

Decomposition Algorithms for Two-Stage Chance-Constrained Programs

Xiao Liu ^{*1}, Simge Küçükyavuz ^{†1} and James Luedtke ^{‡2}

¹ *Department of Integrated Systems Engineering, The Ohio State
University*

² *Department of Industrial and Systems Engineering, University of
Wisconsin-Madison*

March 1, 2014

Abstract

We study a class of chance-constrained two-stage stochastic optimization problems where second-stage feasible recourse decisions incur additional cost. In addition, we propose a new model, where “recovery” decisions are made for the infeasible scenarios to obtain feasible solutions to a relaxed second-stage problem. We develop decomposition algorithms with specialized optimality and feasibility cuts to solve this class of problems. Computational results on a chance-constrained resource planing problem indicate that our algorithms are highly effective in solving these problems compared to a mixed-integer programming reformulation and a naive decomposition method.

Keywords— two-stage stochastic programming , chance constraints , Benders decomposition , cutting planes

1 Introduction

Chance-constrained mathematical programs (CCMPs) aim to find optimal solutions to problems where the probability of an undesirable outcome is limited by a given threshold, ϵ . The first optimization problem with disjoint chance constraints is defined by Charnes et al. [10]. Charnes and Cooper [9] establish the deterministic equivalent for chance-constrained programs. Miller and Wagner [16] study the mathematical properties of joint chance-constrained programs with independent random variables. Prékopa [20] studies joint probabilistic

*liu.2738@osu.edu

†kucukyavuz.2@osu.edu

‡jrluedt1@wisc.edu

constraints with dependent random variables and proposes an equivalent deterministic convex program under certain assumptions on the distribution of the random right-hand side.

Luedtke and Ahmed [14] show that for more general distributions, sample-average approximation (SAA) can be applied to find good feasible solutions and statistical bounds to the original CCMPs. Related studies can also be found in [6, 7, 8, 18]. The resulting sampled 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. However, the weakness of the linear programming relaxation of this big-M formulation and its large size make it hard to solve. Luedtke et al. [15] and Küçükyavuz [12] study strong valid inequalities for the deterministic equivalent formulation of chance-constrained problems with random right-hand sides. An alternative reformulation for this class of problems involves using the concept of $(1 - \epsilon)$ -efficient points [21]. Sen [24] studies a disjunctive programming reformulation by using $(1 - \epsilon)$ -efficient points. Dentcheva et al. [11] give reformulations of CCMPs based on the $(1 - \epsilon)$ -efficient points, and obtain valid bounds for the objective value. Beraldi and Ruszczyński [3] propose a branch-and-bound algorithm based on the enumeration of the exponentially many $(1 - \epsilon)$ -efficient points. See also Beraldi and Ruszczyński [2], Ruszczyński [22] and Saxena et al. [23] for algorithms based on the $(1 - \epsilon)$ -efficient points reformulation. For problems with special structures, formulations that do not involve additional binary variables are developed in Song and Luedtke [26] and Song et al. [25].

Most of the earlier work in CCMPs, including the aforementioned studies, can be seen as single-stage (i.e., static) decision-making problems where the decisions are made such that after the uncertain data is realized, there is a low probability of an undesirable outcome. Zhang et al. [32] 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 paper, we study such algorithms for *two-stage* CCMPs, where after the uncertain parameters are revealed, we would like to determine recourse actions that incur additional cost. This is similar to traditional two-stage stochastic programs (*without* chance constraints), where some decisions are made in the first stage before the uncertain parameters are revealed. In the second stage, *recourse* decisions are made to satisfy the second stage problems for all possible scenarios at a minimum total expected cost. Van Slyke and Wets [28] propose the L-shaped decomposition method, which is an adaptation of the Benders decomposition algorithm [1] to such stochastic programs. (See also Birge and Louveaux [5] for a multi-cut implementation.) However, these methods cannot be directly applied to the two-stage CCMP, since both the feasibility and optimality cuts of the Benders method work on the assumption that all second stage problems should be feasible, which is not the case for CCMPs. Luedtke [13] overcomes one of these difficulties by developing a valid “feasibility cut” for a special case of two-stage CCMPs with no additional costs for the second stage variables. A few studies [29, 30] 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. Hence, an interesting research question is whether we can derive strong optimality cuts for two-stage CCMPs. In this paper, we answer this question in the affirmative. In a concurrent work, Zeng et al. [31] propose a decomposition algorithm for two-stage CCMP, which is based on bilinear feasibility and optimality cuts. However, these cuts need to be linearized by adding additional variables which are constrained by “big- M ” inequalities. This yields a significantly larger master problem formulation, and in our experiments we found that this approach also yielded weak lower bounds, although more promising results were reported in Zeng et al. [31] on their test instances.

