# An Introduction to Two-Stage Stochastic Mixed-Integer Programming

Simge Küçükyavuz

Department of Integrated Systems Engineering, The Ohio State University, `kucukyavuz.2@osu.edu`

Suvrajeet Sen

Department of Industrial and Systems Engineering, University of Southern California, `s.sen@usc.edu`

ABSTRACT: This paper provides an introduction to algorithms for two-stage stochastic mixed-integer programs. Our focus is on methods which decompose the problem by scenarios representing randomness in the problem data. The design of these algorithms depend on where the uncertainty appears (right-hand-side, recourse matrix and/or technology matrix) and where the continuous and discrete decision variables are (first-stage and/or second-stage). In addition we provide computational evidence that, similar to other classes of stochastic programming problems, decomposition methods can provide desirable theoretical properties (such as finite convergence) as well as enhanced computational performance when compared to solving a deterministic equivalent formulation using an advanced commercial MIP solver.

*Keywords*: integer programming, stochastic programming, chance constraints, cutting planes, disjunctions, Benders

**1. Introduction** Over the years, mixed-integer programming (MIP) and stochastic programming (SP) have not only earned the reputation of addressing some of the more important applications of optimization, but they also represent two of the more challenging classes of optimization models. The discrete and non-convex aspects of MIP, and the need to capture uncertainty via SP, raise serious conceptual and computational challenges. It is then no surprise that a combination of these two classes of models, namely stochastic mixed-integer programming (SMIP), engenders even further hurdles. Nevertheless, success with SMIP models is likely to pay tremendous dividends because of their vast applicability. Applications of SMIP models have appeared in a variety of domains such as planning electricity contracts [56], electric grid operations [58], location under uncertainty [67], capacity expansion for manufactuting [4], military aircraft operations [30], process engineering [33], supply chain design [65] and many more. Because of this growing array of applications, there is also growing interest in models and algorithms for SMIP. We hope that this introductory tutorial will help provide the foundations for further growth of the area. In what follows, we assume that the reader has some basic knowledge in integer programming and stochastic programming. We refer the reader to [66] and [8] for an introduction to these topics. Furthermore, the reader should note that we have omitted some SMIP ideas that have appeared in earlier surveys on this subject ([53], [52]).

There are several alternative approaches to making decisions under uncertainty: in some cases, one may be interested in decisions for which the long-run cost of operations is optimized, while in some others, the operations must not only optimize long-run costs, but may also need to ensure that a pre-specified reliability level is met. The first of these leads to the so-called Stochastic Mixed-Integer Program with Recourse (SMIP-RC) model, and the second type leads to Stochastic Mixed-Integer Program with Recourse and Chance-Constraints (SMIP-RCC). This tutorial will cover these two classes of models in that order.

**2. Stochastic Mixed-Integer Programming with Recourse** For a two-stage SMIP model, there are two decision epochs that are separated by an intervening probabilistic event, which is modeled using a random vector $\tilde{\omega}$. This random vector is defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ and $\mathbb{E}[\cdot]$ denotes the usual mathematical expectation operator taken with respect to $\tilde{\omega}$.

A standard mathematical formulation for two-stage SMIP with Recourse (SMIP-RC) is represented

as follows.

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

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

$$h(x, \omega) = \min \quad y_0 \tag{2a}$$
$$y_0 - g(\omega)^\top y = 0 \tag{2b}$$
$$W(\omega)y \geq r(\omega) - T(\omega)x \tag{2c}$$
$$y \in \mathcal{Y}. \tag{2d}$$

In the interest of brevity, we occasionally use $X = \{x | Ax \geq b\}$ and $Y(x, \omega) = \{y|(2b) - (2d)\}$. The sets $\mathcal{X} \subseteq \mathbb{R}_+^{n_1}$ and $\mathcal{Y} \subseteq \mathbb{R} \times \mathbb{R}_+^{n_2}$ impose non-negativity, and discrete, binary or continuous restrictions on the first and second-stage variables, respectively. 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 $r(\omega) \in \mathbb{Q}^{m_2(\omega)}$ for $\omega \in \Omega$ are given data. Also note that the set $X$ does not include the integrality restrictions $\mathcal{X}$, whereas the set $Y(x, \omega)$ includes the integrality restrictions $\mathcal{Y}$.

The two-stage formulation (1)–(2) has an equivalent monolithic representation – the so-called deterministic equivalent formulation (DEF) obtained by creating copies of the second-stage variables for each scenario – given by

$$
\begin{array}{llllr}
\min \quad c^\top x & +p(1)g(1)^\top y(1) + p(2)g(2)^\top y(2) & + \cdots + & p(|\Omega|)g(|\Omega|)^\top y(|\Omega|)) & \text{(3a)} \\
Ax & & & \geq b & \text{(3b)} \\
T(1)x & +W(1)y(1) & & \geq r(1) & \text{(3c)} \\
T(2)x & \quad + W(2)y(2) & & \geq r(2) & \text{(3d)} \\
\vdots & & \ddots & \vdots & \text{(3e)} \\
T(|\Omega|)x & & & +W(|\Omega|)y(|\Omega|) \geq r(|\Omega|) & \text{(3f)} \\
\end{array}
$$
$$x \in \mathcal{X}, y(\omega) \in \mathcal{Y}, \quad \omega \in \Omega, \tag{3g}$$

where $y(\omega)$ represents the second-stage decisions under scenario $\omega \in \Omega$ and $p(\omega)$ is the probability of occurrence for $\omega \in \Omega$. While this is a very-large scale mixed-integer program that challenges even the state-of-the-art optimization solvers, it has a nice structure. Once we fix the first-stage variables $x$, the problem decomposes into $|\Omega|$ subproblems that are small in comparison and that can be solved independently. The algorithms we will describe exploit this structure. Before getting into the details of SMIP algorithms, it is worthwhile to give the reader a concrete sense of the types of situations which lead to two-stage SMIP models.

EXAMPLE 2.1 (STOCHASTIC SERVER LOCATION AND SIZING PROBLEMS) Consider a situation in which we are planning for new facilities prior to having firm data on where clients may materialize. Such situations arise in preparing for deployment of new services (e.g., communications, transportation, military, emergency response, etc.) without exact knowledge of location and size of demands for the services. A client-server paradigm captures a variety of these applications where the goal is to locate servers (and perhaps, size them) in such a way that the resulting system is most "cost-effective". Because precise location and size of demands are unknown at the time of planning, deterministic optimization of the size and location of servers may be impossible. However, one may be able to create a potential list of scenarios either based on educated guesses (by domain experts, say), or from prior experience. Such approaches are common in planning for natural disaster mitigation, and similar applications mentioned above. To state such a model in a mathematical framework, let $I$ be the index set for client locations and $J$ be the index set for possible server locations. For $i \in I$ and $j \in J$, we define the following.

Parameters

$c_j$: Cost of locating a server at location $j$.

$g_{ij}$: Revenue from a client at location $i$ being served by servers at location $j$.

$g_{j0}$: Loss of revenue because of overflow at server $j$.

$d_{ij}$: Resource requirement of client $i$ for server at location $j$.

$r$: Upper bound on the total number of servers that can be located.

$u$: Upper bound of number of servers located at any one location.

$w$: Server capacity.

$v$: Upper bound of possible number of clients.

$N^i(\omega)$: The number of clients at location $i$ in scenario $\omega$.

Decisions variables

$x_j$: number of servers located at site $j$.

$y_{ij}$: number of clients at location $i$ served by servers at location $j$.

$y_{j0}$: overflow amount at server location $j$.

The stochastic server location and sizing Problem (SSLS) model can be formulated as follows.

$$\min \quad \sum_{j \in J} c_j x_j + \mathbb{E}[h(x, \tilde{\omega})] \tag{4a}$$

$$\sum_{j \in J} x_j \leq r, \tag{4b}$$

$$x_j \in \{0, 1, \ldots, u\}, \quad \forall j \in J, \tag{4c}$$

where $\mathbb{E}[h(x, \tilde{\omega})] = \sum_{\omega \in \Omega} p(\omega) h(x, \omega)$, and for any $x$ satisfying (4b)-(4c) and $\omega \in \Omega$, we have

$$h(x, \omega) = \min \sum_{j \in J} g_{j0} y_{j0} - \sum_{i \in I} \sum_{j \in J} g_{ij} y_{ij} \tag{5a}$$

$$\sum_{i \in I} d_{ij} y_{ij} - y_{j0} \leq w x_j, \quad \forall j \in J, \tag{5b}$$

$$\sum_{j \in J} y_{ij} = N^i(\omega), \quad \forall i \in I, \tag{5c}$$

$$y_{ij} \in \{0, 1, 2, \ldots, v\}, \forall i \in I, j \in J, \tag{5d}$$

$$y_{j0} \geq 0, \quad \forall j \in J. \tag{5e}$$

When the parameters $u, v = 1$, the model is referred to as the stochastic server location problem (SSLP) [2]; otherwise, it is known as the stochastic server location and sizing problem (SSLS) [44]. However, we note that these models can also include some zonal service restrictions as explained in [40]. As with many MIP algorithms, the decomposition schemes that we present are greatly influenced by whether the integer variables are assumed to be binary, or more general integers. Consequently, the SSLP model (which has only binary variables in both stages) should be solved with a more specialized algorithm than the SSLS model which allows general integers in both stages. A classification of SMIP models based on these characteristics emphasizes such distinctions, and helps ascertain which algorithms may be more appropriate for a particular situation. Furthermore, we should also note that the methods described here may not be as appropriate for instances in which the second stage objective only reflects the need for integer feasibility. In such cases, the recourse function for each scenario has the form of an indicator function (i.e., zero when the first stage choice $x$ leads to a feasible second stage MIP and infinity otherwise), and as a result, such problems are better addressed using algorithms with greater focus on integer feasibility. The algorithms presented below are better suited for recourse functions which represent finite costs for all choices of the first stage decisions $x$.

Because (1)–(2) is more demanding than either stochastic LPs (SLP), or deterministic MIPs, algorithms which are intended to solve SMIP-RC models attempt to take advantage of specific structures which arise in applications. One way to characterize special structures is by examining the structure of the recourse matrix $W$. For instance, the Simple Integer Recourse (SIR) model arises when the matrix $W = [I, -I]$ (as in continuous simple recourse models), while requiring the second stage recourse decisions ($y$) to satisfy integrality. Similarly, a model in which $W$ is Totally Unimodular (TU) results in a TU Recourse (TUR) model. Note that these special matrices are also assumed to be fixed, whereas, the general statement in (2) allows random matrices $(W(\omega), T(\omega))$ in the second stage. Fixed recourse matrices are often more amenable for computational methods, especially those based on Benders decomposition. In some special instances, it may be possible to reformulate problems with random recourse to ones with fixed recourse as in the following example.

EXAMPLE 2.2 (REFORMULATION OF RANDOM RECOURSE TO FIXED RECOURSE) Consider a network whose links are failure prone. When the link fails, there can be no flow through it; on the other hand, when the link is up, the flow is allowed to be in the range $[0, u]$, where $u > 0$ denotes the link capacity. In this example, link failures can be modeled in two ways: one in which the link itself is removed from the network topology during failure, and another in which the capacity of the link can be treated as a random variable assuming values $\{0, u\}$, while the incidence matrix of the network flow model remains the same. Note that for the second formulation the flow capacity appears as a random right hand side in the second stage. Thus if link failures were the only random features of the model, then the reformulation induces the fixed recourse property.

