FINITELY CONVERGENT DECOMPOSITION ALGORITHMS FOR TWO-STAGE STOCHASTIC PURE INTEGER PROGRAMS *

MINJIAO ZHANG † AND SİMGE KÜÇÜKYAVUZ †‡

Abstract. We study a class of two-stage stochastic integer programs with general integer variables in both stages and finitely many realizations of the uncertain parameters. Based on Benders' method, we propose a decomposition algorithm that utilizes Gomory cuts in both stages. The Gomory cuts for the second-stage scenario subproblems are parameterized by the first-stage decision variables, i.e., they are valid for any feasible first-stage solutions. In addition, we propose an alternative implementation that incorporates Benders' decomposition into a branch-and-cut process in the first stage. We prove the finite convergence of the proposed algorithms. We also report our preliminary computations with a rudimentary implementation of our algorithms to illustrate their effectiveness.

Key words. Two-stage stochastic pure integer programs, Gomory cuts, Benders' decomposition

1. Introduction.

We investigate a class of two-stage stochastic pure integer programs (SIP) with general integer variables in both stages. We assume that the uncertain data follow a finite discrete distribution, where each realization of the uncertain data is referred to as a scenario. Before the uncertainty is revealed (in the first stage), the decision maker makes strategic decisions. After the uncertain parameters are revealed (in the second stage), the decision maker makes operational decisions in response to the realization of the uncertain parameters to optimize an objective. The typical objective function includes the first-stage cost and the expected second-stage cost. This type of decision-making scheme appears in many applications (see [19] for examples).

Let $\tilde{\omega}$ be a random vector defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$. Consider the following SIP with the first-stage variables $\bar{x} := (\bar{x}_1, \dots, \bar{x}_{n_1}) \in \mathbb{Z}_+^{n_1}$ and the second-stage variables $y := (y_0, y_1, \dots, y_{n_2}) \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}$:

$$\min \quad \bar{c}^\top \bar{x} + \mathbb{E}_{\tilde{\omega}}[f(\bar{x}, \tilde{\omega})] \tag{1.1}$$

$$\text{s.t.} \quad \bar{A}\bar{x} \le b, \tag{1.2}$$

$$\bar{x} \in \mathbb{Z}_+^{n_1}, \tag{1.3}$$

where for a realization $\tilde{\omega} = \omega \in \Omega$, $f(\bar{x}, \tilde{\omega})$ is defined as

$$f(\bar{x}, \omega) = \min \quad y_0 \tag{1.4}$$

$$\text{s.t.} \quad W(\omega)y \le r(\omega) - \bar{T}(\omega)x, \tag{1.5}$$

$$y \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}. \tag{1.6}$$

---

*This material is based upon work supported by the National Science Foundation under Grant No. 1100383. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

†Department of Integrated Systems Engineering, The Ohio State University, 1971 Neil Avenue, 210 Baker Systems Building, Columbus, OH 43210 (`zhang.769@osu.edu`, `kucukyavuz.2@osu.edu`)

‡Corresponding author.

Here, $\bar{c} \in \mathbb{Q}^{n_1}$, $\bar{A} \in \mathbb{Q}^{a\times n_1}$, $b \in \mathbb{Q}^a$, the technology matrix $\bar{T}(\omega) \in \mathbb{Q}^{t(\omega)\times n_1}$, the recourse matrix $W(\omega) \in \mathbb{Q}^{t(\omega)\times(n_2+1)}$, and the second-stage right-hand-side vector $r(\omega) \in \mathbb{Q}^{t(\omega)}$ for $\omega \in \Omega$, where a is the number of constraints in the first-stage problem and $t(\omega)$ is the number of constraints in the second-stage problem for $\omega \in \Omega$. Constraints (1.5) include $\sum_{i=1}^{n_2} g_i(\omega)y_i - y_0 \leq 0$, where $g \in \mathbb{Q}^{n_2}$ is the vector of cost coefficients of the second-stage decision variables $\{y_i\}_{i=1}^{n_2}$ and y_0 represents the optimal objective function value of the second-stage problem.

Let $\bar{X} = \{\bar{x} : (1.2) - (1.3)\}$ and $Y(\bar{x}, \omega) = \{y : (1.5) - (1.6)\}$. We make the following assumptions:

(A1) $\bar{c}$, $\bar{A}$, b, $\bar{T}(\omega)$, $W(\omega)$, $r(\omega)$, $g(\omega)$ are integral.
(A2) $\bar{X}$ is nonempty and bounded.
(A3) $|f(\bar{x}, \tilde{\omega})| < +\infty$ for any $(x, \omega) \in \bar{X} \times \Omega$.
(A4) $Y(\bar{x}, \omega) \neq \emptyset$ for any $(\bar{x}, \omega) \in \bar{X} \times \Omega$.
(A5) Ω is finite, where $m := |\Omega|$.

Assumption (A1) is made without loss of generality because we can scale these rational parameters by appropriate multipliers to obtain integers. Assumption (A2) makes sure that there exists at least one feasible solution $\bar{x} \in \bar{X}$ and the objective function value of the SIP is finite. Assumptions (A3) guarantee a bounded optimal solution. Assumption (A4), known as the relatively complete recourse property, ensures that there exists a feasible solution to the second-stage problems for a feasible first-stage solution. From Assumption (A5), we let ω_i denote the i-th realization (scenario) of $\tilde{\omega}$ with $i = 1, \ldots, m$. Let $p_\omega := \mathcal{P}(\tilde{\omega} = \omega) \in [0, 1] \cap \mathbb{Q}$ denote the probability for the realization $\tilde{\omega} = \omega \in \Omega$, where $\sum_{\omega\in\Omega} p_\omega = 1$. Therefore a deterministic equivalent formulation (DEF) for the SIP problem is given by:

$$
\begin{aligned}
\min \quad & \bar{c}^\top \bar{x} + \sum_{\omega\in\Omega} p_\omega y_0(\omega) \\
\text{s.t.} \quad & \bar{A}\bar{x} \leq b, \\
& \bar{T}(\omega)\bar{x} + W(\omega)y(\omega) \leq r(\omega) && \omega \in \Omega, \\
& \bar{x} \in \mathbb{Z}_+^{n_1}, \\
& y(\omega) \in \mathbb{Z} \times \mathbb{Z}_+^{n_2} && \omega \in \Omega.
\end{aligned}
$$

In this paper, we propose a decomposition algorithm based on Benders' [3] and L-shaped methods [18] to solve this very large-scale pure integer program. Our algorithm solves the first-stage problem and the multiple second-stage problems for each scenario as linear programs, and it utilizes Gomory cuts [8] to convexify the first- and second-stage problems. The proposed algorithm has many attractive features, including its applicability to two-stage integer programs with random recourse and technology matrices, and cost and right-hand-side vectors, as well as its use of optimality cuts that are affine in the first-stage general integer variables. We also give an alternative implementation, which solves the first-stage problem using a branch-and-cut algorithm. Our preliminary computational experience is encouraging.

1.1. Literature Review. In this section, we give a brief overview of related research in two-stage stochastic mixed-integer programming. For a more detailed survey on various algorithms for stochastic mixed-integer programming, we refer the reader to Sen [13].

Laporte and Louveaux [11] propose the L-shaped decomposition algorithm for two-stage stochastic programs with binary variables in the first stage and mixed-integer variables in the second stage. This algorithm requires the solution of the second-stage mixed-integer programs to optimality in each iteration. For problems with mixed 0-1 (binary and continuous) variables in both stages, Carøe and Tind [5] propose a method to update the lift-and-project cuts [2] generated from one scenario to be valid for all other scenarios. Sen and Higle [14] develop a decomposition algorithm for the stochastic integer programs with binary variables in the first stage and mixed 0-1 variables in the second-stage. This method involves generating disjunctive cuts to convexify the master problem and scenario subproblems. Sen and Sherali [15] develop an extension of this algorithm that involves branch-and-cut algorithm to solve the second-stage mixed-integer programs. Both of these algorithms utilize the assumption that the recourse matrix is fixed. Sherali and Zhu [16] develop a decomposition-based branch-and-bound algorithm based on a hyperrectangular partitioning process, which relies on the restriction that the second-stage variables are binary or the first-stage variables are extreme points of the hyperrectangular space.

For two-stage stochastic programs with mixed-integer variables in the first stage and general integer variables in the second stage, Carøe and Tind [6] propose a conceptual method that solves the second-stage program for a given first-stage solution to integer optimality by iteratively adding Gomory cuts, and constructs the optimality cuts that are in terms of a series Chvátal functions, which are non-convex [4]. Ahmed et al. [1] develop a finite terminating branch-and-bound method by reformulating this class of problems with the assumption that the technology matrix is fixed. If the technology matrix is dependent on the scenario, then this method needs to branch on the number of scenarios times more variables, which results in exponentially more iterations. In another line of work, Schultz et al. [12] consider the case with continuous variables in the first stage, integer variables in the second stage, and fixed technology and recourse matrices. They propose a method that enumerates all possible optimal solutions, which are contained in a countable set.

With the assumption that the decision variables are pure integers in both stages, Kong et al. [9] propose an equivalent superadditive dual formulation and use a branch-and-bound or level-set approach to find the optimal solution. Also, Trapp et al. [17] develop an algorithmic framework based on the characterization of the value function by level-sets. However, both Kong et al. [9] and Trapp et al. [17] assume that either the second-stage cost function or the technology and recourse matrices are fixed, i.e., they are not affected by the random parameters. In contrast, in this paper, we allow all these data to be random.

One of the most relevant work to this paper is Gade et al. [7], who develop a decomposition algorithm for two-stage stochastic programs with *binary* first-stage decisions and integer second-stage decisions. They allow the second-stage cost function, technology and recourse matrices to be random. However, because this decomposition method exploits the property that the first-stage variables are binary to derive valid cuts for the second stage, it is not directly extendable to stochastic integer programs with general integer variables in the first stage. To the best of our knowledge, there is no *computationally practical* algorithm for two-stage stochastic integer programs with only pure integer variables in both stages when all second stage data (costs, right hand sides, recourse and technology matrices) are random. In this paper, we propose decomposition algorithms based on Benders' method with Gomory cuts parameterized by both the first-stage and second-stage decision variables.