Similar to a static CCMP, the two-stage CCMP model assumes that we are allowed to ignore the outcomes of a small fraction of the scenarios. This assumption is appropriate when constraint satisfaction is “all or nothing” or if the magnitude of constraint violation is not important, e.g., if this simply results in lost demands. However, in some other problems the magnitude of violation, and not just the probability that a violation occurs, is also important. For example, in an emergency preparedness problem it is important to have a plan that meets all the needs with high probability, but also that does not have excessive shortages in case the plan is not successful. As another example, a power system operator wishes to have a plan in which all energy supply needs are met with high probability, but also wishes to control the amount of shortage in the cases when a shortage occurs. To deal with such problems, we introduce in §2 an alternative model for risk management, which models the need to *recover* from an undesirable outcome. We refer to this model as two-stage CCMP with recovery (CCMPR). We show that a standard two-stage CCMP is a special case of the two-stage CCMPR, and thus algorithms for two-stage CCMPR can be used to solve either problem. In §3 we propose a branch-and-cut based decomposition algorithm for two-stage CCMPR based on optimality cuts that do not involve big- M terms. In §4, we summarize the performance of the proposed decomposition algorithm on a resource planning example. We conclude with §5.

2 Mathematical Models

2.1 Two-Stage Chance-Constrained Programs with Recourse

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. We consider a two-stage problem with first-stage decision variables $x \in X \subseteq \mathbb{R}^{n_1}$, where X is assumed to be a polyhedron representing the deterministic constraints of the problem. For each $x \in X$ and random outcome $\omega \in \Omega$, the second stage problem is defined as:

$$f(x, \omega) := \min_y \{q_\omega^\top y : T_\omega x + W_\omega y \geq h_\omega, y \in \mathbb{R}_+^{n_2}\}. \quad (1)$$

Here, for each $\omega \in \Omega$, T_ω is a $d \times n_1$ matrix, $h_\omega \in \mathbb{R}^d$, W_ω is a $d \times n_2$ matrix, $q_\omega \in \mathbb{R}_+^{n_2}$, and y is the vector of second stage decision variables. We adopt the

convention that $f(x, \omega) = +\infty$ if (1) is infeasible. For any $\omega \in \Omega$, we define

$$P(\omega) = \{x \in X : \exists y \in \mathbb{R}_+^{n_2}, T_\omega x + W_\omega y \geq h_\omega\},$$

as the set of first-stage solutions x for which the second-stage problem (1) has a feasible solution (i.e., $f(x, \omega) < +\infty$ if and only if $x \in P(\omega)$).

Given a cost vector $c \in \mathbb{R}^{n_1}$, the traditional two-stage stochastic program has the form:

$$\min\{c^\top x + \mathbb{E}_\omega[f(x, \omega)] : x \in X\}.$$

This problem implicitly enforces that $x \in P(\omega)$ for almost every $\omega \in \Omega$, since otherwise the objective value is infinite. In contrast, a traditional CCMP without costs in the second stage has the form:

$$\min\{c^\top x : \mathbb{P}\{x \in P(\omega)\} \geq 1 - \varepsilon, x \in X\}.$$

The motivation behind the chance constraint model is that enforcing $x \in P(\omega)$ for almost every $\omega \in \Omega$ may either lead to an infeasible model, or may lead to a model in which the first-stage solutions are too costly. Thus, the constraint that the second-stage must be feasible in all possible realizations is replaced with the relaxed version that enforces this to hold with high probability. This chance-constrained model does not account for the cost of second-stage solutions in the case when they are feasible.

We propose a two-stage CCMP that extends both the traditional two-stage and chance-constrained models. The model uses indicator decision variables \mathbb{I}_ω for $\omega \in \Omega$, where $\mathbb{I}_\omega = 0$ implies that $x \in P(\omega)$:

$$\min_{x, \mathbb{I}} c^\top x + \mathbb{P}(\mathbb{I}_\omega = 0) \mathbb{E}_\omega[f(x, \omega) \mid \mathbb{I}_\omega = 0] \quad (2a)$$

$$\text{s.t. } \mathbb{I}_\omega = 0 \Rightarrow x \in P(\omega), \quad \omega \in \Omega \quad (2b)$$

$$\mathbb{P}\{x \in P(\omega)\} \geq 1 - \varepsilon \quad (2c)$$

$$x \in X, \mathbb{I}_\omega \in \mathbb{B}, \omega \in \Omega. \quad (2d)$$

The idea behind this model is that the recourse model (1) represents the “normal” system operation, which is the desired state. The constraint (2c) together with the logical condition (2b) enforces the chance constraint, which states that the probability that the system has a feasible second stage solution is at least $1 - \varepsilon$. The objective in (2) minimizes the sum of the first-stage costs and the expected second stage costs, averaged only over the outcomes ω for which $\mathbb{I}_\omega = 0$. This ensures that the objective is finite for any feasible solution.