In addition to the structure of the recourse matrix, another feature which is important for the design (or choice) of algorithms involves the location of integer variables in the model. When integer variables appear only in the first stage (i.e., the second stage has only continuous decision variables), the resulting value function $h$ is piecewise linear and convex on its effective domain. As a result, the complicating (integer) variables only appear in the first stage, and therefore the model is amenable to classical Benders decomposition. On the other hand, classical Benders decomposition is no longer applicable when the second stage (recourse) model imposes integer restrictions. To summarize some of the main features of SMIP models, we introduce the following sets which help characterize the scope of models that an algorithm may be capable of addressing.

$$
\begin{aligned}
A \quad &= \text{Model Structure} = \{\text{SIR, TUR, RC, RCC}\} \\
B \quad &= \text{Stages with Binary decision variables} \\
C \quad &= \text{Stages with Continuous decision variables} \\
D \quad &= \text{Stages with Discrete (general integer) decision variables.}
\end{aligned}
$$

There is one final caveat that we impose: we will assume that the random variable $\tilde{\omega}$ is discrete, and has finitely many outcomes with known probabilities. For cases with continuous random variables, or those requiring sampling, we refer the reader to [39].

**3. Optimum-seeking Methods for Stochastic Mixed-Integer Programming with Recourse**
The world of SP and that of MIP have one recurring theme, that of Benders Decomposition [5]. In the SP literature, this form of decomposition came to be known under the umbrella of the so-called $L$-shaped method [62]. Most standard textbooks in MIP as well as SP provide extensive coverage of this method. Very briefly, the central theme of the method is to exploit the decomposability of the first- and second-stage problems, and reformulate the DEF (3) in the epigraphical form using $(x, \eta)$-variables as follows:

$$
\min_{x \in X \cap \mathcal{X}} \{c^\top x + \eta : \eta \geq \mathbb{E}[h(x, \tilde{\omega})]\}, \tag{6}
$$

where as before $X = \{x : Ax \geq b\}$, $\mathcal{X}$ includes the integer requirements, and $\eta$ is a variable capturing the second-stage objective function value.

**3.1 Decomposition for Two-Stage Stochastic Programs with Continuous Second-Stage Variables**   First, we consider the case that $B = D = \{1\}, C = \{1, 2\}$, in other words, the second-stage problems are linear programs and the only integer variables are in the first stage. For this case, ordinary Benders decomposition applies. We review the classical Benders decomposition method, to highlight the challenges once integer variables are introduced to the second-stage problems.

Note that for a given first stage vector $x$, formulation (3) decomposes into subproblems for each $\omega = 1, \dots, |\Omega|$ as

$$\eta_\omega(x) := \min \; g(\omega)^\top y(\omega) \qquad\qquad \text{Dual}$$
$$W(\omega)y(\omega) \geq r(\omega) - T(\omega)x \qquad (\psi_\omega)$$
$$y_\omega \geq 0,$$

which are parametric linear programs where the right-hand-side is parameterized by $x$. So we can reformulate the original problem (3) in the space of $x$ as

$$\min \; c^\top x + \sum_{\omega \in \Omega} p(\omega)\eta_\omega(x)$$
$$\text{s.t.} \quad Ax \geq b$$
$$x \in \mathcal{X}.$$

Note that for the subproblems for a given $x$ and $\omega$, the feasible region changes as a function of $x$, which may make it difficult to understand the behavior of $\eta_\omega(x)$. To overcome this, we take the dual of the subproblem:

$$\eta_\omega(x) = \max \; (r(\omega) - T(\omega)x)^\top \psi_\omega$$
$$W(\omega)^\top \psi_\omega \leq g(\omega)$$
$$\psi_\omega \geq 0.$$

Here, it is important to note that because the second-stage problem is a linear program, the dual is well-defined, and there is no duality gap, the primal and the dual have the same objective function value at an optimal solution. (This is the first key point that makes the classical Benders decomposition work for continuous recourse, which will fail when we introduce integer variables in the second stage.)

Now we observe that as we change $x$, only the objective function of the dual changes, the feasible region stays the same. Using this observation, let $\{\psi_\omega^i\}_{i \in I_\omega}$ be the set of extreme points of the polyhedron $\mathcal{D}(\omega) = \{\psi : W(\omega)^\top \psi_\omega \leq g(\omega), \psi_\omega \geq 0\}$. Let $\{w_\omega^j\}_{j \in J_\omega}$ be the set of extreme rays of the polyhedron $\mathcal{D}(\omega)$. Sets $I_\omega$ and $J_\omega$ are finite. For $x$ such that $\eta_\omega(x)$ is finite, we know that there exists an optimal solution to the dual that is an extreme point. Therefore, we can rewrite $\eta_\omega(x) = \max_{i \in I_\omega}\{(r(\omega) - T(\omega)x)^\top \psi_\omega^i\}$. From this expression, we can see that $\eta_\omega(x)$ is a piecewise linear convex function as shown in Figure 1. Thus, we obtain the Benders Reformulation

$$\min \; c^\top x + \eta$$
$$Ax \geq b \tag{7a}$$
$$\eta \geq \sum_{\omega \in \Omega} p(\omega)(r(\omega) - T(\omega)x)^\top \psi_\omega^i \qquad \forall i \in I_\omega \qquad \omega \in \Omega \tag{7b}$$
$$0 \geq (r(\omega) - T(\omega)x)^\top (w_\omega^j) \qquad\qquad \forall j \in J_\omega \qquad \omega \in \Omega \tag{7c}$$
$$x \in X \cap \mathcal{X}.$$

The constraint set (7b) is known as the *optimality cuts*, and the constraint set (7c) is known as the *feasibility cuts*. When relatively complete recourse is assumed, the second stage problems are feasible for any given $x \in X \cap \mathcal{X}$, hence the dual is bounded and $J_\omega = \emptyset$ for all $\omega \in \Omega$. Hence we arrive at the explicit form of the Benders reformulation (6), where the right-hand side of (7b) represents $\mathbb{E}[h(x, \tilde{\omega})]$. This formulation has exponentially many constraints, but much fewer variables than the original formulation because instead of the vectors $y_\omega, \omega \in \Omega$, we have a single variable $\eta$ that captures the expected value of
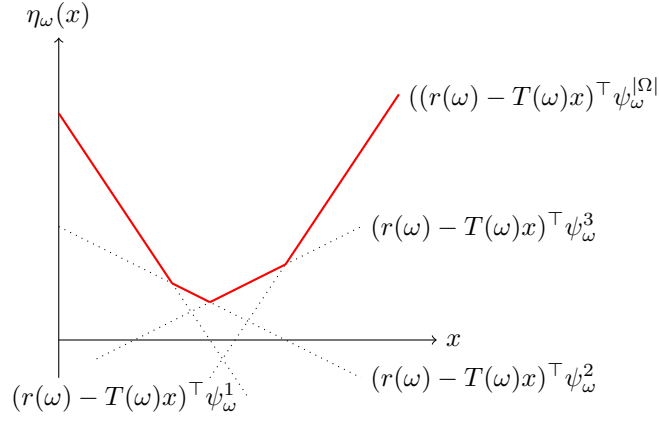
Figure 1: Piecewise-linear function, $\eta_\omega(x)$, for continuous recourse

the second-stage objective function. To solve this reformulation, we use delayed constrained generation, where we do not include all the constraints at once, but we add them as they are violated. Assuming that $J_\omega = \emptyset$ (as in relatively complete recourse), we first solve a relaxed version of (7) with a subset of inequalities (7b) (i.e., subset of dual extreme points, $I'_\omega \subset I_\omega$). This problem is referred to as the Relaxed Master Problem (RMP). We obtain a first-stage solution $x, \eta$ from this relaxation, we then solve the subproblems (or their duals) for this given $x$ to find a dual extreme point $\psi^i_\omega, i \in I_\omega \setminus I'_\omega$ for which the inequality (7b) is violated. If we find any such dual extreme point, then we expand the subset $I'_\omega$ with this extreme point. If not, then we are at an optimal solution to our problem. In summary, for continuous recourse, the expected recourse function is convex, and it can be approximated by a sequence of piecewise linear approximations and this process converges finitely to an optimal solution. This is the second key point that makes the Benders decomposition algorithm work effectively for continuous recourse problems. At each stage of the algorithm, we solve an RMP, which is a mixed-integer *linear* program, due to the affine constraints (7b).

**3.2 Decomposition for Two-Stage Stochastic Mixed-Integer Programs** In the presence of discrete variables in the second stage, the expected recourse function is no longer convex, and in general, it is not even continuous; it is only lower semicontinuous and in general non-convex [10, 50]. Thus, approximating $\mathbb{E}[h(x, \tilde{\omega})]$ becomes a challenge when the second-stage variables are integers (we can no longer use piecewise-linear approximations arising from the LP dual of the subproblems). By using MIP duality, Carøe and Tind [12] suggest a conceptual algorithm which mimics Benders decomposition using MIP price functions. Even if one were to recover MIP price functions using MIP duality, this procedure leads to highly non-convex master programs for each iteration. Given these hurdles, we are not very optimistic about the potential for its practical viability. One variant that has been used in some applications, referred to as the integer L-shaped method, is a combinatorial approach for models which satisfy $B = \{1, 2\}$, $C = \{2\}$, and $D = \{2\}$. Note that in this case, the first stage model has only binary decision variables, and for this structure, Laporte and Louveaux [31] suggest a combinatorially justifiable valid inequality (as a lower bound of the objective). These approximations are similar in spirit to "no-good" cuts suggested in the constraint programming literature. However, the performance of these combinatorial cuts on standard benchmark problems (e.g., stochastic server location problems (SSLP) in [2]) has not been particularly noteworthy. Nevertheless, we have found that these inequalities can be useful in avoiding "tailing" behavior common in many decomposition methods [41].

In the remainder of the section, we will describe two approaches: one based on parametric Gomory cuts [22, 68] which are similar in spirit to some other parametric cutting planes based on disjunctive programming (e.g., [54, 55]); the other approach uses a branch-and-cut based decomposition approach which we refer to as Ancestral Benders Cuts (ABC) [44]. The latter allows very general structures, including randomness in all data elements, as well as continuous and general integer decisions in both

stages. Of course, as instances become more general, the demands on the algorithm begin to increase. Nevertheless, Qi and Sen [44] report optimal solutions for such instances where standard deterministic solvers fail. In any event, we have chosen to summarize these two methods because of their ability to *manage general integer decisions in both stages.* For prior research on more special cases, we refer to [53].

In this section, we first consider the easier case that the first-stage variables are binary (Section 3.2.1). Next, in Section 3.2.2, we consider general integer first-stage variables. Furthermore, we restrict our attention to convexifications based on the Gomory cutting planes, due to the finite convergence results on the Gomory cutting plane algorithm and the practical success of these cuts for deterministic pure integer programs. Finally, in Section 3.2.3, we review an algorithm for the general case of mixed-integer variables in the first and second stages. Throughout this section, we assume relatively complete recourse to ease exposition.

**3.2.1 Binary First Stage, General Integer Second Stage** In this subsection, we describe the decomposition algorithm proposed in [22], for a class of two-stage stochastic integer programs with binary variables in the first stage and general integer variables in the second stage. The approach is similar to the one taken in [54] in that we iteratively approximate the second-stage problems using cutting planes. Sen and Higle [54] use disjunctive cuts, whereas the algorithm we describe here uses Gomory cuts within the proposed decomposition algorithm. The Gomory fractional cutting plane algorithm [23] 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 [24] which have been successfully incorporated into branch-and-bound methods to solve deterministic mixed integer programs [3, 9]. Gomory cuts and their extensions have also been a subject of numerous theoretical studies in the mixed integer programming area [11, 18, 25]. Similarly, the Benders decomposition [5] and *L*-shaped methods [62] 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 [54, 57] 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 second stage. For such instances, the introduction of Gomory cuts would provide significantly stronger approximations as has been demonstrated for deterministic MILP instances.