2. A Decomposition Algorithm with Parametric Gomory Cuts. In this section, we develop a decomposition algorithm with Gomory cuts to solve the two-stage SIP. Our overall approach is to utilize Benders' decomposition algorithm to iteratively add optimality cuts to the first-stage problem to approximate the second-stage value function. In each iteration, Gomory cuts are generated after solving the linear relaxations of the first-stage and second-stage subproblems for each scenario separately.

2.1. Parametric Gomory Cuts. Gomory [8] proposes a class of inequalities and a pure cutting plane algorithm for deterministic pure integer programs. Suppose that the decision variables $z \in \mathbb{Z}_+^n$ satisfy a constraint $\sum_{i=1}^n \delta_i z_i = b_0$, where $\delta \in \mathbb{R}^n$ and $b_0 \in \mathbb{R}$, then the inequality $\sum_{i=1}^n \lfloor \delta_i \rfloor z_i \le \sum_{i=1}^n \delta_i z_i = b_0$ is valid, because $\{\lfloor \delta_i \rfloor\}_{i=1}^n \in \mathbb{Z}^n$ and $z \in \mathbb{Z}_+^n$, then $\sum_{i=1}^n \lfloor \delta_i \rfloor z_i \in \mathbb{Z}$. The resulting Gomory cut is

$$\sum_{i=1}^n \lfloor \delta_i \rfloor z_i \le \lfloor b_0 \rfloor. \tag{2.1}$$

In solving a pure integer program with Gomory's cutting plane method, we solve its linear relaxation and generate a Gomory cut in each iteration to cut off a fractional solution. Gomory [8] shows that the optimal integer solution can be found using this pure cutting plane method in finitely many iterations, when the linear programs are solved using the lexicographic dual simplex method and the Gomory cut is generated from the fractional variable with the smallest index.

Given a particular first-stage solution, $\bar{x}$, the Gomory cut obtained as we solve the linear relaxation of the second-stage subproblem for $\omega \in \Omega$ is not necessarily valid for all other feasible first-stage solutions. Our first goal is to develop a Gomory cut $\pi(\omega)^\top y(\omega) \le \pi_0(\bar{x}, \omega)$ that is valid for the linear relaxation of the DEF, where $\pi(\omega) \in \mathbb{Z}^{n_2+1}$ and $\pi_0(\bar{x}, \omega)$ is an affine function of the first-stage decisions $\bar{x}$.

In the rest of this paper, we redefine matrices $\bar{A}$, $\bar{T}(\tilde{\omega})$, and $W(\tilde{\omega})$ to include the slack variables in both stages, which are added for implementing simplex method. For a given optimal solution to the linear relaxation of the first-stage problem, let $\bar{A}_{B_1} = [\bar{A}_{B_1(1)}, \ldots, \bar{A}_{B_1(a)}]$ denote the corresponding basis matrix for the first-stage problem, in which $B_1(1), \ldots, B_1(a)$ are the indices of the columns in the basis matrix and B_1 stands for basis for the first-stage problem. Denote $\bar{x}_{B_1} = (\bar{x}_{B_1(1)}, \ldots, \bar{x}_{B_1(a)})$ as the first-stage basic variables. Note that $\bar{x}_{B_1} = \bar{A}_{B_1}^{-1} b$. In addition, let $\bar{T}_{B_1}(\omega) = [\bar{T}_{B_1(1)}(\omega), \ldots, \bar{T}_{B_1(a)}(\omega)]$, $\omega \in \Omega$ be defined similarly. Then we solve the linear relaxation of the second-stage problem for the given $\bar{x}$, and let $W_{B_2}(\omega)$ denote the corresponding basis matrix to the second-stage problem for $\omega \in \Omega$, where B_2 stands for basis for the second-stage problem. Note that B_2 is dependent on $\tilde{\omega}$, but we drop this dependence for notational convenience. Then the second-stage basic variables $y_{B_2}(\omega_i) = W_{B_2}(\omega_i)^{-1}(r(\omega_i) - \bar{T}(\omega_i)\bar{x})$ for $\omega_i \in \Omega$.

LEMMA 2.1. $\begin{bmatrix} \bar{A}_{B_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \bar{T}_{B_1}(\omega_1) & W_{B_2}(\omega_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \bar{T}_{B_1}(\omega_2) & \mathbf{0} & W_{B_2}(\omega_2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{T}_{B_1}(\omega_m) & \mathbf{0} & \mathbf{0} & \cdots & W_{B_2}(\omega_m) \end{bmatrix}$ *is a feasible basis matrix for DEF.*

Proof. First, because the columns of the matrices $\bar{A}_{B_1}$, $W_{B_2}(\omega_1)$, ..., $W_{B_2}(\omega_m)$ are linearly independent, clearly the matrix $\begin{bmatrix} \bar{A}_{B_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \bar{T}_{B_1}(\omega_1) & W_{B_2}(\omega_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \bar{T}_{B_1}(\omega_2) & \mathbf{0} & W_{B_2}(\omega_2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{T}_{B_1}(\omega_m) & \mathbf{0} & \mathbf{0} & \cdots & W_{B_2}(\omega_m) \end{bmatrix}$ is linearly independent.

To show the feasibility of the basis matrix, we also need to prove that

$$\begin{bmatrix} \bar{A}_{B_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \bar{T}_{B_1}(\omega_1) & W_{B_2}(\omega_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \bar{T}_{B_1}(\omega_2) & \mathbf{0} & W_{B_2}(\omega_2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{T}_{B_1}(\omega_m) & \mathbf{0} & \mathbf{0} & \cdots & W_{B_2}(\omega_m) \end{bmatrix}^{-1} \begin{bmatrix} b \\ r(\omega_1) \\ r(\omega_2) \\ \vdots \\ r(\omega_m) \end{bmatrix} \geq \mathbf{0}.$$

Because $\bar{x}_{B_1} = \bar{A}_{B_1}^{-1} b \geq 0$ and $y_{B_2}(\omega_i) = W_{B_2}(\omega_i)^{-1}(r(\omega_i) - \bar{T}_{B_1}(\omega_i)\bar{x}_{B_1}) \geq 0$ for $i = 1, \ldots, m$, we have

$$\begin{bmatrix} \bar{A}_{B_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \bar{T}_{B_1}(\omega_1) & W_{B_2}(\omega_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \bar{T}_{B_1}(\omega_2) & \mathbf{0} & W_{B_2}(\omega_2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{T}_{B_1}(\omega_m) & \mathbf{0} & \mathbf{0} & \cdots & W_{B_2}(\omega_m) \end{bmatrix}^{-1} \begin{bmatrix} b \\ r(\omega_1) \\ r(\omega_2) \\ \vdots \\ r(\omega_m) \end{bmatrix}$$
$$= \begin{bmatrix} \bar{A}_{B_1}^{-1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -W_{B_2}(\omega_1)^{-1}\bar{T}_{B_1}(\omega_1)\bar{A}_{B_1}^{-1} & W_{B_2}(\omega_1)^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ -W_{B_2}(\omega_2)^{-1}\bar{T}_{B_1}(\omega_2)\bar{A}_{B_1}^{-1} & \mathbf{0} & W_{B_2}(\omega_2)^{-1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -W_{B_2}(\omega_m)^{-1}\bar{T}_{B_1}(\omega_m)\bar{A}_{B_1}^{-1} & \mathbf{0} & \mathbf{0} & \cdots & W_{B_2}(\omega_m)^{-1} \end{bmatrix} \begin{bmatrix} b \\ r(\omega_1) \\ r(\omega_2) \\ \vdots \\ r(\omega_m) \end{bmatrix}$$

$$= \begin{bmatrix} \bar{A}_{B_1}^{-1}b \\ -W_{B_2}(\omega_1)^{-1}\bar{T}_{B_1}(\omega_1)\bar{A}_{B_1}^{-1}b + W_{B_2}(\omega_1)^{-1}r(\omega_1) \\ -W_{B_2}(\omega_2)^{-1}\bar{T}_{B_1}(\omega_2)\bar{A}_{B_1}^{-1}b + W_{B_2}(\omega_2)^{-1}r(\omega_2) \\ \vdots \\ -W_{B_2}(\omega_m)^{-1}\bar{T}_{B_1}(\omega_m)\bar{A}_{B_1}^{-1}b + W_{B_2}(\omega_m)^{-1}r(\omega_m) \end{bmatrix} = \begin{bmatrix} \bar{x}_{B_1} \\ y_{B_2}(\omega_1) \\ y_{B_2}(\omega_2) \\ \vdots \\ y_{B_2}(\omega_n) \end{bmatrix} \geq \mathbf{0}.$$

□

For given first-stage basis matrix $\bar{A}_{B_1}$, second-stage basis matrices $W_{B_2}(\omega)$, and the submatrices $\bar{T}_{B_1}(\omega)$ for $\omega \in \Omega$, let

$$G := \begin{bmatrix} \bar{A}_{B_1}^{-1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -W_{B_2}(\omega_1)^{-1}\bar{T}_{B_1}(\omega_1)\bar{A}_{B_1}^{-1} & W_{B_2}(\omega_1)^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ -W_{B_2}(\omega_2)^{-1}\bar{T}_{B_1}(\omega_2)\bar{A}_{B_1}^{-1} & \mathbf{0} & W_{B_2}(\omega_2)^{-1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -W_{B_2}(\omega_m)^{-1}\bar{T}_{B_1}(\omega_m)\bar{A}_{B_1}^{-1} & \mathbf{0} & \mathbf{0} & \cdots & W_{B_2}(\omega_m)^{-1} \end{bmatrix}.$$

Then the Gomory cuts generated from any row of

$$G \begin{bmatrix} \bar{A} \\ \bar{T}(\omega_1) \\ \bar{T}(\omega_2) \\ \vdots \\ \bar{T}(\omega_m) \end{bmatrix} \bar{x} + G \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ W(\omega_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & W(\omega_2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & W(\omega_m) \end{bmatrix} y = G \begin{bmatrix} b \\ r(\omega_1) \\ r(\omega_2) \\ \vdots \\ r(\omega_m) \end{bmatrix} \tag{2.2}$$

are valid for DEF. Such cuts are referred to as *parametric Gomory cuts* in the rest of this paper, because they are parameterized with respect to the first-stage decision variables $\bar{x}$. We demonstrate these cuts on an instance that is adapted from the test set 1 in Ahmed et al. [1]. Throughout the paper, we let $[i, j] := \{i, i+1, \ldots, j\}$ for $i, j \in \mathbb{Z}$.