Throughout the paper we make the following assumptions:

- A1:** The random vector ω has finite support, i.e., $\Omega := \{\omega_1, \omega_2, \dots, \omega_m\}$, and each outcome is equally likely ($\mathbb{P}\{\omega = \omega_k\} = 1/m$ for $k = 1, \dots, m$).
- A2:** X and $P(\omega)$ for $\omega \in \Omega$ are non-empty polyhedra;
- A3:** $P(\omega), \omega \in \Omega$ have the same recession cone, i.e., $C := \{r \in \mathbb{R}^n : x + \lambda r \in P(\omega) ; \forall x \in P(\omega), \lambda \geq 0\}$ for all $\omega \in \Omega$;

A4: There does not exist an extreme ray, \tilde{r} , of X with $c^\top \tilde{r} < 0$,

where **(A1)**-**(A3)** directly follow [13]. To simplify notation, let $P_k = P(\omega_k)$, $W_k = W_{\omega_k}$, $h_k = h_{\omega_k}$, $q_k = q_{\omega_k}$, $T_k = T_{\omega_k}$, $z_k = \mathbb{I}_{\omega_k}$, and $f(x, k) = f(x, \omega_k)$ for all $k \in K$, where $K = \{1, 2, \dots, m\}$. Assumption **(A4)** together with $f(x, \omega) \geq 0$ (because $q, y \geq 0$) ensures that there is a bounded optimal solution to the two-stage CCMP.

2.2 Two-Stage Chance-Constrained Programs with Recovery

Model (2) ignores the outcome of scenarios that are not enforced to have a feasible second-stage solution under the normal operations ($\mathbb{I}_\omega = 0$), and instead just enforces that the probability of this selected set of “ignored outcomes” be small. In this section, we extend this model to also include a cost for scenarios in which the normal operation is not enforced to be feasible ($\mathbb{I}_\omega = 1$). The idea is to introduce a secondary “recovery” model, that in some way represents the system operation in cases when we do not operate under the normal operation defined by (1). For example, the recovery problem may relax some of the constraints of the normal operational model, by including decision variables that measure and penalize the magnitude of the violation of such constraints. Formally, for any $\omega \in \Omega$ and $x \in X$, we define the recovery operation problem as follows:

$$\bar{f}(x, \omega) = \min_{\bar{y}} \{ \bar{q}_\omega^\top \bar{y} : \bar{T}_\omega x + \bar{W}_\omega \bar{y} \geq \bar{h}_\omega, \bar{y} \in \mathbb{R}_+^{\bar{n}_2} \} \quad (3)$$

where \bar{T}_ω , \bar{W}_ω are $\bar{d} \times n_1$ and $\bar{d} \times \bar{n}_2$ matrices, respectively, $\bar{h}_\omega \in \mathbb{R}^{\bar{d}}$, $\bar{q} \in \mathbb{R}_+^{\bar{n}_2}$, and \bar{y} is the vector of recovery decisions. Note that the dimension \bar{n}_2 of \bar{y} is not necessarily the same as the dimension n_2 of the recourse decision vector y (similarly for d and \bar{d}). For example, the recovery problem may be identical to the normal recourse problem except for the addition of new recovery decision variables that guarantee a feasible solution always exists.

We then introduce the two-stage chance-constrained problem with recovery (CCMPR) model as follows:

$$\min_{x, \mathbb{I}} c^\top x + \mathbb{P}(\mathbb{I}_\omega = 0) \mathbb{E}_\omega[f(x, \omega) | \mathbb{I}_\omega = 0] + \mathbb{P}(\mathbb{I}_\omega = 1) \mathbb{E}_\omega[\bar{f}(x, \omega) | \mathbb{I}_\omega = 1]$$

$$\text{s.t. } \mathbb{I}_\omega = 0 \Rightarrow x \in P(\omega), \quad \omega \in \Omega \quad (4a)$$

$$\mathbb{P}\{x \in P(\omega)\} \geq 1 - \varepsilon \quad (4b)$$

$$x \in X, \mathbb{I}_\omega \in \mathbb{B}, \omega \in \Omega. \quad (4c)$$

The only difference between the CCMPR and CCMP models is the inclusion of the term $\mathbb{E}_\omega[\bar{f}(x, \omega) | \mathbb{I}_\omega = 1]$ in the objective, which captures the expected cost of the recovery operations, conditioned over the scenarios that are selected to operate in recovery mode.

We make the following assumptions in addition to assumptions **(A1)** - **(A4)**:

B1: $\bar{f}(x, \omega) < +\infty$ for all $x \in X, \omega \in \Omega$.