In this subsection, we describe 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 $\pi_0$ is a function of $x$, we will refer to these valid inequalities as parametric Gomory cuts. We exploit the facial or extreme-point property of binary vectors to derive valid inequalities $\pi(\omega)^\top y(\omega) \geq \pi_0(x, \omega)$ where $\pi_0(\cdot, \omega)$ is 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. Consequently, they will result in mixed-integer *linear* first-stage master problems as was the case for continuous recourse.

To derive valid inequalities within a decomposition algorithm, suppose that a binary first-stage vector $\bar{x}$ is given. Without loss of generality, we can derive the cut coefficients in a translated space for which the point $\bar{x}$ 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 $(\bar{x}, \omega) \in (X \cap \mathcal{X}) \times \Omega$ 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 nonbasic 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 $\omega$. Multiplying (2b)-(2c) by $B^{-1}$ we obtain

$$y_{\mathcal{B}} + B^{-1}Ny_{\mathcal{N}} = B^{-1}\begin{pmatrix} 0 \\ r - T\bar{x} \end{pmatrix} =: \nu(\bar{x}). \tag{8}$$

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 $\Gamma$ be denoted by $\gamma_{ij}$. Let $\mathcal{B}_i$ denote the $i$th basic variable. We pick a candidate source row corresponding to $y_{\mathcal{B}_i}$ for which $\nu_i(\bar{x}) \notin \mathbb{Z}$ for generating the Gomory cut. The corresponding row in (8) may be written by rearranging the terms as

$$y_{\mathcal{B}_i} + \sum_{j \in \mathcal{N}} \bar{w}_{ij}y_j = \nu_i(\bar{x}), \tag{9}$$

where $\nu_i(x) = \rho_i - \sum_{j=1}^{n_1} \gamma_{ij}x_j \notin \mathbb{Z}$. Gomory fractional cut can be used to derive a valid inequality in the $y$ space that cuts off this fractional solution. Gomory [23] uses the observation that because all $y$ variables are integer, we can round up the coefficients of $y$ in the left-hand side of equality (9) to obtain an integer term on the left-hand side. In other words,

$$\underbrace{y_{\mathcal{B}_i} + \sum_{j \in \mathcal{N}} \lceil \bar{w}_{ij} \rceil y_j}_{\in \mathbb{Z}} \geq y_{\mathcal{B}_i} + \sum_{j \in \mathcal{N}} \bar{w}_{ij}y_j = \underbrace{\nu_i(\bar{x})}_{\notin \mathbb{Z}}. \tag{10}$$

Because the left-hand-side must be integer in the feasible solution, whereas $\nu_i(\bar{x})$ is not integral, we can derive a valid inequality by rounding up the right-hand side as well, to obtain the valid inequality

$$y_{\mathcal{B}_i} + \sum_{j \in \mathcal{N}} \lceil \bar{w}_{ij} \rceil y_j \geq \lceil \nu_i(\bar{x}) \rceil. \tag{11}$$

Letting $\xi(\beta) := \lceil \beta \rceil - \beta$, and subtracting the equation (9) from the inequality (11) we obtain a valid inequality

$$\sum_{j \in \mathcal{N}} \xi(\bar{w}_{ij})y_j \geq \xi(\nu_i(\bar{x})). \tag{12}$$

Note that this valid inequality cuts off the current integer infeasible solution, because $\sum_{j \in \mathcal{N}} \xi(\bar{w}_{ij})y_j = 0$ and $\xi(\nu_i(\bar{x})) > 0$. In fact, Gomory [23] shows that a carefully implemented pure cutting plane algorithm repeatedly solving LP relaxations and cutting off integer infeasible points with the use of Gomory cuts, finitely converges to the integer optimal solution. This process is referred to as the convexification of the integer program, as it relaxes the integer program with a series of progressively tighter linear programs until the linear programming relaxation yields an integer solution. Once the integer program is replaced with a linear program, Benders decomposition immediately applies. However, there are still some hurdles we need to overcome to be able to apply Benders decomposition. First, the convexification of the second-stage integer program is required for each feasible $x$. Alternatively, the cuts should be parametric in $x$, i.e., written in variable space $x$ as opposed to the current solution $\bar{x}$, but this involves nonlinear inequalities. In particular, inequalities (7b) will now involve terms such as $\lceil \nu_i(x) \rceil$. Clearly, the ceiling operator is no longer linear and the corresponding relaxed master programs are no longer mixed-integer linear programs. Secondly, for a given $\bar{x}$, we should not implement the Gomory cutting plane algorithm to its full conclusion, as this process is finite but slow. If we have a non-optimal $\bar{x}$, we would be spending a lot of effort in convexifying the second-stage problem for an undesirable solution. We next discuss how to overcome these issues.

Our first goal is to derive linear cuts in the $(x, y)$-space. Reintroducing the dependence on $x$, we rewrite equality (9) as

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

With a slight abuse of notation, the derivation here uses $x$ to denote points in the translated space. Because we are deriving cuts in the translated space at $\bar{x} = 0$, we can treat the $x$ variables as "nonbasic" in the current solution. Now a Gomory fractional cut in the $(x, y)$-space can be written as

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

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

$$\sum_{j \in \mathcal{N}} \xi(\bar{w}_{ij})y_j \geq \xi(\rho_i) - \sum_{j=1}^{n_1} \xi(\gamma_{ij})x_j. \tag{13}$$

Reintroducing the dependence on $(x, \omega)$, we note that inequality (13) 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 $\pi_0$ is affine in $x$. When $x = \bar{x}$, inequality (13) is the usual Gomory fractional cut valid for $Y(\bar{x}, \omega)$. Moreover, inequality (13) is valid for $Y(x, \omega)$ for all $x \in X \cap \mathcal{X}$. We illustrate these concepts in the next example.

EXAMPLE 3.1 Consider the two-stage single-scenario (deterministic) program $\min\{-x + h(x) : x \in \{0, 1\}\}$, where $h(x) = \min\{-y_1 : 2y_1 + 3y_2 = 4 + x, y_1, y_2 \in \mathbb{Z}_+\}$. Suppose that at the first iteration of solving the relaxed master problem, the first-stage solution is $x = 1$. In the second-stage problem, we have the constraint $2y_1 + 3y_2 = 5$, and the optimal simplex tableau for the LP relaxation gives the source row $y_1 + 1.5y_2 = 2.5$, where $y_1 = 2.5$ is basic and $y_2$ is non-basic. This solution is integer infeasible and the Gomory cut that cuts off this solution is $0.5y_2 \geq 0.5$. However, this cut is not globally valid, it is only valid if $x = 1$. (Consider the feasible solution $x = 0, y_1 = 2, y_2 = 0$.) Therefore, we need cuts that are parameterized in $x$. An alternative cut, valid for all $x$ would be $0.5y_2 \geq \lceil 2 + 0.5x \rceil - 2 - 0.5x$. However, as discussed earlier, such cuts would lead to nonlinear master problems. The parameterized inequalities we describe overcome these difficulties. First, they are valid for any $x$; second, they maintain the linearity of the master problems. Let $\bar{x} := 1 - x$ (translate so that the current solution is the origin). Now we can rewrite the source row as:
$$y_1 + \frac{3}{2}y_2 = 2 + \frac{(1 - \bar{x})}{2},$$
where $\bar{x}$ is seen as a non-basic variable. Then, we obtain the valid Gomory cut
$$\frac{1}{2}\bar{x} + \frac{1}{2}y_2 \geq \frac{1}{2},$$
which is equivalent to $y_2 \geq 1 - \bar{x} = x$. This is a valid inequality for all $x$, it cuts off the current fractional solution, and it is linear in $x$. We emphasize that the lifting of the Gomory cut heavily relies on the facial property of the binary programs. Because $x$ is either 0, in which case it is naturally taken to be non-basic, or 1, in which case we complement it to obtain $\bar{x}$ and treat $\bar{x}$ as non-basic, we were able to obtain valid parameterized Gomory cuts. When first-stage variables $x$ are general integer in the range $[a, b]$, we will no longer be able to perform this complementation trick for a first-stage solution where an integer variable has a value strictly between its bounds. The cuts we describe in Section 3.2.2 will address this challenge by performing a different lifting operation.

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

$$h_\ell^{k-1}(x, \omega) = \min y_0 \tag{14a}$$
$$y_0 - g(\omega)^\top y = 0 \tag{14b}$$
$$W_{k-1}(\omega)y \geq r_{k-1}(\omega) - T_{k-1}(\omega)x \tag{14c}$$
$$y \in \mathbb{R}_+^{n_2}, y_0 \in \mathbb{R}, \tag{14d}$$

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 (13) 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 (13) 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 $\psi_\omega^k$ denote the corresponding optimal dual multipliers. An affine under-estimator of $\mathbb{E}[h(x, \tilde{\omega})]$, or an *optimality cut*, can be generated as follows:

$$\eta - \sum_{\omega \in \Omega} p(\omega)(\psi_\omega^k)^\top (r_k(\omega) - T_k(\omega)x) \geq 0. \tag{15}$$

Thus, a lower bounding approximation for (6) at iteration $k$, denoted by $\mathrm{MP}^k$, can be written as follows:

$$\min \quad c^\top x + \eta \tag{16a}$$

$$Ax \geq b \tag{16b}$$

$$\eta - \sum_{\omega \in \Omega} p(\omega)(\psi_\omega^t)^\top (r_t(\omega) - T_t(\omega)x) \geq 0, \qquad t = 1, \ldots, k \tag{16c}$$

$$x \in \mathbb{B}^{n_1}, \eta \in \mathbb{R}. \tag{16d}$$

The algorithm executes as follows: We initialize the master problem with no optimality cuts, and subproblems using their linear relaxations. We start with some $x^1 \in X \cap \mathcal{X}$. In iteration $k$, using the first-stage solution $x^k$, we solve the scenario approximation problems $\mathrm{SP}^{k-1}(x^k, \omega)$ 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. On the other hand, for each scenario $\omega$ for which the solution $y(\omega)$ is non-integral, we generate a Gomory cut (13) 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 $\mathrm{SP}^k(x^k, \omega)$ using the lexicographic dual simplex method (required for finite convergence) and update the best upper bound if integer solutions are obtained for $\mathrm{SP}^k(x^k, \omega)$ for all $\omega \in \Omega$. If no cut is generated for a scenario $\omega$, 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 (15) using the optimal dual multipliers of the $\mathrm{SP}^k(x^k, \omega)$. We solve the updated master problem $\mathrm{MP}^k$ (16) using branch-and-bound, to obtain a first-stage solution $x^{k+1}$ and a lower bound for the SIP and increment the iteration index. We repeat this process until the lower and upper bounds are within an optimality tolerance.

A few remarks are in order. First, note that unlike standard Benders decomposition, the second-stage LP relaxation problem in this algorithm is updated with the Gomory cuts 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 [13]. Second, the reader might recall that for mixed binary second-stage recourse problems, Sen and Higle [54] 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. Third, this algorithm generates one Gomory cut for each first-stage solution, instead of generating a series of Gomory cuts that converge to an integer solution. This prevents us from slow convergence of pure cutting plane algorithms, when the first-stage solution at hand is not promising. Finally, note that this decomposition algorithm is also applicable when continuous and general integer variables are present in the first stage, when their coefficients in the technology matrix are zeros.