Example 1. *Consider a two-stage SIP with $a = 2$, $t(\omega) = 2$ and $m = 3$, whose DEF is given by*

$$\min \quad -6m\bar{x}_1 - 16m\bar{x}_2 - \sum_{i=1}^{m} y_0(\omega_i)$$

$$\text{s.t.} \quad \bar{x}_1 \leq 5, \tag{2.3}$$

$$\bar{x}_2 \leq 5, \tag{2.4}$$

$$y_0(\omega_1) - 17y_1(\omega_1) - 20y_2(\omega_1) - 24y_3(\omega_1) - 28y_4(\omega_1) \leq 0, \tag{2.5}$$

$$3y_1(\omega_1) + 4y_2(\omega_1) + 5y_3(\omega_1) + 5y_4(\omega_1) \leq 24 - 2\bar{x}_1, \tag{2.6}$$

$$7y_1(\omega_1) + y_2(\omega_1) + 4y_3(\omega_1) + 3y_4(\omega_1) \leq 23 - 3\bar{x}_2, \tag{2.7}$$

$$y_0(\omega_2) - 17y_1(\omega_2) - 19y_2(\omega_2) - 24y_3(\omega_2) - 28y_4(\omega_2) \leq 0, \tag{2.8}$$

$$(2.9)\qquad 3y_1(\omega_2)+3y_2(\omega_2)+4y_3(\omega_2)+6y_4(\omega_2)\le 27-3\bar{x}_1,$$
$$(2.10)\qquad 6y_1(\omega_2)+y_2(\omega_2)+4y_3(\omega_2)+3y_4(\omega_2)\le 22-\bar{x}_2,$$
$$(2.11)\qquad y_0(\omega_3)-16y_1(\omega_3)-19y_2(\omega_3)-24y_3(\omega_3)-29y_4(\omega_3)\le 0,$$
$$(2.12)\qquad 2y_1(\omega_3)+3y_2(\omega_3)+4y_3(\omega_3)+6y_4(\omega_3)\le 29-4\bar{x}_1,$$
$$(2.13)\qquad 6y_1(\omega_3)+2y_2(\omega_3)+4y_3(\omega_3)+3y_4(\omega_3)\le 23-4\bar{x}_2,$$
$$\bar{x}\in\mathbb{Z}^2_+,$$
$$y(\omega_i)\in\mathbb{Z}^5_+,\quad i\in[1,3].$$

First, we introduce the slack variables $\bar{x}_3$, $\bar{x}_4$, $\{y_5(\omega_i)\}_{i\in[1,3]}$, $\{y_6(\omega_i)\}_{i\in[1,3]}$, and $\{y_7(\omega_i)\}_{i\in[1,3]}$ to put the problem in standard form. Then, we solve the linear relaxation of the first-stage problem

$$\begin{aligned}\min\quad & -18\bar{x}_1-48\bar{x}_2\\ \text{s.t}\quad & \bar{x}_1+\bar{x}_3=5,\\ & \bar{x}_2+\bar{x}_4=5,\\ & \bar{x}\in\mathbb{R}^4_+,\end{aligned}$$

by lexicographic simplex method. The optimal tableau is

	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\bar{x}_4$
330	0	0	18	48
5	1	0	1	0
5	0	1	0	1

Thus $\bar{A}_{B_1}=\begin{bmatrix}1&0\\0&1\end{bmatrix}$, $\bar{A}^{-1}_{B_1}=\begin{bmatrix}1&0\\0&1\end{bmatrix}$, *and the optimal solution is* $\bar{x}=(5,5,0,0)$. *Then we solve the linear relaxation of the second-stage subproblem for given $\bar{x}$ for the first scenario by lexicographic simplex method. The optimal tableau is*

	$y_0(\omega_1)$	$y_1(\omega_1)$	$y_2(\omega_1)$	$y_3(\omega_1)$	$y_4(\omega_1)$	$y_5(\omega_1)$	$y_6(\omega_1)$	$y_7(\omega_1)$
77.71	0	8.71	0	5.71	0	1	4.57	1.71
77.71	1	8.71	0	5.71	0	1	4.57	1.71
0.29	0	-3.71	1	-0.71	0	0	0.43	-0.71
2.57	0	3.57	0	1.57	1	0	-0.14	0.57

Here, $W_{B_2}(\omega_1)=\begin{bmatrix}1&-20&-28\\0&4&5\\0&1&3\end{bmatrix}$, *and* $\bar{T}_{B_1}(\omega_1)=\begin{bmatrix}0&0\\2&0\\0&3\end{bmatrix}$.

Therefore, $\begin{bmatrix}\bar{A}^{-1}_{B_1} & \mathbf{0}\\ -W_{B_2}(\omega_1)^{-1}\bar{T}_{B_1}(\omega_1)\bar{A}^{-1}_{B_1} & W_{B_2}(\omega_1)^{-1}\end{bmatrix}=\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\-9.14&-5.14&1&4.57&1.71\\-0.86&2.14&0&0.43&-0.71\\0.29&-1.71&0&-0.14&0.57\end{bmatrix}$.

Consider the source row corresponding to $y_0(\omega_1)$*:*

$$\begin{bmatrix} -9.14 \\ -5.14 \\ 1 \\ 4.57 \\ 1.71 \end{bmatrix}^\top \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -17 & -20 & -24 & -28 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 3 & 4 & 5 & 5 & 0 & 1 & 0 \\ 0 & 3 & 0 & 0 & 0 & 7 & 1 & 4 & 3 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x} \\ y \end{bmatrix} = \begin{bmatrix} -9.14 \\ -5.14 \\ 1 \\ 4.57 \\ 1.71 \end{bmatrix}^\top \begin{bmatrix} 5 \\ 5 \\ 0 \\ 24 \\ 23 \end{bmatrix}.$$

The Gomory cut obtained from this row is $6y_1(\omega_1) + 3y_2(\omega_1) + 5y_3(\omega_1) + 5y_4(\omega_1) \leq 38 - 2\bar{x}_1 - 3\bar{x}_2$ *after substituting out the slack variables. This cut is valid for the second-stage problem for* ω_1 *for any given* $\bar{x}$*.*

If we generate the Gomory cut directly from the source row $y_0(\omega_1) + 8.71y_1(\omega_1) + 5.71y_3(\omega_1) + y_5(\omega_1) + 4.57y_6(\omega_1) + 1.71y_7(\omega_1) = 77.71$ *in the second-stage optimal tableau, then the Gomory cut obtained is*

$$6y_1(\omega_1) + 3y_2(\omega_1) + 5y_3(\omega_1) + 5y_4(\omega_1) \leq 13, \tag{2.14}$$

after substituting out $y_5(\omega_1)$*,* $y_6(\omega_1)$*, and* $y_7(\omega_1)$*. However, this inequality is not necessarily valid for other* $\bar{x} \in \bar{X}$*. For example, for* $\bar{x} = (0, 0, 5, 5) \in \bar{X}$*, the constraints* (2.6)-(2.7) *are reduced to* $3y_1(\omega_1) + 4y_2(\omega_1) + 5y_3(\omega_1) + 5y_4(\omega_1) + y_6(\omega_1) = 24$ *and* $7y_1(\omega_1) + y_2(\omega_1) + 4y_3(\omega_1) + 3y_4(\omega_1) + y_7(\omega_1) = 23$*. The second stage solution* $y(\omega_1) = (82, 2, 0, 2, 0) \in \{y(\omega_1) \in \mathbb{Z}_+^6 : (2.5) - (2.7), \bar{x} = (0, 0, 5, 5)\}$ *violates inequality* (2.14)*, hence inequality* (2.14) *is not valid for* $\bar{x} = (0, 0, 5, 5)$*.*

Next, we develop a decomposition algorithm using parametric Gomory cuts to solve two-stage pure SIPs.

