory cuts revisited. Operations Research Letters **19**(1), 1–9 (1996)

5. Benders, J.: Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik **4**(1), 238–252 (1962)

6. Birge, J., Louveaux, F.: A multicut algorithm for two-stage stochastic linear programs. European Journal of Operational Research **34**(3), 384–392 (1988)

7. Birge, J., Louveaux, F.: Introduction to stochastic programming. 2 edn. Springer (2011)

8. Bixby, R., Rothberg, E.: Progress in computational mixed integer programming: a look back from the other side of the tipping point. Annals of Operations Research **149**(1), 37–41 (2007)

9. Blair, C., Jeroslow, R.: The value function of an integer program. Mathematical Programming **23**(1), 237–273 (1982)

10. Borozan, V., Cornuéjols, G.: Minimal valid inequalities for integer constraints. Mathematics of Operations Research **34**(3), 538–546 (2009)

11. Carøe, C., Tind, J.: A cutting-plane approach to mixed 0-1 stochastic integer programs. European Journal of Operational Research **101**(2), 306–316 (1997)

12. Carøe, C., Tind, J.: L-shaped decomposition of two-stage stochastic programs with integer recourse. Mathematical Programming **83**(1), 451–464 (1998)

13. Chen, B., Küçükyavuz, S., Sen, S.: Finite disjunctive programming characterizations for general mixed-integer linear programs. Operations Research **59**(1), 202–210 (2011)

14. Chvàtal, V.: Edmonds polytopes and a hierarchy of combinatorial problems. Discrete Mathematics **4**(4), 305–337 (1973)

15. Gade, D.: Algorithms and reformulations for large-scale integer and stochastic integer programs. Ph.D. thesis, The Ohio State University (2012)

16. Gomory, R.: Outline of an algorithm for integer solutions to linear programs. Bulletin of the American Mathematical Society **64**(5), 275–278 (1958)

17. Gomory, R.: An algorithm for the mixed integer problem. Tech. Rep. RM-2597, RAND Corporation (1960)

18. Gomory, R.: Some polyhedra related to combinatorial problems. Linear Algebra and its Applications **2**(4), 451–558 (1969)

19. Kong, N., Schaefer, A., Hunsaker, B.: Two-stage integer programs with stochastic right-hand sides: a superadditive dual approach. Mathematical Programming **108**(2), 275–296 (2006)

20. Laporte, G., Louveaux, F.: The integer L-shaped method for stochastic integer programs with complete recourse. Operations Research Letters **13**(3), 133–142 (1993)

21. Louveaux, F., Schultz, R.: Stochastic integer programming. In: Shapiro, A., Ruszczyński, A. (eds.) Stochastic Programming, *Handbooks in Operations Research and Management Science*, vol. 10, pp. 213–266. Elsevier (2003)

22. Mayr, E.: Some complexity results for polynomial ideals. Journal of complexity **13**(3), 303–325 (1997)

23. Nemhauser, G., Wolsey, L.: Integer and combinatorial optimization. John Wiley & Sons, New York (1988)

24. Nourie, F., Venta, E.: An upper bound on the number of cuts needed in Gomory's method of integer forms. Operations Research Letters **1**(4), 129–133 (1982)

25. Ntaimo, L.: Disjunctive decomposition for two-stage stochastic mixed-binary programs with random recourse. Operations Research **58**(1), 229–243 (2010)

26. Ntaimo, L., Sen, S.: The million-variable march for stochastic combinatorial optimization. Journal of Global Optimization **32**(3), 385–400 (2005)

27. Ntaimo, L., Sen, S.: A comparative study of decomposition algorithms for stochastic combinatorial optimization. Computational Optimization and Applications **40**(3), 299–319 (2008)

28. Schultz, R.: Continuity properties of expectation functions in stochastic integer programming. Mathematics of Operations Research **18**(3), 578–589 (1993)

29. Schultz, R., Stougie, L., van der Vlerk, M.: Solving stochastic programs with integer recourse by enumeration: A framework using Gröbner basis. Mathematical Programming **83**(1), 229–252 (1998)

30. Sen, S.: Algorithms for stochastic mixed-integer programming models. In: Aardal, K., Nemhauser, G., Weismantel, R. (eds.) Discrete Optimization, *Handbooks in Operations Research and Management Science*, vol. 12, pp. 515–558. Elsevier (2003)

31. Sen, S., Higle, J.: The C^3 theorem and a D^2 algorithm for large scale stochastic mixed-integer programming: set convexification. Mathematical Programming **104**(1), 1–20 (2005)

32. Sen, S., Higle, J., Ntaimo, L.: A summary and illustration of disjunctive decomposition with set convexification. In: Woodruff, D. (ed.) Network Interdiction and Stochastic Integer Programming, pp. 105–125. Springer (2003)

33. Sen, S., Sherali, H.: On the convergence of cutting plane algorithms for a class of nonconvex mathematical programs. Mathematical Programming **31**(1), 42–56 (1985)

34. Sen, S., Sherali, H.: Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. Mathematical Programming **106**(2), 203–223 (2006)

35. Sherali, H., Adams, W.: A reformulation-linearization technique for solving discrete and continuous nonconvex problems, *Nonconvex Optimization and its Applications*, vol. 31. Kluwer Academic Publishers (1999)

36. Sherali, H., Fraticelli, B.: A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. Journal of Global Optimization **22**(1), 319–342 (2002)

37. Sherali, H., Zhu, X.: On solving discrete two-stage stochastic programs having mixed-integer first-and second-stage variables. Mathematical Programming **108**(2), 597–616 (2006)

38. Van Slyke, R., Wets, R.: L-shaped linear programs with applications to optimal control and stochastic programming. SIAM Journal of Applied Mathematics **17**(4), 638–663 (1969)

39. Yuan, Y., Sen, S.: Enhanced cut generation methods for decomposition-based branch and cut for two-stage stochastic mixed-integer programs. INFORMS Journal on Computing **21**(3), 480–487 (2009)

40. Zanette, A., Fischetti, M., Balas, E.: Lexicography and degeneracy: can a pure cutting plane algorithm work? Mathematical Programming **130**(1), 1–24 (2010)