Gade et al. [22] prove that the resulting algorithm is finitely convergent under several assumptions. Next we summarize their computational experience with the Gomory-driven decomposition algorithm on instances of the stochastic server location problem (SSLP) [38] with at least 50 scenarios. These instances are available online as a part of the stochastic integer programming test problem library (SIPLIB) and their names are in the form SSLP_$|I|$_$|J|$_$|\Omega|$ to describe the problem size with respect to the number of client locations, server locations, and scenarios. The original instances have $B = \{1, 2\}$, $C = \{2\}$ and $D = \emptyset$. These instances are converted to pure integer second stage by changing the declaration of the continuous variables to integers. The results are summarized in Table 1. Here we report the

Table 1: Comparative computational results for two-stage stochastic pure IPs.

| Instances | DEF | | Gomory | |
|---|---|---|---|---|
| | Time | Gap | Time | Gap |
| SSLP_5_25_50 | 2.03 | 0.00 | 0.18 | 0.00 |
| SSLP_5_25_100 | 1.72 | 0.00 | 0.22 | 0.00 |
| SSLP_5_50_50 | 1.06 | 0.00 | 0.27 | 0.00 |
| SSLP_5_50_100 | 3.56 | 0.00 | 0.48 | 0.00 |
| SSLP_5_50_1000 | 212.64 | 0.00 | 2.88 | 0.00 |
| SSLP_5_50_2000 | 1020.54 | 0.00 | 5.73 | 0.00 |
| SSLP_10_50_50 | 801.49 | 0.01 | 109.2 | 0.02 |
| SSLP_10_50_100 | 3667.22 | 0.10 | 218.42 | 0.02 |
| SSLP_10_50_500 | 3601.32 | 0.38 | 740.38 | 0.03 |
| SSLP_10_50_1000 | 3610.06 | 3.56 | 1615.42 | 0.02 |
| SSLP_10_50_2000 | 3601.55 | 18.59 | 2729.61 | 0.02 |

solution time in seconds, and the percentage optimality gap at termination (with an hour time limit), in the columns `Time` and `Gap`, respectively. We compare the solution times of the deterministic equivalent formulation (DEF) and the decomposition algorithm using Gomory cuts. As can be seen from this table, DEF is unable to solve the larger instances within the time limit of 3600 seconds. In fact, for the largest instance with 2000 scenarios (in the last row), the optimality gap at termination is 18.59%. However, the Gomory-based decomposition algorithm is highly effective on this class of problems. For the same instance, 0.02% optimal solution is obtained after 2729 seconds.

**3.2.2 General Integer First and Second Stages** Note that the algorithm described in Section 3.2.1 relies on the facial property of the first-stage model. In this section, we describe an extension of this algorithm, proposed in [68], to the case when general integer variables are present in the first stage. In this case, the first-stage problem no longer has the facial property. As before, our first goal is to develop a Gomory cut $\pi(\omega)^\top y(\omega) \geq \pi_0(x,\omega)$ that is valid for the linear relaxation of the deterministic equivalent formulation (DEF), where $\pi_0(x,\omega)$ is an affine function of the first-stage decisions $x$, in the absence of the facial property.

For purposes of utilizing the simplex method and deriving Gomory cuts, we redefine matrices $A$, $T(\omega)$, and $W(\omega)$ to include the slack variables in both stages. For a given optimal solution to the linear relaxation of the first-stage problem, let $\bar{A}_{B_1} = [A_{B_1(1)}, \ldots, A_{B_1(a)}]$ 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 $\omega$, 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 $\omega_i \in \Omega$. The key observation in deriving affine inequalities is that

$$\begin{bmatrix} \bar{A}_{B_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \bar{T}_{B_1}(\omega_1) & W_{B_2}(\omega_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \bar{T}_{B_1}(\omega_2) & \mathbf{0} & W_{B_2}(\omega_2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{T}_{B_1}(\omega_{|\Omega|}) & \mathbf{0} & \mathbf{0} & \cdots & W_{B_2}(\omega_{|\Omega|}) \end{bmatrix}$$

is a feasible basis matrix for DEF.

For a 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

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

Then the Gomory cuts generated from any row of

$$G \begin{bmatrix} \bar{A} \\ \bar{T}(\omega_1) \\ \bar{T}(\omega_2) \\ \vdots \\ \bar{T}(\omega_{|\Omega|}) \end{bmatrix} x + G \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ W(\omega_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & W(\omega_2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & W(\omega_{|\Omega|}) \end{bmatrix} y = G \begin{bmatrix} \bar{b} \\ r(\omega_1) \\ r(\omega_2) \\ \vdots \\ r(\omega_{|\Omega|}) \end{bmatrix} \quad (17)$$

are valid for DEF. These are general-integer versions of the *parametric Gomory cuts* of Section 3.2.1, because they are parameterized with respect to the general integer first-stage decision variables $x$.

The decomposition algorithm follows similarly to the one described in Section 3.2.1. However, there are a few key differences, which we describe next. At iteration $k$ for a given $x^k$ from the master, the subproblems in the form (14) are solved and a parametrized Gomory cut derived from the system (17) is added at fractional solutions. However, due to the need for a basis for the first-stage problem, unlike the algorithm for binary first-stage variables, in the general integer case, the relaxed master problem is not solved to integer optimality with branch-and-bound. Instead at iteration $k$, a relaxed master *linear* program of the form

$$\text{MP}^k: \quad \min \quad c^\top x + \eta$$
$$A^k x \geq b^k,$$
$$(16c).$$

If the solution to this LP is not integral, then a Gomory cut is added to $\text{MP}^k$. Hence, the constraint set $A^k x \geq b^k$ includes the original constraints $Ax \geq b$, and for $k \geq 1$ the Gomory cuts generated for the first-stage problem in iterations $j = 1, \ldots, k$. We refer the reader to [68] for the finite convergence and computational results under the general integer setting.

Finite convergence of a pure cutting plane algorithm based on Gomory cuts for deterministic pure integer programs is crucial in ensuring the finite convergence of the corresponding decomposition algorithm for stochastic pure IPs. Similarly, Sen and Higle [54] utilize the finite convergence of pure cutting plane algorithms based on (two-term) disjunctive cuts for deterministic mixed-binary programs to develop a finitely convergent decomposition algorithm for stochastic mixed-binary programs. However, even though Gomory cutting planes can be generalized to the mixed-integer case, the corresponding Gomory mixed-integer (GMI) cuts do not lead to finitely convergent pure cutting plane algorithms for MIPs. Similarly, pure cutting plane algorithms using disjunctive cutting planes do not converge finitely for deterministic MIPs with both continuous and general integer variables. (See [21] for a review of pure cutting plane algorithms and their convergence for deterministic MIPs.) Hence, it does not seem possible to immediately extend the decomposition algorithms of [22, 54, 68] to problems that contain both continuous and general integer variables in the first and second stages. In the next subsection, we review a decomposition algorithm that overcomes these difficulties.

**3.2.3 Mixed-Integer First and Second Stages** As mentioned earlier, the algorithm we describe here, referred to as the Ancestral Benders Cut (ABC) algorithm, covers all the bases: randomness in all data elements, and the decision variables can be characterized by the property that $B = C = D = \{1, 2\}$. While some might consider this to be a major strength of the algorithm, its main contribution stems from the recognition that this algorithm, in fact, could not have been developed without a new result
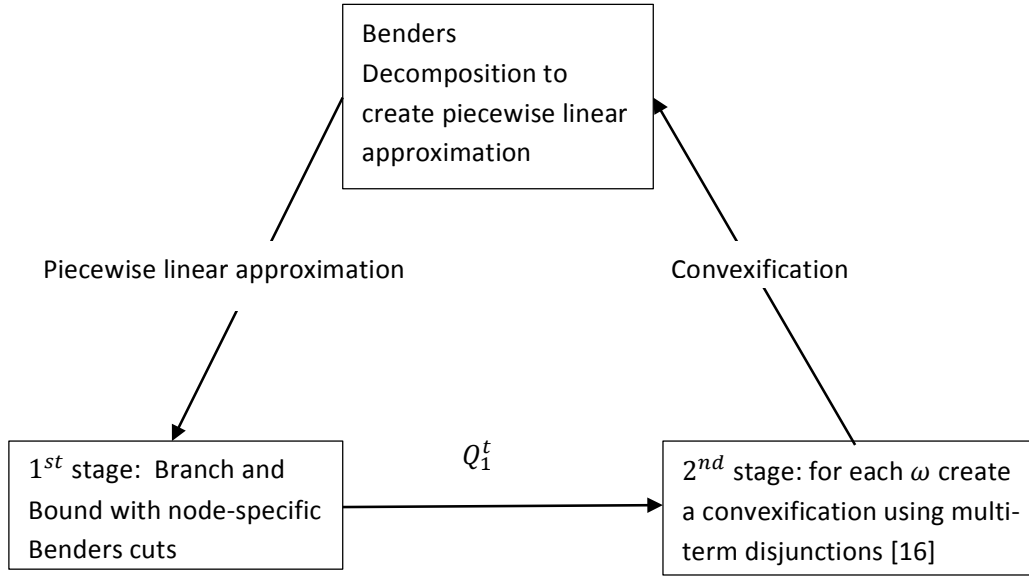
Figure 2: Overview of the ABC Algorithm

in polyhedral theory. In other words, the quest for this algorithm actually led to a new MIP result on the representability and computability of the convex hull of mixed-integer polyhedral sets by discovering finitely many linear inequalities. This finding happens to be critical for a general SMIP algorithm because one can now support the claim that the value function of a mixed-integer programming problem can be represented by generating a sequence of cutting planes, some of which are intended to refine an approximation of the two-stage MIP polyhedron, whereas other cutting planes are used to refine the value function approximation of each MIP in the scenario set. The representability result appears in [16] and its computability appears in [17]. Incidentally, these cuts are based on multi-term disjunctions which we refer to as Cutting Plane Tree (CPT) cuts.

Based on the development mentioned above, Qi and Sen [44] use CPT cuts to develop two versions of the ABC algorithm: one based purely on a polyhedral approximation of the feasible set for each scenario, and another based on a disjunctive representation of each scenario. The first option (creating a polyhedral approximation), referred to as ABC (CPT-D), uses the algorithm in [17] to create cuts in the $(x, y)$ space. These iterations are carried out in two phases: for a given $x^k$, we perform a sequence of iterations indexed by $d$ which create improved approximations by deleting points $x^k, \{y^d\}$. And after a few second-stage iterations, we create a Benders cut (i.e., value function approximation) to communicate to the first stage. The second option (using a disjunctive representation), referred to as ABC (BB-D), uses a truncated branch-and-bound (TBB) process to search for second stage decisions (with fixed $x^k$). Here TBB refers to running BB for a few steps in the second stage. We then stop TBB to create several CPT cuts which provide an approximation with the same second-stage solution (possibly fractional) as the TBB process. Because of the finiteness of CPT cuts, we thus discover a finite set of inequalities and an LP relaxation with the same solution as the TBB process. This second stage LP relaxation is now used to create a Benders cut to pass to the first stage.

Figure 2 depicts the three major pieces of the ABC algorithm. These ingredients provide a fully integrated algorithm that is not only computationally realistic, but also provably convergent. The brief summary below connects the elements of Figure 2 to the description provided above.

    a) The branch-and-bound (B&B) search in the first stage divides the range of first-stage integer variables into a partition consisting of disjoint subsets $\{Q_1^t\}$ covering the entire set $Q_1$. The

partition is refined using a breadth-first search strategy.

b) For any subset $Q_1^t$ of the first-stage decisions $(x)$, we create polyhedral approximations of

$$\mathcal{Z}(t,\omega) := \{(x, y(\omega)) \mid T(\omega)x + W(\omega)y(\omega) \geq r(\omega), x \in X \cap Q_1^t, y(\omega) \in \mathcal{Y} \cap Q_2\}.$$

c) Using Benders decomposition for each box $Q_1^t$, we then create lower bounding approximations. Using these bounds, one can proceed to partition the most promising node, and continue with the branch-and-bound search.