2.2. A Cutting Plane Based Decomposition Algorithm. We introduce two additional variables x_{n_1+1} and x_{n_1+2} to represent the second-stage value function by $x_{n_1+1} - x_{n_1+2}$. Let $x^\top := (\bar{x}^\top, x_{n_1+1}, x_{n_1+2}) \in \mathbb{Z}_+^{n_1+2}$, $c^\top := (\bar{c}^\top, 1, -1)$, $T(\omega) = \left[\bar{T}(\omega)\ \mathbf{0}_{t(\omega)\times 1}\ \mathbf{0}_{t(\omega)\times 1}\right]$, and $A = [\bar{A}\ \mathbf{0}_{a\times 1}\ \mathbf{0}_{a\times 1}]$. Define the master problem at iteration k as MP^k, where

$$\begin{aligned} \mathrm{MP}^k: \quad & \min \quad c^\top x \\ & \text{s.t.} \quad A^k x \leq b^k, \\ & \qquad\ \ x \in \mathbb{R}_+^{n_1+2}, \end{aligned} \tag{2.15}$$

where

1. for $k = 0$, $A^0 = \bar{A}$, $b^0 = b$, $x = \bar{x}$, $c = \bar{c}$,
2. for $k \geq 1$, $A^k x \leq b^k$ includes the original constraints $\bar{A}x \leq b$, the parametric Gomory cuts generated for the first-stage problem, and the optimality cuts

$$\sum_{\omega\in\Omega} p_\omega(\beta^j(\omega))^\top (r^j(\omega) - T^j(\omega)x) \leq x_{n_1+1} - x_{n_1+2} \tag{2.16}$$

generated in iterations $j = 1, \ldots, k-1$, where $\beta^j(\omega)$ is the optimal dual vector of the subproblem $\mathrm{SP}^k(x, \omega)$. The subproblem $\mathrm{SP}^j(x, \omega)$ is given by

$$\begin{aligned} \mathrm{SP}^k(x,\omega): \quad f^k(x,\omega) := \min \quad & y_0(\omega) \\ \text{s.t.} \quad & W^k(\omega)y(\omega) \leq r^k(\omega) - T^k(\omega)x, \\ & y(\omega) \in \mathbb{R}_+^{n_2+1}, \end{aligned} \tag{2.17}$$

where $W^k(\omega)y(\omega) \leq r^k(\omega) - T^k(\omega)x$ includes the original constraints $W(\omega)y(\omega) \leq r(\omega) - T(\omega)x$, and the Gomory cuts generated for the second-stage problem for realization ω in iterations 1 to $k-1$. Note that if the second-stage objective function value $\sum_{i=1}^{n_2} g_i(\omega)y_i$ is not necessarily nonnegative, then we can introduce one more second-stage variable, e.g., $y_0'(\omega) \in \mathbb{R}_+$, to represent the second-stage objective by $y_0(\omega) - y_0'(\omega)$. For ease of exposition of the algorithm, in the rest of the paper, we assume that $y_0(\omega) = \sum_{i=1}^{n_2} g_i(\omega)y_i \geq 0$. We scale inequalities (2.16) and (2.17) so that all coefficients are integral.

Let l_k be the number of rows in matrix A^k, LB and UB be the lower and upper bounds of the optimal objective function value of the DEF. Let $q \in \mathbb{Z}_+$ be the frequency of implementing full Gomory cutting plane method to the master problem. In other words, every q iterations, we implement a pure cutting plane algorithm to solve the master problem to integer optimality. In all other iterations, we solve the linear relaxation of the master problem. (Note that, in these iterations, we could also add one or more violated Gomory cuts.) Initially, we have $l_0 = a$, $LB = -\infty$, $UB = +\infty$, and solve MP^0 to obtain x^0. In iteration k, we first solve the subproblems $\mathrm{SP}^k(x^{k-1}, \omega)$ for $\omega \in \Omega$. If the optimal solution $y^k(\omega) \in \mathbb{Z}_+^{n_2+1}$ for every $\omega \in \Omega$, then we update the upper bound, UB. Otherwise, we generate a parametric Gomory cut from the fractional component in $y^k(\omega)$ with the smallest index. The matrices $T^k(\omega)$, $W^k(\omega)$, and $r^k(\omega)$ are updated with this parametric Gomory cut. Then we solve the updated subproblem by lexicographic dual simplex method and update the upper bound if the optimal solution $y^k(\omega) \in \mathbb{Z}_+^{n_2+1}$ for every $\omega \in \Omega$. We add the optimality cut (2.16) obtained from the subproblems $\{\mathrm{SP}^k(x^{k-1}, \omega)\}_{\omega\in\Omega}$ to the master problem MP^{k-1}. We solve the resulting master problem MP^k with the updated matrix A^k and vector b^k to obtain the optimal solution x^k. If $x^k \notin \mathbb{Z}_+^{n_1+2}$ and $k = 0 \bmod q$, then we construct a Gomory cut corresponding to the fractional component in x^k with the smallest index. Then we update the matrix A^k with this Gomory cut and solve the updated problem with lexicographic dual simplex method. If $x^k \notin \mathbb{Z}_+^{n_1+2}$ again, then we continue to add Gomory cuts to the master problem, and re-solve it with lexicographic dual simplex method until $x^k \in \mathbb{Z}_+^{n_1+2}$. The lower bound is updated by the optimal objective function value of MP^k. If $UB - LB \leq \epsilon$, where ϵ is a very small nonnegative constant, then we have found an optimal integer solution and we stop. Otherwise, we repeat this process until $UB - LB \leq \epsilon$.

For a given q, the algorithm for solving a SIP problem is given in Algorithm 1, in which Algorithm 2 is for solving the subproblems. Note that if $q = 1$, then

Algorithm 1 becomes a pure Gomory cutting plane algorithm for the master problem, and it stops with an integer first-stage solution. In contrast, we solve at most two linear programs and add at most one Gomory cut for each second stage subproblem at every iteration. Although Gomory cutting plane algorithm is finitely convergent, its convergence may be slow for practical purposes. However, solving the master problem to integer optimality with this algorithm at every iteration is not necessary. A potentially more practical implementation is to solve the master problem to integer optimality at every $q > 1$ iterations.

Algorithm 1: Algorithm for solving a SIP problem

```
Initialization: k ← 0, l_k ← a, LB ← −∞, UB ← +∞, A^k ← A,
W^k(ω) ← W(ω), T^k(ω) ← T(ω), r^k(ω) ← r(ω);
Solve MP^k with lexicographic simplex to obtain (x_1^k, ..., x_{n_1}^k);
Set LB as the objective function value of MP^k;
Let the basis matrix be A_{B_1}^k = [A_{B_1(1)}^k, ..., A_{B_1(l_k)}^k] and
T_{B_1}^k(ω) := [T_{B_1(1)}^k(ω), ..., T_{B_1(l_k)}^k(ω)] for each ω ∈ Ω;
while UB-LB > ϵ do
    k ← k + 1;
    Input k, UB, c, x^{k−1}, A^{k−1}, A_{B_1}^{k−1}, W^{k−1}(ω), T^{k−1}(ω), T_{B_1}^{k−1}(ω), r^{k−1}(ω)
    for all ω ∈ Ω to Algorithm 2 ;
    Add the optimality cut returned from Algorithm 2 to MP^{k−1} to get MP^k
    with the updated matrix A^k and vector b^k, l_k ← l_k + 1;
    Solve MP^k with lexicographic dual simplex to obtain (x_1^k, ..., x_{n_1+2}^k);
    while x^k ∉ Z_+^{n_1+2} and k = 0 mod q do
        Generate a Gomory cut from the source row corresponding to
        min{j : j ∈ {1, ..., n_1 + 2}, x_j^k ∉ Z_+};
        Add the Gomory cut to MP^k, and update A^k and b^k;
        l_k ← l_k + 1;
        Solve MP^k using the lexicographic dual simplex method to obtain
        (x_1^k, ..., x_{n_1+2}^k);
    end
    Update LB as the objective function value of MP^k;
    Let the basis matrix be A_{B_1}^k = [A_{B_1(1)}^k, ..., A_{B_1(l_k)}^k] and
    T_{B_1}^k(ω) := [T_{B_1(1)}^k(ω), ..., T_{B_1(l_k)}^k(ω)] for each ω ∈ Ω;
end
Return x^k, UB, LB.
```

Next we illustrate Algorithm 1 on Example 1.

Example 1. *(Continued). Let $q = 1$. In the rest of this example, we define $x = (x_1, x_2, x_3, x_4) := (\bar{x}_1, \bar{x}_2, x_3, x_4)$, where $x_3 - x_4$ represents $\sum_{i=1}^{3} y_0(\omega_i)$.*

Initialization. Let $UB = +\infty$, $LB = -\infty$, $A^0 \leftarrow A$, $T^0(\omega_i) = T(\omega_i)$, $W^0(\omega_i) = W(\omega_i)$, $r^0(\omega_i) = r(\omega_i)$, $i \in [1, 3]$. We solve MP^0 with lexicographic simplex method and obtain $x^0 = (5, 5, 0, 0)$.

Iteration 1. We solve subproblems $SP^1(x^0, \omega_i)$, $i \in [1, 3]$ with lexicographic simplex

Algorithm 2: Algorithm for solving the subproblems

Given k, UB, c, x^{k-1}, A^{k-1}, $A_{B_1}^{k-1}$, $W^{k-1}(\omega)$, $T^{k-1}(\omega)$, $T_{B_1}^{k-1}$, $r^{k-1}(\omega)$ for all $\omega \in \Omega$, solve subproblems $\mathrm{SP}^{k-1}(x^{k-1}, \omega)$ using the lexicographic dual simplex method to obtain $y^k(\omega)$;

if $y^k(\omega) \in \mathbb{Z}_+^{n_2+1}$ *for all* $\omega \in \Omega$ **then**

  $W^k(\omega) \leftarrow W^{k-1}(\omega)$, $T^k(\omega) \leftarrow T^{k-1}(\omega)$, $r^k(\omega) \leftarrow r^{k-1}(\omega)$,

  $UB \leftarrow \min\{UB, c^\top x^{k-1} + \sum_{\omega\in\Omega} p_\omega f^{k-1}(x^{k-1}, \omega)\}$;

else

  For each $\omega \in \Omega$, let $i^* = \min\{i \in \{0, \dots, n_2\}, y_i^k(\omega) \notin \mathbb{Z}\}$;

  Let $W_{B_2}^{k-1}(\omega)$ be the basis matrix corresponding to $y^k(\omega)$;

  Let $d_{i^*}^k(\omega) := \left[-(W_{B_2}^{k-1}(\omega))_{i^*}^{-1} T_{B_1}^{k-1}(\omega) A_{B_1}^{k-1^{-1}} \quad W_{B_2}^{k-1}(\omega)_{i^*}^{-1} \right]$, where $(W_{B_2}^{k-1}(\omega))_{i^*}^{-1}$ is the i^*th row of the matrix $W_{B_2}^{k-1}(\omega)^{-1}$;