The details underlying these steps are beyond the scope of the current tutorial, but it is appropriate to summarize one of the computational experiments reported in [44] for the SSLS instances described in Example 2.1. Table 2 shows the results. The instances are named SSLS_$(m \times u)$_$(n \times v)$_$|\Omega|$, to describe the parameters $m, n, u, v, |\Omega|$ as described in Example 2.1. The number of variables (**Var**) and constraints (**Constr**) refer to numbers in the Deterministic Equivalent Formulation (DEF). The metrics reported in the column labeled **ABC(BB-D)** refer to the number of iterations and in parentheses, the numerical quantities refer to the number of master nodes, the number of second-stage leaf nodes, and the running time. Of the 27 instances that were tried, ABC was able to report optimal solutions for 25, and the total time required was 3.18 hours. In comparison, CPLEX 12.6 requires 12.81 hours and solved 15 instances successfully. While the ones that were not solved to optimality showed small gaps, one should still note that the total time used by CPLEX was approximately four times greater than ABC. Based on these comparisons, the ABC approach shows far greater potential for SMIP models.

**4. Two-Stage Chance-Constrained Mixed-Integer Programming** In two-stage stochastic MIP models described in Section 3, it is required that the second-stage problems are feasible for every scenario. This requirement can lead to conservative solutions. For example, in a power generation setting, the second-stage problem may impose that all load (demand) must be met on time over a planning horizon, given the first-stage decisions on on/off status of the generators. However, given high levels of uncertainty, especially with renewable power generation, it may not be possible to ensure that all demand is met in all possible circumstances. In contrast, it is often desirable to impose constraints such that the loss-of-load probability is no more than a pre-specified risk level $\epsilon \in [0, 1]$. In essence, the first-stage solution is required to lead to feasible solutions in the second-stage most of the time, but in rare circumstances, such as extreme weather events, it is possible that the second-stage problem is infeasible for the given first-stage solution. Such restrictions lead to models that involve chance constraints ($A = RCC$).

For ease of exposition, we will first assume the setting $B = \{1\}, D = \{1\}, C = \{1, 2\}$. In other words, the second-stage problems are assumed to be linear programs. From our earlier development in Section 3, it will be easily seen that this is not a restricting assumption so long as the algorithms developed for chance-constrained programs work within a Benders decomposition framework. If integer variables are present in the second-stage, then convexification schemes proposed in Section 3 are directly applicable to the Benders algorithm developed in this section.

The chance-constrained mixed-integer programming with recourse (SMIP-RCC) model is given by

$$\min \quad c^\top x + \mathbb{P}(x \in \mathcal{P}(\tilde{\omega})) \mathbb{E}[h(x, \tilde{\omega}) | x \in \mathcal{P}(\tilde{\omega})], \tag{18a}$$

$$\mathbb{P}(x \in \mathcal{P}(\tilde{\omega})) \geq 1 - \epsilon \tag{18b}$$

$$x \in X \cap \mathcal{X}, \tag{18c}$$

where $\mathcal{P}(\omega) = \{x : \exists y \text{ satisfying } W(\omega)y \geq r(\omega) - T(\omega)x, y \in \mathcal{Y}\}$ and

$$h(x, \omega) = \min \quad g(\omega)^\top y \tag{19a}$$

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

$$y \in \mathcal{Y}. \tag{19c}$$

This model is first introduced in [34] (see also [35] for a related model that does not involve the second-stage cost function $h(x, \omega)$ and considers only the feasibility of the second-stage problem). It is important

Table 2: Computational results with **ABC/BB-D** on SSLS instances.

| **Instance**: SSLS | **Var** | **Constr** | **ABC(BB-D)** $K(\mathcal{T}_1, \mathcal{T}_2, s)$ | **DEF** Time (Gap) | |
|---|---|---|---|---|---|
| $(2 \times 5)\_(5 \times 5)\_50$ | 602 | 601 | 11 (13, 16, 0.30) | 0.09 | |
| $(2 \times 5)\_(5 \times 5)\_100$ | 1202 | 1201 | 9 (13, 2, 0.38) | 0.2 | |
| $(2 \times 5)\_(5 \times 5)\_500$ | 6002 | 6001 | 12 (13, 22, 3.58) | 0.73 | |
| $(2 \times 5)\_(10 \times 5)\_50$ | 1102 | 1101 | 14 (15, 50, 0.52) | 0.96 | |
| $(2 \times 5)\_(10 \times 5)\_100$ | 2202 | 2201 | 13 (13, 515, 4.84) | T.O (0.15%) | |
| $(2 \times 5)\_(10 \times 5)\_500$ | 11002 | 11001 | 13 (17, 258, 4.43) | 0.76 | |
| $(2 \times 5)\_(15 \times 5)\_50$ | 1602 | 1601 | 20 (23, 344, 4.26) | T.O (0.24%) | |
| $(2 \times 5)\_(15 \times 5)\_100$ | 3202 | 3201 | 16 (21, 100, 1.64) | 0.78 | |
| $(2 \times 5)\_(15 \times 5)\_500$ | 16002 | 16001 | 15 (17, 411, 51.28) | T.O (0.06%) | |
| $(3 \times 5)\_(5 \times 5)\_50$ | 903 | 651 | 23 (33, 2, 0.59) | 0.27 | |
| $(3 \times 5)\_(5 \times 5)\_100$ | 1803 | 1301 | 23 (33, 2, 1.23) | 0.14 | |
| $(3 \times 5)\_(5 \times 5)\_500$ | 9003 | 6501 | 23 (35, 5, 7.09) | 163.77 | |
| $(3 \times 5)\_(10 \times 5)\_50$ | 1653 | 1151 | 36 (59, 343, 4.68) | T.O (0.19%) | $K$ is the total |
| $(3 \times 5)\_(10 \times 5)\_100$ | 3303 | 2301 | 32 (45, 8632, 158.61) | 2715.04 | |
| $(3 \times 5)\_(10 \times 5)\_500$ | 16503 | 11501 | 28 (41, 566, 27.15) | T.O. (0.02%) | |
| $(3 \times 5)\_(15 \times 5)\_50$ | 2403 | 1651 | 37 (55, 9435, 191.92) | 9.73 | |
| $(3 \times 5)\_(15 \times 5)\_100$ | 4803 | 3301 | 29 (33, 1290, 19.56) | T.O. (0.14%) | |
| $(3 \times 5)\_(15 \times 5)\_500$ | 24003 | 16501 | 40 (57, 8984, 1069.92) | T.O. (0.16%) | |
| $(4 \times 5)\_(5 \times 5)\_50$ | 1204 | 701 | 58 (87, 8, 1.59) | 1.59 | |
| $(4 \times 5)\_(5 \times 5)\_100$ | 2404 | 1401 | 59 (91, 9, 3.36) | 5.07 | |
| $(4 \times 5)\_(5 \times 5)\_500$ | 12004 | 7001 | 57 (87, 2, 20.99) | 4.46 | |
| $(4 \times 5)\_(10 \times 5)\_50$ | 2204 | 1201 | 63 (91, 130, 3.60) | 0.98 | |
| $(4 \times 5)\_(10 \times 5)\_100$ | 4404 | 2401 | 68 (111, 6419, 261.89) | T.O. (0.15%) | |
| $(4 \times 5)\_(10 \times 5)\_500$ | 22004 | 12001 | 64 (97, 8683, 745.61) | T.O. (0.05%) | |
| $(4 \times 5)\_(15 \times 5)\_50$ | 3204 | 1701 | 104 (161, 5849, 1653.67) | T.O. (0.09%) | |
| $(4 \times 5)\_(15 \times 5)\_100$ | 6404 | 3401 | T.O. (>3600) | T.O. (0.01%) | |
| $(4 \times 5)\_(15 \times 5)\_500$ | 32004 | 17001 | T.O. (>3600) | T.O. (1.29%) | |

number of iterations; $\mathcal{T}_1$ is the total number of nodes in the B&B tree in the master problem; $\mathcal{T}_2$ is the maximum number of leaf nodes encountered during solving the subproblem; $s$ is the total running time (secs.) and time limit is 3600 secs; *Gap* is the MIP gap returned by CPLEX 12.6 when it timed out (T.O)

to note that in this model, the undesirable outcomes $\omega$ such that $x \notin \mathcal{P}(\omega)$ are simply ignored. Liu et al. [34] propose an extension of the two-stage model (18), where they allow so-called recovery decisions for the undesirable scenarios. In this tutorial, we restrict our attention to the simpler problem and refer the reader to [34] for extensions of the proposed methods when recovery decisions are available. Note that the SMIP-RCC model subsumes the SMIP-RC model as a special case when $\epsilon = 0$.

Most of the work in chance-constrained mathematical programs (CCMP) prior to this two-stage model (see, e.g., [14, 15, 37, 42]) can be seen as single-stage (i.e., static) decision-making problems where the decisions are made here and now, and there are no recourse actions once the uncertainty is revealed (i.e., $\mathcal{P}(\omega) = \{x : T(\omega)x \geq r(\omega)\}$ and $h(x, \omega) = 0$). In addition, Zhang et al. [69] consider multi-stage CCMPs and give valid inequalities for the deterministic equivalent formulation, and observe that decomposition algorithms are needed to solve large-scale instances of these problems. In this tutorial, we review such a decomposition algorithm for the two-stage problem.

The two-stage chance-constrained problem can be formulated as a large-scale deterministic mixed-integer program by introducing a big-$M$ term for each inequality in the chance constraint and a binary variable for each scenario. In particular, analogous to the two-stage SMIP model, the deterministic equivalent formulation (DEF) of the two-stage chance-constrained MIP (SMIP-RCC) may be stated as

$$\min \quad c^\top x + \sum_{\omega \in \Omega} p(\omega) g(\omega)^\top y(\omega) z(\omega) \tag{20a}$$

$$Ax \geq b, \tag{20b}$$

$$T(\omega)x + W(\omega)y(\omega) \geq r(\omega) - Mz(\omega), \qquad \omega \in \Omega \tag{20c}$$

$$\sum_{\omega \in \Omega} p(\omega) z(\omega) \leq \epsilon, \tag{20d}$$

$$x \in \mathcal{X}, y(\omega) \in \mathcal{Y}, \qquad \omega \in \Omega \tag{20e}$$

$$z(\omega) \in \{0, 1\} \qquad \omega \in \Omega, \tag{20f}$$

where $z(\omega)$ is a binary variable that equals 0 only if the second-stage problem for scenario $\omega$ has a feasible solution, and $M$ is a large enough constant that makes constraint (20c) trivially satisfied if $z(\omega) = 1$, i.e., if the second-stage problem for scenario $\omega$ need not be feasible. The knapsack constraint (20d) is the linear reformulation of the joint chance constraint (18b). There are multiple difficulties with this formulation that are not present in the two-stage SMIP counterpart without the chance constraint. First, note that the objective function (20a) is nonlinear. In addition, the big-$M$ formulation in constraints (20c) leads to weak linear programming relaxations. A difficulty that is shared with the two-stage SMIP counterpart is the large size of the problem due to the copies of the variables $y(\omega)$ for $\omega \in \Omega$. In addition, (20) has a large number of binary variables $z(\omega), \omega \in \Omega$ introduced to capture the chance constraint, which are not present in the two-stage SMIP counterpart.

Luedtke et al. [36] and Küçükyavuz [29] study strong valid inequalities for the deterministic equivalent formulation of chance-constrained problems with random right-hand sides. Alternative reformulations for this class of problems employ the concept of $(1 - \epsilon)$-efficient points, which are an exponential number of points representing the multivariate value-at-risk associated with the chance constraint (18b) [19, 32, 43, 51]. Such alternative formulations lead to specialized branch-and-bound algorithms described in [6, 7, 47, 49]. For problems with special structures, formulations that do not involve additional binary variables can be developed (see, e.g., [59]). In this section, we consider a general class of problems and describe an algorithm that exploits the decomposable nature of the formulation (20). This algorithm also replaces the weak big-$M$ constraints (20c) with stronger optimality and feasibility cuts.

**4.1 Benders Decomposition-Based Branch-and-Cut Algorithm** Note that the traditional Benders method cannot be directly applied to SMIP-RCC, since both the feasibility and optimality cuts of the Benders method work on the assumption that all second stage problems must be feasible, which is not the case for CCMPs. For general recourse problems, feasibility and optimality cuts different from the traditional Benders cuts must be developed.

Let $\eta_\omega$ represent a lower bounding approximation of the optimal objective function value of the second-stage problem under scenario $\omega \in \Omega$. Without loss of generality, we assume that $\eta_\omega \geq 0$ for all $\omega \in \Omega$. At each iteration of a Benders decomposition method, a sequence of relaxed master problems (RMP) of the following form is obtained.

$$\min \quad c^\top x + \sum_{\omega \in \Omega} p(\omega)\eta_\omega \tag{21a}$$

$$Ax \geq b, \tag{21b}$$

$$\sum_{\omega \in \Omega} p(\omega)z(\omega) \leq \epsilon, \tag{21c}$$

$$(x, z) \in \mathcal{F}, \tag{21d}$$

$$(x, z, \eta) \in \mathcal{O}, \tag{21e}$$

$$x \in \mathcal{X} \tag{21f}$$

$$z(\omega) \in \{0, 1\} \qquad\qquad \omega \in \Omega. \tag{21g}$$

Here, $\mathcal{F}$ and $\mathcal{O}$ denote the set of feasibility and optimality cuts generated so far, respectively. Next we describe these cuts in detail.

At iteration $k$, let $(x^k, z^k)$ be the optimal solution to the RMP. Given this first-stage solution, during the course of solving the second-stage MILP subproblem for outcome $\omega$, suppose that we solve the following LP relaxation

$$h(x, \omega) = \min g(\omega)^\top y \tag{22a}$$

$$W_k(\omega)y \geq r_k(\omega) - T_k(\omega)x^k \tag{22b}$$

$$y \geq 0, \tag{22c}$$

where $W_k(\omega), T_k(\omega)$, and $r_k(\omega)$ include the original recourse and technology matrices, and the right-hand side, as well as any additional convexification cuts added to this subproblem until iteration $k$.

- Feasibility cuts: Note that $z^k(\omega) = 0$ for some $\omega \in \Omega$ implies that the restrictions in the chance constraint must be satisfied, i.e., the second-stage problem must be feasible. Hence, we check if the second-stage linear program (22) is indeed feasible. If it is infeasible for some $\bar{\omega} \in \Omega$, then $z^k(\bar{\omega}) = 0$ is an inconsistent first-stage solution with respect to $x^k$. Hence a feasibility cut must be added to deem the current solution infeasible.

  First, we obtain an extreme ray $\psi_{\bar{\omega}}$ associated with the dual of (22) for scenario $\bar{\omega}$ that yields the inconsistent solution. Next, we solve the following single-scenario optimization problem for all scenarios $\omega \in \Omega$ and $\phi = \psi_{\bar{\omega}}^\top T_k(\bar{\omega})$:

$$q_\omega(\phi) = \qquad\qquad \min \quad \phi x$$
$$T_k(\omega)x + W_k(\omega)y \geq r_k(\omega),$$
$$x \in X \cap \mathcal{X}, y \in \mathcal{Y}.$$

We sort the values $q_\omega(\phi)$ for $\omega \in \Omega$, to obtain a permutation $\sigma$ such that:

$$q_{\sigma_1}(\phi) \geq q_{\sigma_2}(\phi) \geq \cdots \geq q_{\sigma_{|\Omega|}}(\phi).$$

Let $a$ be the largest integer such that $\sum_{j=1}^a p(\sigma_j) \leq \epsilon$. Then, consider a subset $S = \{s_1, s_2, \ldots, s_l\} \subseteq \{\sigma_1, \sigma_2, \ldots, \sigma_a\}$ such that $q_{s_i}(\phi) \geq q_{s_{i+1}}(\phi)$ for $i = 1, \ldots, l$, where $s_1 = \sigma_1$ and $s_{l+1} = \sigma_{a+1}$. Luedtke [35] shows that the inequality

$$\phi x + \sum_{i=1}^l \left( q_{s_i}(\phi) - q_{s_{i+1}}(\phi) \right) \beta_{s_i} \geq q_{s_1}(\phi) \tag{23}$$

is a valid feasibility cut that cuts off the current infeasible first-stage solution $(x^k, z^k)$ with $z^k(\bar{\omega}) = 0$. The separation problem of inequality (23) can be solved with sorting [26]. One may

also use stronger inequalities than (23), as proposed in [1, 29, 36, 70], however this is at the expense of harder separation problems.

If for all $\omega \in \Omega$, the second-stage problem associated with scenario $\omega$ such that $z^k(\omega) = 0$ is indeed feasible, then the current solution $(x^k, z^k)$ is a feasible solution and no feasibility cuts are necessary. However, optimality cuts may be needed. Next we describe how to obtain valid optimality cuts.

- Optimality cuts: Given a first-stage solution $(x^k, z^k)$, let $\psi_\omega$ be the dual vector associated with the optimal basis of the second-stage problem (22) for scenario $\omega$ at this iteration.

A few studies [63, 64] have attempted to integrate Benders decomposition to solve two-stage CCMP, but the optimality cuts in these algorithms involve undesirable "big-$M$" coefficients, which lead to weak lower bounds and computational difficulties. In particular, the following inequality is a valid optimality cut

$$\eta_\omega + M_\omega z(\omega) \geq \psi_\omega^\top (r(\omega) - T(\omega)x), \tag{24}$$

where $M_\omega, \omega \in \Omega$ is assumed to be large enough so that inequality (24) is redundant whenever $z(\omega) = 1$. Typically, such big-$M$ coefficients are obtained by observation of the problem structure, without resorting to secondary optimization problems. We refer to inequality (24) as the big-$M$ optimality cut. Next we describe a strong optimality cut that leads to faster convergence to an optimal solution.

If $z^k(\omega) = 0$ at the current solution of the master problem and the associated original second-stage problem (22) is feasible, then we solve the second-stage problem (22) and obtain the optimal dual vector $\psi_\omega$. It is easy to see that $\eta_\omega \geq \psi_\omega^\top (r(\omega) - T(\omega)x)$ is a valid optimality cut for $x \in X \cap \mathcal{X}$ (in fact for $x \in \mathcal{P}(\omega)$) if $z(\omega) = 0$. (Note that this is the traditional Benders optimality cut.) However, it may not be valid for all $x \in X \cap \mathcal{X}$ if $z(\omega) = 1$. To obtain a valid optimality cut we solve the following secondary problem with $\phi = \psi_\omega^\top T(\omega)$:

$$\bar{v}_\omega(\phi) = \min \quad \phi x$$
$$x \in X \cap \mathcal{X}, \quad y \geq 0.$$

Then we add the optimality cut of the form

$$\eta_\omega + \left( \psi_\omega^\top r(\omega) - \bar{v}_\omega(\phi) \right) z(\omega) \geq \psi_\omega^\top (r(\omega) - T(\omega)x). \tag{25}$$

To see the validity of this inequality at $z(\omega) = 1$, note that in this case, the second-stage objective function contribution for scenario $\omega$ is zero (this scenario is ignored in the objective). Furthermore, inequality (25) evaluated at $z(\omega) = 1$ reduces to $\eta_\omega \geq \bar{v}_\omega(\phi) - \phi x$. Because $\bar{v}_\omega(\phi) - \phi x \leq 0$ for all $x \in X \cap \mathcal{X}$ and $\eta_\omega \geq 0$, this inequality is trivially satisfied.

The finite convergence of the resulting algorithm is proven in [34] under certain assumptions.

To illustrate the benefits of decomposition, we summarize a set of computational experiments that appear in [34] in Table 3. We test our approach on a resource planning problem adapted from [35]. It consists of a set of resources (e.g., server types), denoted by $i \in I$, which can be used to meet demands of a set of customer types, denoted by $j \in J$. In the first stage, the number of servers to staff is determined, the second-stage problem then allocates these servers to meet the clients' demands with high probability $(1 - \epsilon)$. In Table 3, we report our experiments when the server utilization rates are deterministic, and the only uncertainty is in demand. However, Liu et al. [34] report detailed experiments when the utilization rates are also uncertain, and when demand is not met, recovery options, such as outsourcing, are available.

We compare the proposed "Strong" decomposition algorithm which uses the optimality cuts (25), against two other approaches: the deterministic equivalent formulation (DEF) (20) and the "Basic" decomposition approach which uses the strong feasibility cuts (23) and the big-$M$ optimality cuts (24) with an appropriate choice of big-$M$ as described in [34]. In this table, we report the optimality gap at termination for instances which reach the time limit of 3600 seconds. (In some cases, the optimality gap cannot be calculated because even the LP relaxation cannot be solved within the time limit, which we

Table 3: Comparative computational results for two-stage CCMPs.

| Instances | | DEF | Basic | Strong | |
|---|---|---|---|---|---|
| $(|\Omega|, \epsilon)$ | $(|I|, |J|)$ | Gap | Gap | Time | Gap |
| | (5,10) | 4.60 | 2.34 | 166 | 0 |
| (2000, 0.05) | (10,20) | - | 2.93 | 483 | 0 |
| | (15,30) | - | 2.69 | 1106 | 0 |
| | (5,10) | 4.64 | 2.61 | 279 | 0 |
| (2500, 0.05) | (10,20) | - | 3.08 | 711 | 0 |
| | (15,30) | - | 2.88 | 1819 | 0.09 |
| | (5,10) | 7.1 | 5.46 | 723 | 0 |
| (2000, 0.1) | (10,20) | - | 5.99 | 1069 | 0 |
| | (15,30) | - | 6.27 | 1032 | 0 |
| | (5,10) | 7.63 | 5.32 | 641 | 0 |
| (2500, 0.1) | (10,20) | - | 5.79 | 1198 | 0 |
| | (15,30) | - | 6.03 | 2112 | 0.02 |

indicate with a "-".) For DEF and Basic, all instances tested hit the time limit, so we only report the solution time for Strong decomposition. From Table 3 we see that the deterministic equivalent formulation is not able to solve any instances to optimality within the time limit. Moreover, it fails to find a feasible solution to the LP relaxation for larger instances. The basic decomposition algorithm which uses the big-$M$ optimality cuts, makes a big improvement over the deterministic equivalent formulation. However, because of the weak lower bound resulting from the big-$M$ optimality cuts, it is still not capable of solving any of the 36 instances within the time limit. (For each instance type we have three randomly generated instances.) The optimality gaps are between 2-6% for the basic decomposition algorithm after an hour. In contrast, the Strong decomposition algorithm, based on the proposed strong optimality cuts, is able to solve most of the instances to optimality. For the two unsolved instances, the average optimality gap is less than 0.1%.