  Generate a Gomory cut from the source row

  $d_{i^*}^k(\omega) \begin{bmatrix} A^{k-1} & \mathbf{0} \\ T^{k-1}(\omega) & W^{k-1}(\omega) \end{bmatrix} \begin{bmatrix} x \\ y(\omega) \end{bmatrix} = d_{i^*}^k(\omega) \begin{bmatrix} b^{k-1} \\ r^{k-1}(\omega) \end{bmatrix}$;

  Add the cut to $\mathrm{SP}^{k-1}(x^{k-1}, \omega)$ to get $\mathrm{SP}^k(x^{k-1}, \omega)$ with updated $W^k(\omega), T^k(\omega), r^k(\omega)$;

  Solve $\mathrm{SP}^k(x^{k-1}, \omega)$ using the lexicographic dual simplex method to obtain $y^k(\omega)$;

  if $y^k(\omega) \in \mathbb{Z}_+^{n_2+1}$ *for all* $\omega \in \Omega$ **then**

  $UB \leftarrow \min\{UB, \bar{c}^\top \bar{x}^{k-1} + \sum_{\omega\in\Omega} p_\omega f^k(x^{k-1}, \omega)\}$;

end

return optimality cut (2.16) and $y^k(\omega), UB, W^k(\omega), T^k(\omega), r^k(\omega)$ for all $\omega \in \Omega$ to Algorithm 1;

method. The optimal solution to $SP^1(x^0, \omega_1)$ is $y^1(\omega_1) = (77.71, 0, 0.29, 0, 2.57) \notin \mathbb{Z}_+^5$. As demonstrated earlier, we add a parametric Gomory cut

$$6y_1(\omega_1) + 3y_2(\omega_1) + 5y_3(\omega_1) + 5y_4(\omega_1) \le 38 - 2x_1 - 3x_2 \tag{2.18}$$

to $SP^1(x^0, \omega_1)$, and obtain $y^1(\omega_1) = (76, 0, 1, 0, 2) \in \mathbb{Z}_+^5$. The optimal solution to $SP^1(x^0, \omega_2)$ is $y^1(\omega_2) = (76, 0, 4, 0, 0)$, which is integral. The optimal solution to $SP^1(x^0, \omega_3)$ is $y^1(\omega_3) = (28.5, 0, 1.5, 0, 0) \notin \mathbb{Z}_+^5$, so we add a parametric Gomory cut

$$3y_1(\omega_3) + y_2(\omega_3) + 2y_3(\omega_3) + y_4(\omega_3) \le 11 - 2x_2 \tag{2.19}$$

to $SP^1(x^0, \omega_3)$. The new optimal solution $y^1(\omega_3) = (28, 0, 0, 0, 1) \in \mathbb{Z}_+^5$. We return the optimality cut

$$453x_1 + 678x_2 - 15x_3 + 15x_4 \le 8355 \tag{2.20}$$

to the master problem MP^0 to obtain MP^1. The optimal solution to the master problem MP^1 is $x^1 = (0, 5, 0, 331) \in \mathbb{Z}_+^4$. We update the lower bound as $LB = -571$. Because $y^1(\omega_i) \in \mathbb{Z}_+^5$, $i \in [1, 3]$, we update the upper bound as $UB = -510$.

Iteration 2. With $x^1 = (0, 5, 0, 331)$, the optimal solution to subproblem $SP^2(x^1, \omega_1)$

is $y^2(\omega_1) = (123.43, 0, 4.57, 0, 1.14) \notin \mathbb{Z}_+^5$. *We add a parametric Gomory cut*

$$6y_1(\omega_1) + 3y_2(\omega_1) + 5y_3(\omega_1) + 5y_4(\omega_1) \le 34 - x_1 - 3x_2 \tag{2.21}$$

to $SP^2(x^1, \omega_1)$ *and obtain* $y^2(\omega_1) = (122.4, 0, 5, 0, 0.8)$. *Similarly, the optimal solution to subproblem* $SP^2(x^1, \omega_2)$ *is* $y^2(\omega_2) = (171, 0, 9, 0, 0)$, *and the optimal solution to subproblem* $SP^2(x^1, \omega_3)$ *is* $y^2(\omega_3) = (28, 0, 0, 0, 1)$, *which are both integral. The resulting optimality cut is*

$$417x_1 + 678x_2 - 15x_3 + 15x_4 \le 8211, \tag{2.22}$$

and the solution x^2 *to the master problem* MP^2 *is* $(0, 5, 0, 321.4)$. *Because* $x^2 \notin \mathbb{Z}_+^4$, *we add a Gomory cut* $27x_1 + 46x_2 - x_3 + x_4 \le 551$ *to the master problem and re-solve it using lexicographic dual simplex to obtain* $x^2 = (0, 5, 0, 321) \in \mathbb{Z}_+^4$. *The lower bound is updated as* $LB = -561$. *We do not update the upper bound because* $y^2(\omega_1) \notin \mathbb{Z}_+^5$.

Iteration 3. With $x^2 = (0, 5, 0, 321)$, *the optimal solution to subproblem* $SP^3(x^2, \omega_1)$ *is* $y^3(\omega_1) = (122.4, 0, 5, 0, 0.8) \notin \mathbb{Z}_+^5$. *We add a parametric Gomory cut*

$$3y_1(\omega_1) + 2y_2(\omega_1) + 3y_3(\omega_1) + 3y_4(\omega_1) \le 22 - 2x_2 \tag{2.23}$$

and obtain $y^3(\omega_1) = (120, 0, 6, 0, 0)$.*The optimal solution to subproblem* $SP^3(x^2, \omega_2)$ *and* $SP^3(x^2, \omega_3)$ *are* $y^3(\omega_2) = (171, 0, 9, 0, 0) \in \mathbb{Z}_+^5$ *and* $y^3(\omega_3) = (28, 0, 0, 0, 1) \in \mathbb{Z}_+^5$. *The corresponding optimality cut is*

$$69x_1 + 150x_2 - 3x_3 + 3x_4 \le 1707. \tag{2.24}$$

The optimal solution to the master problem MP^3 *is* $x^3 = (0, 4.5, 0, 344) \notin \mathbb{Z}_+^4$. *After adding two Gomory cuts* $26x_1 + 47x_2 - x_3 + x_4 \le 555$ *and* $25x_1 + 48x_2 - x_3 + x_4 \le 559$ *to the master problem. Re-solving the master problem with lexicographic dual simplex, we obtain an integer solution* $x^3 = (0, 5, 0, 319)$. *The lower bound is updated as* $LB = -559$ *and the upper bound is* $UB = -559$. *Therefore, we have found the optimal integer solution* $\bar{x} = (0, 5)$ *and* $y = ((120, 0, 6, 0, 0), (171, 0, 9, 0, 0), (28, 0, 0, 0, 1))$.

3. Finite Convergence. We define $X^k := \{x : A^k x \le b^k, x \in \mathbb{Z}_+^{n_1+2}\}$ as the set of integer points in the feasible region of master problem MP^k at iteration k.

LEMMA 3.1. $\{X^k\}_{k=1,\dots}$ *are all bounded, and* $X^1 \supseteq X^2 \supseteq \cdots \supseteq X^k \supseteq \cdots$.

Proof. For a given $\bar{x} \in \bar{X}$, because of Assumption (A3) that $|f(\bar{x}, \omega)| < +\infty$, there exists an integer $M(\bar{x}) \ge \sum_{\omega \in \Omega} p_\omega f(\bar{x}, \omega)$. In addition, $x_{n_1+1} - x_{n_1+2} \le \sum_{\omega \in \Omega} p_\omega f(\bar{x}, \omega)$. Therefore, $M(\bar{x}) \ge x_{n_1+1} - x_{n_1+2}$ for a given $\bar{x} \in \bar{X}$. Recall, from Assumption (A2), that $\bar{X}$ is bounded. Then we have $M := \max_{\bar{x} \in \bar{X}} M(\bar{x}) \ge x_{n_1+1} - x_{n_1+2}$.

Let $A^k_{n_1+1}$ and $A^k_{n_1+2}$ denote the coefficient columns for x_{n_1+1} and x_{n_1+2}. Because all optimality cuts added are in the format of inequality (2.16), then $A^k_{n_1+1} = -A^k_{n_2+1}$. Therefore at least one of x_{n_1+1} and x_{n_1+2} is non-basic. Hence, $M \ge x_{n_1+1}$, $M \ge x_{n_1+2}$, and $\bar{X} \times [0, M] \times [0, M] \supseteq X^k$ for $k = 1, \dots$, i.e., $\{X^k\}_{k=1,\dots}$ are all bounded.

In addition, $X^1 \supseteq X^2 \supseteq \cdots \supseteq X^k \supseteq \cdots$ is obvious because the master problem MP^k includes all the constraints in master problems MP^l for $1 \leq l < k$. Hence, $\bar{X} \times [0, M] \times [0, M] \supseteq X^1 \supseteq X^2 \supseteq \cdots \supseteq X^k \supseteq \cdots$. □

THEOREM 3.2. *Suppose that assumptions (A1)-(A5) are satisfied, then Algorithm 1 finds an optimal solution to* (1.1)-(1.6) *in finitely many iterations.*

Proof. In iteration k, where $k = 0 \ mod\ q$, we solve the master problem MP^k with a Gomory cutting plane algorithm in a finite number of steps to obtain $x^k \in X \subset \mathbb{Z}_+^{n_1+2}$. In iteration $k+1$, there are three cases to consider.

(i) Suppose that $y^{k+1}(\omega) \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}$ for all $\omega \in \Omega$, and $x^{k+1} = x^k$. Then the solution $(\bar{x}^{k+1}, \{y^{k+1}(\omega)\}_{\omega\in\Omega})$ must be the optimal solution to (1.1)-(1.6) because $x^{k+1}_{n_1+1} - x^{k+1}_{n_1+2} \geq \sum_{\omega\in\Omega} p_\omega f^{k+1}(x^{k+1}, \omega)$, i.e., $LB = \bar{c}^\top \bar{x}^{k+1} + x^{k+1}_{n_1+1} - x^{k+1}_{n_1+2} = UB = \bar{c}^\top \bar{x}^k + \sum_{\omega\in\Omega} p_\omega f^{k+1}(x^{k+1}, \omega)$.