**5. Approximations in Stochastic Mixed-Integer Programming with Recourse**  This section covers two specialized model forms, namely, simple integer recourse, and totally unimodular recourse, as well as bounds for more general two-stage SMIP with random variables in any data element of the second stage, and continuous and/or discrete variables in any stage.

**5.1 Simple Integer Recourse (SIR) Models with Random RHS**  A study of this class of models dates back to the dissertation of Maarten van der Vlerk [60]. This class of models is the pure integer analog of the continuous simple recourse model. For these models, one assumes that only the right-hand-side vector is random, and hence it is more convenient to just replace $r(\omega)$ by $\omega$ in this description. Let $\omega_i$ and $t_i$ denote the $i^{th}$ element of $\omega$ and the $i^{th}$ row of $T$ respectively, and let $\chi_i = t_i^\top x$. Moreover, define a scalar function $\lceil v \rceil^+ = \max\{0, \lceil v \rceil\}$ and $\lfloor v \rfloor^- = \max\{0, -\lfloor v \rfloor\}$. Finally, assume that the second stage costs satisfy $g_i^+, g_i^- > 0$, $i = 1, \ldots, m_2$. Then the statement of the SIR model is as follows.

$$\min_{x \in X \subseteq \mathbb{R}_+^{n_1}} \{c^\top x + \sum_i \mathbb{E}[g_i^+ \lceil r_i(\tilde{\omega}) - \chi_i \rceil^+ + g_i^- \lfloor r_i(\tilde{\omega}) - \chi_i \rfloor^-] \mid \chi - Tx = 0\}. \qquad (26)$$

While the above model has the form of a single stage problem, it is not difficult to envision the underlying two-stage structure. One may therefore classify this model according to the sets $B = \emptyset, C = \{1\}, D = \{2\}$. Klein Haneveld et al. [27, 28] have shown that whenever $\tilde{\omega}$ has finite support, the convex hull of the expected recourse function can be computed by enumerating over each dimension $i$. Let the $i^{th}$ component of the expected recourse function in (26) be defined as follows.

$$\hat{R}_i(\chi_i) = \mathbb{E}\left[g_i^+ \lceil \omega_i - \chi_i \rceil^+ + g_i^- \lfloor \omega_i - \chi_i \rfloor^-\right],$$

where $\chi_i$ denotes the $i^{th}$ component of the vector $\chi$. The above function is clearly discontinuous because

the floor/ceiling operations involve $\chi_i$. Defining $R_i(\chi_i)$ by eliminating the discontinuous (floor/ceiling) operations yields the *continuous counterpart* of $\hat{R}_i(\chi_i)$. Because $g_i^+, g_i^-$ are both assumed to be positive, we observe that

$$R_i(\chi_i) \leq \hat{R}_i(\chi_i) \leq R_i(\chi_i) + \max\{g_i^+, g_i^-\}. \tag{27}$$

The following result summarizes the nature of the approximation.

THEOREM 5.1 *Let $\hat{R}_i^c$ denote any convex function that satisfies (27), and let $(\hat{R}_i^c)'_+$ denote its right directional derivative. Then, for $a \in \mathbb{R}$ the following function is a cumulative distribution function (cdf).*

$$P_i(a) = \frac{(\hat{R}_i^c)'_+(a) + g_i^+}{g_i^+ + g_i^-}.$$

*Moreover, if $\vartheta_i$ is a random variable with cdf $P_i$, then for all $\chi_i \in \mathbb{R}$,*

$$\hat{R}_i^c(\chi_i) = g_i^+ \mathbb{E}\left[(\vartheta_i - \chi_i)^+\right] + g_i^- \mathbb{E}\left[(\chi_i - \vartheta_i)^+\right] + \frac{g_i^+ c_i^+ + g_i^- c_i^-}{g_i^+ + g_i^-}, \tag{28}$$

*where $(v)^+ = \max\{0, v\}$. The quantities $c_i^+, c_i^-$ are asymptotic discrepancies between $\hat{R}_i$ and $R_i$, and are defined as follows.*

$$c_i^+ = \lim_{\chi_i \to \infty} \hat{R}_i(\chi_i) - R(\chi_i), \quad \text{and} \quad c_i^- = \lim_{\chi_i \to -\infty} \hat{R}_i(\chi_i) - R(\chi_i).$$

The mathematical property which relates the SIR problem to its continuous counterpart is the basis for the approximations to solve (26). Note that unlike (26), the expectations in (28) do not include any ceiling/floor functions. Hence if one identifies random variables $\vartheta_i$ with cdf $P_i$, then, we may use the continuous counterpart to obtain an approximation of the SIR model. Since our focus is on approximations of the recourse function, we will not go into the details of creating such a cdf. Nevertheless, we point the reader to the original papers where it is suggested to use a Graham-scan method to find a two-dimensional convex-hull of $(\chi_i, R_i(\chi_i))$ [27, 28].

**5.2 Totally Unimodular Recourse (TUR) Models with Random RHS** As with the SIR case, TUR models may be described as having $B = \emptyset, C = \{1\}, D = \{2\}$. Many applications in transportation/logistics use recourse decisions based on network flows in the second stage (recourse). These applications often lead to recourse matrices $W$ that satisfy total unimodularity and in this sense, allows recourse decisions to be more general than in SIR. Nevertheless, the development to date allows randomness to only appear in the right hand side vector (i.e., $g, T, W$ in (2) are assumed to be fixed). Consequently, it is notationally convenient to write the second stage functional constraints in the form $Wy \geq \omega - Tx$. These models also require a few assumptions on the second stage problem: a) the complete recourse assumption (that the second stage is feasible for all $x \in \mathbb{R}^{n_1}$), b) the dual recourse problem is feasible for all outcomes, and c) all elements of the random vector have finite expectations. In fact, the complete recourse assumption can be relaxed to the more common "relatively complete recourse" assumption (that the second stage is feasible for all $x \in X$). Finally, there is one more assumption on the optimal dual multipliers of the second stage problem, namely a *monotonicity property* mentioned below.

The basic strategy is one of relaxing the second-stage integrality (on $y$) while perturbing the distribution of the random variable $\omega$. We focus on the so-called $\alpha$-approximation [61], although stronger bounds are available in [46]. An $\alpha$ approximation of the recourse function $h$ replaces the second stage recourse function by the following, which is defined for any $\alpha \in \mathbb{R}^{m_2}$ (with $m_2$ denoting the number of rows in the second stage problem).

$$\begin{aligned} h_\alpha(x, \omega) = \min \quad & g^\top y \\ & y \in \{y \mid Wy \geq \lceil \omega \rceil_\alpha - T(\omega)x, y \geq 0\}, \end{aligned} \tag{29}$$

where the operation $\lceil \omega \rceil_\alpha := \lceil \omega - \alpha \rceil + \alpha$ is performed on an element-by-element basis. Thus, given $\alpha$, a random variable $\omega$ (which may be continuous) gets mapped to one that assumes values in $\alpha + \mathbb{Z}^{m_2}$. More importantly, the function $h_\alpha(x, \omega)$ is a convex function of $x$ since there are no integrality requirements in

(29). Thus the strategy of convex approximations is to introduce a change of measure to accommodate the integer restrictions of the second stage.

To give the reader a sense of the form of the bound, assume that there is exactly one dual feasible vector, say $\hat{\psi}$. Then consider the difference $h(x,\omega) - h_\alpha(x,\omega)$, or ultimately, the difference $H(x) - H_\alpha(x)$ where $H(x) = \mathbb{E}[h(x,\tilde{\omega})]$ and $H_\alpha(x) = \mathbb{E}[h_\alpha(x,\tilde{\omega})]$. In order to bound the difference, let $\chi_i = t_i^\top x$, and note that

$$
\begin{aligned}
(\hat{\psi}^\top)\left[\lceil \omega - Tx \rceil - (\lceil \omega \rceil_\alpha - Tx)\right] &= \sum_i \hat{\psi}_i \left[\lceil \omega_i - \chi_i \rceil + \chi_i - \lceil \omega_i \rceil_{\alpha_i}\right] \\
&= \sum_i \hat{\psi}_i \left[\lceil \omega_i \rceil_{\chi_i} - \lceil \omega_i \rceil_{\alpha_i}\right].
\end{aligned}
\tag{30}
$$

Hence the difference $H(x) - H_\alpha(x)$ can be calculated by using the expectation of the rightmost term in (30). When the set of dual multipliers is not a singleton as assumed above, one may not obtain the above equality, but a term similar to the above right hand side may be used to bound the difference $H(x) - H_\alpha(x)$. Romeijnders et al. [46] assume that there exists a certain *monotonicity property* of optimal dual multiplier which yields a unique multiplier $\psi^*$ which can be used in place of $\hat{\psi}$ in (30).

Let $|\Delta|f_i$ denote the *total variation on* $\mathbb{R}$ of the $i^{th}$ marginal probability density function (pdf). Then the authors show that $\|H - H_\alpha\|_\infty \leq \sum_{i=1}^m \psi_i^* \max\{\frac{|\Delta|f_i}{8}, 1 - \frac{2}{|\Delta|f_i}\}$. To get a sense of how well these error bounds perform computationally, please refer to [46].

**5.3 Bounds for SMIP with Recourse: Beyond LP Relaxation**  An upper bound for SMIP can be obtained by choosing any MIP-feasible $x \in X \cap \mathcal{X}$ and computing $c^\top x + H(x)$ which provides an upper bound. A common choice is $x = x_E$ which denotes an optimum solution of the expected value problem (i.e., one where the random elements are replaced by their expectations [8]). Hence in this subsection, we mainly focus on lower bounds. While lower bounds in SMIP-RC can be computed by simply using its LP relaxation, one can get stronger lower bounds by solving some deterministic MIP problems. Given the recent progress with deterministic MIP solvers, such bounding schemes can be useful. One such idea is due to [48]. The bounds here are applicable to a fairly general setting of SMIP-RC models, namely $A = RC, B = C = D = \{1, 2\}$. The main reason for this generality is that the basic idea comes from choosing fewer scenarios to create a collection of smaller deterministic equivalent recourse problems, which are then solved to optimality using a deterministic MIP solver. The basic thrust of this approach is to change the probability measure appropriately, while choosing only a subset of scenarios for the approximation. We remind the reader that standard bounding strategies in SLP, such as the well known wait-and-see (WS) lower bound is also applicable to SMIP-RC.

Suppose we are given a finite collection of scenarios, say $\Omega = \{1, \ldots, K\}$, and with each outcome $\omega \in \Omega$ we associate a probability $p(\omega)$. Let the power set of $\Omega$ be denoted $2^K$. Based on a "reference scenario," labelled "0" (with probability $p(0)$) and a subset with $\Gamma \in 2^K$ define $\rho(\Gamma) = \sum_{\omega \in \Gamma} p(\omega)$. Thus $\rho(\Gamma)$ gives the total probability of scenarios in the subset $\Gamma$. Also, note that we are allowed to choose a reference scenario such that $p(0) = 0$. Following [48] define the following aggregation:

$$
z(\Gamma) = \min_{x \in X \cap \mathcal{X}} c^\top x + p(0)h(x, \omega_0) + (1 - p(0))\left[\sum_{\omega \in \Gamma} \frac{p(\omega)}{\rho(\Gamma)} h(x, \omega)\right]
$$

where as usual, $h$ is defined by (2).

It is important to bear in mind that if an instance has MIP requirements in both first and second stage models, the above approximation requires MIP-feasibility in both stages. As a result, so long as the second stage has only a very small number of scenarios, the approximation can be solved using the deterministic equivalent formulation (DEF). If however, the number of scenarios is moderately large, we recommend the adoption of SMIP decomposition methods such as those discussed in the previous sections.