(ii) Suppose that $x^{k+1} \neq x^k$. It is impossible that x^{k+1} and x^k are alternative optimal solutions to MP^{k+1} because lexicographic dual simplex method ensures that the optimal solution obtained is always the lexicographically smallest one. Therefore, x^k must violate the optimality cut generated in iteration $k+1$, so x^k will not be visited again in a future iteration.

(iii) Suppose that $y^{k+1}(\omega) \notin \mathbb{Z} \times \mathbb{Z}_+^{n_2}$ for some $\omega \in \Omega$, and $x^{k+1} = x^k$. Define $X := X^1$, which contains all feasible x. Note that for a given $(x, \omega) \in X \times \Omega$, because of Assumptions (A3) and (A4), there exists an integer $K(x, \omega) < +\infty$ such that the optimal solution $y(x, \omega) \in Y(\bar{x}, \omega)$ can be found by Algorithm 2 in at most $K(x, \omega)$ iterations. This is essentially because of the finite convergence of the Gomory cutting plane method [8]. Therefore, it takes at most $\max_{\omega\in\Omega} K(x^k, \omega)$ iterations to add parametric Gomory cuts to the second-stage subproblems to obtain an integral second-stage solution. Let $\ell \in [0, \max_{\omega\in\Omega} K(x^k, \omega)]$ be the largest number such that $y^{k+2}, \ldots, y^{k+\ell}$ are all fractional, and $x^{k+2}, \ldots, x^{k+\ell}$ are all equal to x^k. Then in iteration $k + \ell + 1$,

- if $y^{k+\ell+1} \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}$ and $x^{k+\ell+1} = x^k$, then we have found an optimal solution (the same argument as in case (i));
- if $x^{k+\ell+1} \neq x^k$, then x^k will not be visited again in a future iteration (the same argument as in case (ii));

Hence, any $x \in X$ will be visited in at most a finite number of consecutive iterations. In every q-th iteration, an integral first-stage solution can be found by Algorithm 1 with finitely many Gomory cuts [8]. Because X is bounded, there are finitely many $x \in X$. Also, Ω is finite as stated in Assumption (A5). Thus, there are finitely many $(x, \omega) \in X \times \Omega$. In the worst case, there exists an integer $K = \sum_{x\in X} \max_{\omega\in\Omega} K(x, \omega)$ such that either Algorithm 1 terminates in $k < qK$ iterations or $y^k(x, \omega) \in \mathbb{Z} \times \mathbb{Z}_+^{n_2}$ and $f^k(x^k, \omega) = f(x^k, \omega)$ for all $k \geq qK$. Then the convergence follows from the convergence of the Benders' decomposition method. □

4. A Branch-and-Cut Based Decomposition Algorithm. Exploiting the finite convergence of the branch-and-bound [10] and branch-and-cut methods, we develop an alternative branch-and-cut based decomposition algorithm with a breadthfirst strategy. For ease of exposition, we first describe the branch-and-bound (B&B) implementation. Let FP^t denote the master problem at the tth node of the B&B tree. This problem is the same as that defined in Section 2.2, but in which constraints (2.15) include the bounds on the variables introduced during the B&B process, and constraints (2.16) only include the Gomory cuts generated that are valid for node t of the branch-and-bound tree. Let $\mathrm{FP}^0 = \mathrm{MP}^0$. Let $\mathcal{L}$ be a collection of FP^t problems for all leaf nodes, t, in the B&B process, and LB^t and UB^t be the corresponding lower and upper bounds, respectively. Initially, $\mathcal{L}$ contains problem FP^0 with upper bound $UB^0 = +\infty$ and lower bound $LB^0 = -\infty$. Let $T^* = -\infty$ be the objective function value of the incumbent solution. In addition, we denote $\mathrm{CP}^t(x, \omega)$ as the subproblems defined in Section 2.2, but in which constraint (2.17) only includes the parametric Gomory cuts generated that are valid for node t of the branch-and-bound tree. We first solve problem FP^0 to obtain solution x^0. Let $k = 1$. In iteration k, if list $\mathcal{L} \neq \emptyset$, then let $j(k)$ be the smallest index among the problems in the list $\mathcal{L}$.

- If the lower bound $LB^{j(k)} > T^*$, then we prune node $j(k)$ because it is impossible to obtain a better integer solution branching from this node.
- Otherwise:
  - If the solution $x^{j(k)}$ is fractional, then we branch on the first fractional component of $x^{j(k)}$ at node $j(k)$ of the B&B tree to obtain the problems FP^{2k-1} and FP^{2k}. We solve these problems by lexicographic dual simplex method to obtain x^{2k-1} and x^{2k}, respectively. Update $LB^{2k-1} = c^\top x^{2k-1}$ and $LB^{2k} = c^\top x^{2k}$. Let problems FP^{2k-1} and FP^{2k} substitute problem $\mathrm{FP}^{j(k)}$ in the list $\mathcal{L}$. Also let the second-stage problems $\mathrm{CP}^{2k-1}(x, \omega)$ and $\mathrm{CP}^{2k}(x, \omega)$ be the same as $\mathrm{CP}^{j(k)}(x, \omega)$, and $UB^{2k-1} = UB^{2k} = +\infty$.
  - If the solution $x^{j(k)}$ is integral, then we solve problem $\mathrm{CP}^{j(k)}(x^{j(k)}, \omega)$ for all $\omega \in \Omega$ by lexicographic dual simplex method. For each $\omega \in \Omega$, if $y^{j(k)}(\omega)$, the solution to $\mathrm{CP}^{j(k)}(x^{j(k)}, \omega)$, is fractional, then we add a parametric Gomory cut to problem $\mathrm{CP}^{j(k)}(x^{j(k)}, \omega)$ and re-solve $\mathrm{CP}^{j(k)}(x^{j(k)}, \omega)$ by lexicographic dual simplex method. If the solution $y^{j(k)}(\omega) \in \mathbb{Z}_+^{n_2+1}$ for all $\omega \in \Omega$, then we update $UB^{j(k)}$ by $\min\{UB^{j(k)}, \bar{c}^\top \bar{x}^{j(k)} + \sum_{\omega \in \Omega} p_\omega y_0^{j(k)}(\omega)\}$. If $UB^{j(k)} = LB^{j(k)}$ and $UB^{j(k)} < T^*$, then we let let $T^* = UB^{j(k)}$ and update incumbent solution $(\bar{x}^*, \{y^*(\omega)\}_{\omega \in \Omega})$ by $(\bar{x}^{j(k)}, \{y^{j(k)}(\omega)\}_{\omega \in \Omega})$. In addition, we add the optimality cut obtained from $\mathrm{CP}^{j(k)}(x^{j(k)}, \omega)$ to the problem $\mathrm{FP}^{j(k)}$, update problem $\mathrm{FP}^{j(k)}$ in list $\mathcal{L}$, and solve problem $\mathrm{FP}^{j(k)}$ by lexicographic dual simplex method to obtain a new $x^{j(k)}$. Let $LB^{j(k)} = c^\top x^{j(k)}$.

Once the list $\mathcal{L}$ becomes empty, we output the incumbent solution $(\bar{x}^*, \{y^*(\omega)\}_{\omega \in \Omega})$ as the optimal solution and T^* as the optimal objective function value. The detailed algorithm is described by Algorithm 3. Note that we can have an alternative implementation, where we add any valid inequalities (Gomory or any other class) when solving the master problem, which we refer to as the branch-and-cut based decomposition algorithm.