Let $k$ denote an integer which reflects the cardinality of the subsets $\Gamma$ and define the "Expected value of the Group Subproblem Objectives", or $EGSO(k)$ for short, as follows.

$$EGSO(k) := \frac{1}{\binom{K-1}{k-1}(1-p(0))} \left[ \sum_{|\Gamma|=k} \rho(\Gamma)z(\Gamma) \right].$$

One of the main results in [48] is that for any chosen reference scenario, the following holds.

$$WS \leq EGSO(1) \leq \cdots \leq EGSO(K) = \text{Value of RC},$$

where $WS$ denotes the wait-and-see value, and the largest value above yields the value of the Recourse (RC) Problem. The above monotonicity suggests that stronger lower bounds require greater computational effort (in terms of the number of MIPs solved), as expected. Note that it is also appropriate to use the decisions obtained during the lower bounding process to evaluate the objective function value of the model, so that an upper bound can be calculated thus providing a sense of the quality of the proposed solution.

**6. Conclusions**    In this tutorial, we reviewed recent results on stochastic mixed-integer programs with recourse (SMIP-RC), as well their extension to SMIP-RCC. We showed that while these models inherit the challenges of discrete variables in MIPs and uncertainty of data in SLP, we can overcome these challenges by novel decomposition algorithms that integrate knowledge at the confluence of these two seemingly separate fields. We provided preliminary evidence that the algorithms described here are computationally viable. To enable greater penetration of SMIP models arising in various applications, additional algorithmic and computational advances may be necessary. Practical extensions of these algorithms to multi-stage problems will expand their reach further. In this context, one can already envision variants based on the progressive hedging principle [45], together with Lagrangian bounds as in [20]. Such methods are already on the horizon in a forthcoming dissertation by S. Atakan. Finally, while we considered risk in the SMIP-RCC constraints, we restricted our attention to risk-neutral (expected value) objectives. However, we would like to note that the algorithms described in this tutorial can be immediately extended to risk-averse objectives involving convex risk measures, such as mean-risk objectives involving conditional value-at-risk.

# References

[1] A. Abdi and R. Fukasawa. On the mixing set with a knapsack constraint. *Mathematical Programming*, 157(1):191–217, 2016.

[2] S. Ahmed, R. Garcia, N. Kong, L. Ntaimo, G. Parija, F. Qiu, and S. Sen. SIPLIB: A stochastic integer programming test problem library. http://www.isye.gatech.edu/~sahmed/siplib, 2015.

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

[4] F. Barahona, S. Berman, S. Gunluk, and S. Holmes. Robust capacity planning in semiconductor manufacturing. *Naval Research Logistics*, 52:459–468, 2005.

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

[6] P. Beraldi and A. Ruszczyński. A branch and bound method for stochastic integer programs under probabilistic constraints. *Optimization Methods and Software*, 17(3):359–382, 2002.

[7] P. Beraldi and A. Ruszczyński. The probabilistic set covering problem. *Operations Research*, 50(6):956–967, 2002.

[8] J. R. Birge and F. Louveaux. *Introduction to stochastic programming*. Springer Verlag, New York, 1997.

[9] R. Bixby and E. Rothberg. 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.

[10] C. E. Blair and R. G. Jeroslow. The value function of an integer program. *Mathematical Programming*, 23(1):237–273, 1982.

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

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

[13] C. C. Carøe and J. Tind. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 101(2):306–316, 1998.

[14] A. Charnes and W. W. Cooper. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research*, 11(1):18–39, 1963.

[15] A. Charnes, W.W. Cooper, and G. Y. Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4(3):235–263, 1958.

[16] B. Chen, S. Küçükyavuz, and S. Sen. Finite disjunctive programming characterizations for general mixed-integer linear programs. *Operations Research*, 59:202–210, 2011.

[17] B. Chen, S. Küçükyavuz, and S. Sen. A computational study of the cutting plane tree algorithm for general mixed-integer linear programs. *Operations Research Letters*, 40:15–19, 2012.

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

[19] D. Dentcheva, A. Prékopa, and A. Ruszczyński. Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming*, 89(1):55–77, 2000.

[20] D. Gade, G. Hackbeil, S.M. Ryan, R. J-B. Wets, J.-P. Watson, and D.L. Woodruff. Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. *Mathematical Programming*, 157(1):47–67, 2016.

[21] D. Gade and S. Küçükyavuz. Pure cutting-plane algorithms and their convergence. In J. J. Cochran, L. A. Cox, P. Keskinocak, J. P. Kharoufeh, and J. C. Smith, editors, *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc., 2011.

[22] D. Gade, S. Küçükyavuz, and S Sen. Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. *Mathematical Programming*, 144(1-2):39–64, 2014.

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

[24] R. E. Gomory. An algorithm for the mixed integer problem. Technical Report RM-2597, RAND Corporation, 1960.

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

[26] O. Günlük and Y. Pochet. Mixing mixed-integer inequalities. *Mathematical Programming*, 90(3):429–457, 2001.

[27] W. K. Klein Haneveld, L. Stougie, and M. H. van der Vlerk. On the convex hull of the simple integer recourse objective function. *Annals of Operations Research*, 56:209–224, 1995.

[28] W. K. Klein Haneveld, L. Stougie, and M. H. van der Vlerk. An algorithm for the construction of convex hulls in simple integer recourse programming. *Annals of Operations Research*, 64:67–81, 1996.

[29] S. Küçükyavuz. On mixing sets arising in chance-constrained programming. *Mathematical Programming*, 132:31–56, 2012.

[30] A. Langer, R. Venkataraman, U. Palekar, and L. Kale. Parallel branch-and-bound for two-stage stochastic integer optimization. In *High Performance Computing (HiPC), 20th International Conference on*, Bengaluru, India, 2013.

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

[32] M. Lejeune. Pattern-based modeling and solution of probabilistically constrained optimization problems. *Operations Research*, 60(6):1356–1372, 2012.

[33] X. Li, E. Armagan, A. Tomasgard, and P.I. Barton. Stochastic pooling problem for natural gas production network design and operation under uncertainty. *AIChE Journal*, 57(8):2120–2135, 2011.

[34] X. Liu, S. Küçükyavuz, and J. Luedtke. Decomposition algorithms for two-stage chance-constrained programs. *Mathematical Programming*, 157(1):219–243, 2016.

[35] J. Luedtke. A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Mathematical Programming*, 146:219–244, 2014.

[36] J. Luedtke, S. Ahmed, and G. Nemhauser. An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, 122(2):247–272, 2010.

[37] B. L. Miller and H. M. Wagner. Chance constrained programming with joint constraints. *Operations Research*, 13(6):930–965, 1965.

[38] A. Nemirovski and A. Shapiro. Scenario approximation of chance constraints. In G. Calafiore and F. Dabbene, editors, *Probabilistic and Randomized Methods for Design under Uncertainty*, pages 3–48. Springer, 2005.

[39] V. I. Norkin, Y. M. Ermoliev, and A. Ruszczynski. On optimal allocation of indivisibles under uncertainty. *Operations Research*, 46(3):381–395, 1998.

[40] L. Ntaimo and S. Sen. The million variable "march" for stochastic combinatorial optimization. *Journal of Global Optimization*, 32:385–400, 2005.

[41] L. Ntaimo and S. Sen. A comparative study of decomposition algorithms for stochastic combinatorial optimization. *Computational Optimization and Applications*, 40:299–319, 2008.

[42] A. Prékopa. Contributions to the theory of stochastic programming. *Mathematical Programming*, 4(1):202–221, 1973.

[43] A. Prékopa. Dual method for the solution of a one-stage stochastic programming problem with random RHS obeying a discrete probability distribution. *Mathematical Methods of Operations Research*, 34(6):441–461, 1990.

[44] Y. Qi and S. Sen. The ancestral Benders' cutting plane algorithm with multi-term disjunctions for mixed-integer recourse decisions in stochastic programming. *Mathematical Programming*, 161:193–235, 2017.

[45] R.T Rockafellar and R. J-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16:119–147, 1991.

[46] W. Romeijnders, D. P. Morton, and M. H. van der Vlerk. Assessing the quality of convex approximations for two-stage totally unimodular integer recourse models. *INFORMS Journal on Computing*, 29(2):211–231, 2017.

[47] A. Ruszczyński. Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. *Mathematical Programming*, 93(2):195–215, 2002.

[48] B. Sandikci, N. Kong, and A. Schaefer. A hierarchy of bounds for stochastic mixed-integer programs. *Mathematical Programming*, 138:253–272, 2013.

[49] A. Saxena, V. Goyal, and M. A. Lejeune. MIP reformulations of the probabilistic set covering problem. *Mathematical Programming*, 121(1):1–31, 2010.

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

[51] S. Sen. Relaxations for probabilistically constrained programs with discrete random variables. *Operations Research Letters*, 11(2):81–86, 1992.

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

[53] S. Sen. *Stochastic Integer Programming Algorithms: Beyond Benders' Decomposition*. Wiley Handbook in OR/MS, World Wide Web, 2010.

[54] 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(1):1–20, 2005.

[55] S. Sen and H. D. Sherali. On the convergence of cutting plane algorithms for a class of nonconvex mathematical programs. *Mathematical Programming*, 106(2):203–223, 2006.

[56] S. Sen, L. Yu, and T. Genc. A stochastic programming approach to power portfolio optimization. *Operations Research*, 54(1):55–72, 2006.

[57] H. D. Sherali and B. M. P. Fraticelli. 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.

[58] J. Shi and S. S. Oren. Wind power integration through stochastic unit commitment with topology control recourse. In *19th Power Systems Computation Conference, (PSCC 2016)*, Genova, Italy, 2016.

[59] Y Song, J. Luedtke, and S. Küçükyavuz. Chance-constrained binary packing problems. *INFORMS Journal on Computing*, 26(4):735–747, 2014.

[60] M. H. van der Vlerk. *Stochastic programming with integer recourse.* PhD thesis, University of Groningen, The Netherlands, 1995.

[61] M. H. van der Vlerk. Convex approximations for a class of mixed-integer recourse models. *Annals of Operations Research*, 177:139–150, 2010.

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

[63] J. Wang and S. Shen. Risk and energy consumption tradeoffs in cloud computing service via stochastic optimization models. In *Proceedings of the 5th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2012)*, Chicago, IL, 2012.

[64] S. Wang, Y. Guan, and J. Wang. A chance-constrained two-stage stochastic program for unit commitment with uncertain wind power output. *IEEE Transactions on Power Systems*, 27:206–215, 2012.

[65] A. Werner, K. T. Uggen, M. Fodstad, A-G. Lium, and R. Egging. Stochastic mixed-integer programming for integrated portfolio planning in the LNG supply chain. *The Energy Journal*, 35(1):2120–2135, 2011.

[66] L. A. Wolsey. *Integer programming.* Wiley-Interscience, New York, NY, USA, 1998.

[67] S. Xie, X. Li, and Y. Ouyang. Decomposition of general facility disruption correlations via augmentation of virtual supporting stations. *Transportation Research Part B*, 80:64–81, 2015.

[68] M. Zhang and S. Küçükyavuz. Finitely convergent decomposition algorithms for two-stage stochastic pure integer programs. *SIAM Journal on Optimization*, 24(4):1933–1951, 2014.

[69] M. Zhang, S. Küçükyavuz, and S. Goel. A branch-and-cut method for dynamic decision making under joint chance constraints. *Management Science*, 60(5):1317–1333, 2014.

[70] M. Zhao, K. Huang, and B. Zeng. A polyhedral study on chance constrained program with random right-hand side. *Mathematical Programming*, pages 1–46, 2016. Article in Advance. DOI: 10.1007/s10107-016-1103-6.