Algorithm 3: Branch-and-bound based algorithm for solving SIP

List $\mathcal{L} = \{\text{FP}^0\}$, $k = 1$, $T^* = -\infty$, $UB^0 = +\infty$, $LB^0 = -\infty$, $\text{CP}^0(x, \omega) \leftarrow \text{CP}(x, \omega)$ for $\omega \in \Omega$;
Solve problem FP^0 by lexicographic simplex method to obtain x^0;
while $\mathcal{L} \neq \emptyset$ **do**
    Let $j(k) = \min\{j : \text{FP}^j \in \mathcal{L}\}$;
    if $LB^{j(k)} > T^*$ **then**
        $\mathcal{L} \leftarrow \mathcal{L} \setminus \{\text{FP}^{j(k)}\}$;
    else if $LB^{j(k)} < T^*$ *and* $x^{j(k)} \notin \mathbb{Z}_+^{n_1+2}$ **then**
        Branch on $x_{i(k)}^{j(k)}$ with $i(k) = \min\{i : x_i^{j(k)} \notin \mathbb{Z}_+, i \in [1, n_1 + 2]\}$ to obtain FP^{2k-1}, FP^{2k};
        Solve FP^{2k-1} and FP^{2k} by lexicographic simplex method to obtain x^{2k-1} and x^{2k-1}, respectively;
        $LB^{2k-1} = c^\top x^{2k-1}$, $LB^{2k} = c^\top x^{2k}$;
        For $\omega \in \Omega$, $\text{CP}^{2k-1}(x, \omega) \leftarrow \text{CP}^{j(k)}(x, \omega), \text{CP}^{2k}(x, \omega) \leftarrow \text{CP}^{j(k)}(x, \omega)$;
        $UB^{2k-1} = UB^{2k} = +\infty$;
        $\mathcal{L} \leftarrow \mathcal{L} \cup \{\text{FP}^{2k-1}, \text{FP}^{2k}\} \setminus \{\text{FP}^{j(k)}\}$;
    else
        For each $\omega \in \Omega$, solve $\text{CP}^{j(k)}(x^{j(k)}, \omega)$ by lexicographic simplex method to obtain $y^{j(k)}(\omega)$;
        if $y^{j(k)}(\omega) \notin \mathbb{Z}_+^{n_2+1}$ *for some* $\omega \in \Omega$ **then**
            For every $\omega \in \Omega$ such that $y^{j(k)}(\omega) \notin \mathbb{Z}_+^{n_2}$, generate a parametric Gomory cut for $y_{u(k)}^{2k-l}(\omega)$ with $u(k) = \min\{u : y_u^{2k-l}(\omega) \notin \mathbb{Z}_+, u \in [1, n_2]\}$;
            Add the parametric Gomory cut to $\text{CP}^{j(k)}(x, \omega)$, and solve $\text{CP}^{j(k)}(x^{j(k)}, \omega)$ by lexicographic dual simplex method to obtain a new $y^{j(k)}(\omega)$;
        end
        if $y^{j(k)}(\omega) \in \mathbb{Z}_+^{n_2+1}$ **then**
            $UB^{j(k)} = \min\{UB^{j(k)}, \bar{c}^\top \bar{x}^{j(k)} + \sum_{\omega \in \Omega} p_\omega y_0^{j(k)}(\omega)\}$;
        if $UB^{j(k)} - LB^{j(k)} < \epsilon$ *and* $UB^{j(k)} < T^*$ **then**
            $(\bar{x}^*, \{y^*(\omega)\}_{\omega \in \Omega}) \leftarrow (\bar{x}^{j(k)}, \{y^{j(k)}(\omega)\}_{\omega \in \Omega})$, $T^* = UB^{j(k)}$,
            $\mathcal{L} \leftarrow \mathcal{L} \setminus \{FP^{j(k)}\}$;
        if $UB^{j(k)} - LB^{j(k)} < \epsilon$ *and* $UB^{j(k)} > T^*$ **then**
            $\mathcal{L} \leftarrow \mathcal{L} \setminus \{FP^{j(k)}\}$;
        if $UB^{j(k)} - LB^{j(k)} > \epsilon$ **then**
            Add optimality cut derived from $\text{CP}^{j(k)}(x^{j(k)}, \omega)$ to $\text{FP}^{j(k)}$, and update list $\mathcal{L}$ with the updated $\text{FP}^{j(k)}$;
            Solve $\text{FP}^{j(k)}$ by lexicographic dual simplex method to obtain $x^{j(k)}$;
            $LB^{j(k)} = c^\top x^{j(k)}$;
        end
    $k \leftarrow k + 1$;
end
Return $(\bar{x}^*, \{y^*(\omega)\}_{\omega \in \Omega})$ and T^*;

PROPOSITION 4.1. *Suppose that assumptions (A1)-(A5) are satisfied, then Algorithm 3 finds an optimal solution to* (1.1)-(1.6) *in finitely many iterations.*

Proof. There exists an integer $M(x) < +\infty$ such that the branch-and-bound algorithm takes $M(x)$ iterations to find x. Because there are finitely many $x \in X$, there exists an integer $M \leq \sum_{x \in X} M(x)$ such that the optimal first-stage solution can be found after M iterations. Suppose that the solution x^j to problem FP^j is integral. It takes finitely many iterations to find the optimal solution $y^j(\bar{x}, \omega) \in Y(\bar{x}, \omega)$ for each $\omega \in \Omega$ because of the finite convergence of Benders' method [3] and Gomory cutting plane algorithm. Therefore, Algorithm 3 finds an optimal solution to (1.1)-(1.6) in finitely many iterations. ∎

Note that Proposition 4.1 also holds for the branch-and-cut based decomposition algorithm using any valid inequalities for the master problem. Next we illustrate Algorithm 3 on Example 1.

EXAMPLE 1. *(Continued). To be concise, we do not demonstrate all the iterations in detail. Until the end of iteration 2 of Algorithm 1, we obtained integral x as we solve the master problems. Thus, in iteration 1 of Algorithm 3, we have*

$$FP^0 = \min\{-18x_1 - 48x_2 + x_3 - x_4 : (2.3), (2.4), (2.20), (2.22)\},$$

$$CP^0(x, \omega_1) = \min\{-y_0(\omega_1) : (2.5), (2.6), (2.7), (2.18), (2.21)\},$$

$$CP^0(x, \omega_2) = \min\{-y_0(\omega_2) : (2.8), (2.9), (2.10)\},$$

and

$$CP^0(x, \omega_3) = \min\{-y_0(\omega_3) : (2.11), (2.12), (2.13), (2.19)\}.$$

Note that for Algorithm 3 the superscripts are the indices of the B&B nodes. We start with $k = 1$.

Since $x^0 = (0, 5, 0, 321.4) \notin \mathbb{Z}_+^4$, we branch on the root node to generate two leaf nodes with two bounding inequalities $x_4 \leq 321$ and $x_4 \geq 322$. Therefore,

$$FP^1 = \min\{-18x_1 - 48x_2 + x_3 - x_4 : (2.3), (2.4), (2.20), (2.22), x_4 \leq 321\},$$

and

$$FP^2 = \min\{-18x_1 - 48x_2 + x_3 - x_4 : (2.3), (2.4), (2.20), (2.22), x_4 \geq 322\},$$

Solving problems FP^1 and FP^2 by lexicographic dual simplex method, we get $x^1 = (0.014, 5, 0, 321), x^2 = (0, 5, 0.6, 322), LB^1 = -561.26$ and $LB^2 = -561.4$. In addition, $CP^1(x, \omega)$ and $CP^2(x, \omega)$ are the same as $CP^0(x, \omega)$ for $\omega \in \Omega$, and $UB^1 = UB^2 = +\infty$.

Iteration 2. Note that $k = 2$. Because $j(k) = 1$ and $x_1^1 \notin \mathbb{Z}_+$, we branch on node 1 to generate two new leaf nodes with two bounding inequalities $x_1 \leq 0$ and $x_1 \geq 1$ to *obtain the associated master problems FP^3 and FP^4. In particular,*

$$FP^3 = \min\{-18x_1 - 48x_2 + x_3 - x_4 : (2.3), (2.4), (2.20), (2.22), x_4 \le 321, x_1 \le 0\}.$$

Solving problem FP^3 by lexicographic dual simplex method, we get $x^3 = (0, 5, 0, 321)$ and $LB^3 = -561$. In addition, $CP^3(x, \omega)$ is the same as $CP^1(x, \omega)$ for $\omega \in \Omega$, and $UB^3 = +\infty$.

Iteration 3. Note that $k = 3$. Because $j(k) = 2$ and $x_3^2 \notin \mathbb{Z}_+$, we branch on node 2 to generate two new leaf nodes with two bounding inequalities $x_3 \le 0$ and $x_3 \ge 1$ to obtain the associated master problems FP^5 and FP^6.

Iteration 4. Note that $k = 4$. Because $j(k) = 3$ and $x^3 \in \mathbb{Z}_+^4$, we solve $CP^3(\omega)$ for each $\omega \in \Omega$. The parametric Gomory cut added for $CP^3(x, \omega_1)$ is given by inequality (2.23), and the optimality cuts generated for FP^3 is given by (2.24). $UB^3 = -559$ because $y^3(\omega) \in \mathbb{Z}_+^5$ for each $\omega \in \Omega$. Also, because $x^3 = (0, 4.96, 0, 321) \notin \mathbb{Z}_+^4$, we branch on node 3 to obtain the associated master problems FP^7 and FP^8. In particular,

$$FP^8 = \min\{-18x_1 - 48x_2 + x_3 - x_4 : \\ (2.3), (2.4), (2.20), (2.22), (2.24), x_4 \le 321, x_1 \le 0, x_2 \ge 5\}.$$

Solving problem FP^8 by lexicographic dual simplex method, we get $x^8 = (0, 5, 0, 319)$ and and $LB^8 = -559$. This gives an incumbent solution $\bar{x} = (0, 5)$ and $y = ((120, 0, 6, 0, 0),\ (171, 0, 9, 0, 0),\ (28, 0, 0, 0, 1))$ with $T^* = -559$. After exploring the remaining leaf nodes, it is shown that this incumbent solution is optimal.

We illustrated a branch-and-bound based decomposition algorithm in this example. If, instead, Gomory cuts are added while solving the master problem in a *branch-and-cut* based algorithm, then we obtain the optimal solution in fewer iterations.

5. Preliminary Computational Results. To demonstrate the performance of the proposed decomposition algorithms on larger sized instances, we test them on Example 1 with $m = 5, 10, 20, 50, 100$, and 200. For each m, five random instances are generated, and we report the average performance. The second-stage cost function $g(\tilde{\omega})$, technology matrix $\bar{T}(\tilde{\omega})$, recourse matrix $W(\tilde{\omega})$, and right-hand-side vector $r(\tilde{\omega})$ are generated using discrete distributions on intervals:

- $g_1(\tilde{\omega}) \in [-19, -16]$, $g_2(\tilde{\omega}) \in [-22, -19]$, $g_3(\tilde{\omega}) \in [-26, -23]$, $g_4(\tilde{\omega}) \in [-31, -28]$
- $\bar{T}_{11}(\tilde{\omega}) \in [1, 4]$, $\bar{T}_{22}(\tilde{\omega}) \in [1, 4]$, $\bar{T}_{12}(\tilde{\omega}) = \bar{T}_{21}(\tilde{\omega}) = 0$
- $W_{11}(\tilde{\omega}) \in [2, 5]$, $W_{12}(\tilde{\omega}) \in [3, 6]$, $W_{13}(\tilde{\omega}) \in [4, 7]$, $W_{14}(\tilde{\omega}) \in [5, 8]$, $W_{21}(\tilde{\omega}) \in [6, 9]$, $W_{22}(\tilde{\omega}) \in [1, 4]$, $W_{23}(\tilde{\omega}) \in [3, 6]$, $W_{24}(\tilde{\omega}) \in [2, 5]$
- $\bar{r}_i(\tilde{\omega}) \in [20, 29]$, $i \in [1, 2]$.

We first solve our instances by Algorithm 1 (denoted by DG) and branch-and-cut based decomposition Algorithm 3 (denoted by BCDG). We restrict that at most 50 Gomory cuts can be added to the master problem in each iteration with algorithm DG. In algorithm BCDG, we allow to add at most 10 Gomory cuts when

$(x_1,\ldots,x_{n_1}) \in \mathbb{Z}_+^{n_1}$ and $(x_{n_1+1}, x_{n_1+2}) \notin \mathbb{Z}_+^2$. Also, for a particular $x \in \mathbb{Z}_+^{n_1+2}$, if there still exists $i \in [1,m]$ such that $y(\omega_i) \notin \mathbb{Z}_+^{n_2}$ after 20 consecutive iterations, then we call IBM ILOG CPLEX to solve the subproblems as IPs to find an integer solution $\{y(\omega)\}_{\omega\in\Omega}$ to update the upper bound. For sake of comparison, we solve the deterministic equivalent formulation by branch-and-cut method with Gomory cuts only (denoted by BCG), in which the lexicographic dual simplex method is employed to solve the linear relaxation after a cut is added. We allow at most five Gomory cuts to be added in each iteration of BCG. For these three algorithms, because of the limited flexibility of customizing the solution process in a commercial optimization software, such as CPLEX, we implement lexicographic dual simplex method, branch-and-bound process, Gomory cuts generation on our own (with C++ language) instead of calling any external solvers. The only time we use CPLEX is in Algorithm BCDG to obtain upper bound. We run our codes on 1-GHz dual core AMD Opteron(tm) processor 1218 with 2 GB RAM. In addition, we also solve the DEF of these instances by IBM ILOG CPLEX 12.0 with two alternative settings: (i) default CPLEX setting with only Gomory cuts (denoted by CPLEX), (ii) default CPLEX setting (denoted by CPLEX-D). A time limit of 1 hour is imposed.

In Table 5.1, we summarize the performance of algorithm DG with $m = 5$, algorithm BCG with $m \in \{5, 10, 20\}$, and algorithms BCDG, CPLEX, and CPLEX-D with $m \in \{5, 10, 20, 50, 100, 200\}$. Column **D. itrtn** reports the number of optimality cuts returned in the decomposition scheme, and column **S. itrtn** reports the number of times the lexicographic dual simplex method is called. Column **Cuts** reports the number of Gomory cuts added. For Algorithms DG and BCDG, the first and second numbers in the parenthesis are the numbers of Gomory cuts added to the master problem and subproblems, respectively. Column **B-B nodes** reports the number of B&B tree nodes explored. Column **Gap** shows the gap between the best lower bound and the optimal objective function value, and column **Time** reports the solution time.The number in parenthesis in this column is the number of instances that are solved to optimality within time limit. In addition, in column **CPLEX#**, we report the number of instances for which CPLEX is called in Algorithm BCDG for solving the subproblems to integer optimality to obtain an upper bound.

From our computational experience summarized in Table 5.1, we observe that algorithm DG cannot solve the instances with $m \geq 5$ effectively. Because of the computer memory limit and numerical issues, algorithm DG is forced to stop before the time limit is reached. In row DG*, only 2 out of 5 instances are solved successfully, so the computational results reported are based on these instances only. Similarly, algorithm BCG cannot solve the instances with $m \geq 20$ efficiently. The computational results in row BCG* are based on 2 out of 5 instances.

Comparing Algorithms BCDG with DG and BCG, BCDG takes fewer decomposition iterations and lexicographic dual simplex iterations, explores fewer B&B tree nodes, and solves our test problems more efficiently. Comparing Algorithm BCDG with solving the DEF using default CPLEX, BCDG explores significantly fewer B&B tree nodes and solve the instances a lot faster. Despite the disadvantage of not having utilized the state-of-the-art linear programming solver of CPLEX, our in-house implementation of BCDG already outperforms the branch-and-cut method for the DEF

TABLE 5.1
Computational results on Example 1 with varying number of scenarios

m	Algorithm	D. itrtn	S. itrtn	Cuts	B-B nodes	Gap(%)	Time(sec)	CPLEX#
5	DG*	24.5	141.5	(53, 51)	0	0	0.97	-
	BCDG	21.2	224	(48.4, 43.2)	8.2	0	0.47	0
	BCG	0	175.8	235	51.4	0	0.25	-
	CPLEX	0	-	5.2	10.4	0	0.04	-
	CPLEX-D	0	-	5	6.8	0	0.06	-
10	BCDG	40.8	697.2	(88, 145.2)	18.2	0	1.05	1
	BCG	0	2870.8	3769	762.2	0	10.09	-
	CPLEX	0	-	8	890	0	0.24	-
	CPLEX-D	0	-	20.4	132.8	0	0.16	-
20	BCDG	46.4	1295.8	(66.2, 244.8)	13.8	0	1.58	0
	BCG*	0	2551	3850	576	0	12.75	-
	CPLEX	0	-	16.4	2401.2	0	0.46	-
	CPLEX-D	0	-	34.8	59.8	0	0.14	-
50	BCDG	73.8	4629.8	(95.6, 756.8)	17.4	0	5.01	3
	CPLEX	0	-	38.8	10757816	0.58	1323.76(1)	-
	CPLEX-D	0	-	116.2	33344	0	10.39	-
100	BCDG	78.4	9682.2	(104, 1643)	20.2	0	11.54	5
	CPLEX	0	-	72.8	13067815	1.04	≥3600(5)	-
	CPLEX-D	0	-	292.4	1707425	0.02	38.20(1)	-
200	BCDG	66.6	16178.2	(102.8, 2674.6)	17	0	22.32	5
	CPLEX	0	-	112.8	7915714	1.23	≥3600(5)	-
	CPLEX-D	0	-	613	1918403	0.23	≥3600(5)	-

employed by CPLEX default for problems with a moderate number of scenarios. In addition, we observe that the increase in the solution times is linear with respect to the number of scenarios using BCDG, whereas it is exponential using the DEF. We can conclude that the decomposition method significantly reduces the computational burden, and the branch-and-bound and Gomory cuts help expedite the computational process.

6. Conclusion. We study a class of two-stage stochastic pure integer programs with general integer variables in both stages (SIP). We consider a very general class of problems, where the cost function of the second-stage decision variables, technology and recourse matrices, and the right-hand-side of the constraints could be affected by random parameters. We assume that the random parameters have finite support. Instead of solving the large-size deterministic equivalent of the two-stage SIP, we propose a decomposition algorithm based on Benders' method to solve the second-stage problem for each scenario separately, and return an approximation of the second-stage cost function to the first-stage problem. Our method generates Gomory cuts parameterized with respect to the first-stage decision variables, i.e., they are valid for the deterministic equivalent. We also propose an alternative algorithm that implements Benders' decomposition method in the branch-and-bound process. We prove that the optimal solution can be found within finitely many iterations. Our results with a preliminary implementation of our algorithms are very encouraging. As part of our future work, we plan to develop a more robust implementation of our algorithms to solve a larger class of SIPs.

Acknowledgment

We would like to thank Suvrajeet Sen for the beneficial discussions.

REFERENCES

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

[2] E. Balas, S. Ceria, and G. Cornuéjols, *A lift-and-project cutting plane algorithm for mixed 0c1 programs*, Mathematical Programming, 58 (1993), pp. 295–324.

[3] J. Benders, *Partitioning procedures for solving mixed-variables programming problems*, 4 (1962), pp. 238–252.

[4] C. E. Blair and R. G. Jeroslow, *The value function of an integer program*, Mathematical Programming, 23 (1982), pp. 237–273.

[5] C. C. Carøe and J. Tind, *A cutting-plane approach to mixed 0-1 stochastic integer programs*, European Journal of Operational Research, 101 (1997), pp. 306–316.

[6] ———, *L-shaped decomposition of two-stage stochastic programs with integer recouse*, Mathematical Programming, 101 (1998), pp. 306–316.

[7] D. Gade, S. Küçükyavuz, and S. Sen, *Decomposition algorithms with parametric gomory cuts for two-stage stochastic integer programs*, Mathematical Programming, (2013).

[8] R. E. Gomory, *Outline of an algorithm for integer solutions to linear programs*, Bulletin of the American Mathematical Society, 64 (1958), pp. 275–278.

[9] N. Kong, A. J. Schaefer, and B. Hunsaker, *Two-stage integer programs with stochastic right-hand sides: a superadditive dual formulation*, Mathematical Programming, 108 (2006), pp. 275–296.

[10] A. Land and A. Doig, *An automatic method of solving discrete programming problems*, Econometrica: Journal of the Econometric Society, (1960).

[11] G. Laporte and F. V. Louveaux, *The integer L-shaped method for stochastic integer programs with complete recourse*, Operations Research Letters, 13 (1993), pp. 133–142.

[12] R. Schultz, L. Stougie, and M. H. van der Vlerk, *Solving stochastic programs with integer recourse by enumeration: A framework using Gröbner basis*, Mathematical Programming, 183 (1998), pp. 229–252.

[13] S. Sen, *Algorithms for stochastic mixed-integer programming models*, in Handbook in Operations Research and Management Science: Discrete Optimization, K. Aardal, G. Nemhauser, and R. Weismantel, eds., Elsevier, 2005, pp. 515–558.

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

[15] S. Sen and H. D. Sherali, *On the convergence of cutting plane algorithms for a class of nonconvex mathematical programs*, Mathematical Programming, 106 (2006), pp. 203–223.

[16] H. D. Sherali and X. Zhu, *On solving discrete two-stage stochastic programs having mixed-integer first- and second- stage variables*, Mathematical Programming, 108 (2006), pp. 597–616.

[17] A. Trapp, O. A. Prokopyev, and A. J. Schaefer, *On a level-set characterization of the value function of an integer program and its application to stochastic programming*, Operations Research, 61 (2013), pp. 498–511.

[18] R. Van Slyke and R. Wets, *L-shaped linear programs with applications to optimal control and stochastic programming*, 17 (1969), pp. 638–663.

[19] S. W. Wallace and W. T. Ziemba, eds., *Applications of stochastic programming*, MPS-SIAM series on optimization, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005.