B2: $f(x, \omega) \geq \bar{f}(x, \omega)$ for all $x \in X, \omega \in \Omega$.

The assumption **(B1)** is analogous to the standard relatively complete recourse assumption used in two-stage stochastic programs, which we apply *only* to the recovery model. The motivation behind assumption **(B2)** is that the recovery operation has a larger feasible region than the normal operation due to the introduction of *additional* recovery actions. The use of these recovery actions is either highly undesirable, or possibly not even physically meaningful (e.g., if they are just used to measure constraint violation), and so the chance constraint enforces that in most scenarios they should not be used. However, when they are allowed to be used, all the operations of the normal recourse model are still feasible, and hence the cost in the recovery operation can only be smaller. On the other hand, if the recovery actions are not allowed to be used, then the recovery model reduces to the normal model, and incurs the same cost.

Based on assumption **(A1)**, we once again simplify notation by letting $\bar{W}_k = \bar{W}_{\omega_k}$, $\bar{h}_k = \bar{h}_{\omega_k}$, $\bar{q}_k = \bar{q}_{\omega_k}$, $\bar{T}_k = \bar{T}_{\omega_k}$, and $\bar{f}(x, k) = \bar{f}(x, \omega_k)$ for all $k \in K$. Using **(A1)** and this new notation, we can re-write (4) as:

$$\min_{x, z} c^\top x + \frac{1}{m} \left(\sum_{k=1}^m (1 - z_k) f(x, k) + \sum_{k=1}^m z_k \bar{f}(x, k) \right) \quad (5a)$$

$$\text{s.t. } z_k = 0 \Rightarrow x \in P_k, \quad k \in K \quad (5b)$$

$$\sum_{k=1}^m z_k \leq p \quad (5c)$$

$$x \in X, z \in \mathbb{B}^m, \quad (5d)$$

where $p := \lfloor m\epsilon \rfloor$, and (5c) represents the chance constraint (2c). Throughout the paper, we adopt the convention that $0 \times \infty = 0$.

The deterministic equivalent formulation for two-stage CCMPR (5) is then:

$$\min_{x, y, \bar{y}, z} c^\top x + \frac{1}{m} \sum_{k=1}^m q_k^\top y_k + \frac{1}{m} \sum_{k=1}^m \bar{q}_k^\top \bar{y}_k \quad (6a)$$

$$\text{s.t. } T_k x + W_k y_k + M'_k z_k \geq h_k, \quad k \in K \quad (6b)$$

$$\bar{T}_k x + \bar{W}_k \bar{y}_k + \bar{M}_k (1 - z_k) \geq \bar{h}_k, \quad k \in K \quad (6c)$$

$$\sum_{k=1}^m z_k \leq p \quad (6d)$$

$$x \in X, y \in \mathbb{R}_+^{n_2 \times m}, \bar{y} \in \mathbb{R}_+^{\bar{n}_2 \times m}, z \in \mathbb{B}^m, \quad (6e)$$

where $M'_k, k \in K$ is a vector of sufficiently large numbers to make (6b) redundant when z_k equals to 1, assuming it exists (e.g., when X is compact). Similarly, $\bar{M}_k, k \in K$ is a vector of sufficiently large numbers to make (6c) redundant when $z_k = 0$ (assuming it exists). Nonnegativity of the coefficient vector q_k , together with (6b) and nonnegativity of the recourse variables y imply that when $z_k = 1$

the normal operation cost term $(q_k^\top y_k)$ will be zero, and similarly when $z_k = 0$ the recovery operation cost term $(\bar{q}_k^\top \bar{y}_k)$ will be zero. In this paper, we give a decomposition algorithm with the aim of avoiding the constraints (6b)-(6c), because they lead to weak LP relaxations and also because of the introduction of a large number of variables $y_k, \bar{y}_k, k \in K$.

Remark 1. *In another line of work, an alternative risk-averse two-stage optimization problem is defined where the expectation term in (2a) is replaced with the conditional value-at-risk (CVaR) (see, for example Miller and Ruszczyński [17] and Noyan [19]). Under the assumption that each scenario is equally likely and that $\epsilon = \frac{p}{m}$, such a model would minimize the sum of p worst outcomes (using the representation of CVaR in Bertsimas and Brown [4]) and hence will not lead to solutions with second-stage infeasibility for any scenario. In contrast, model (2) minimizes the sum of $m - p$ best outcomes and allows the remaining outcomes to be infeasible with the assumption that such outcomes can be ignored.*

2.2.1 Special cases

We first observe that the two-stage CCMP model (2) is a special case of the two-stage CCMPR model (4), by setting $\bar{f}(x, \omega) \equiv 0$ for all $x \in X$ and $\omega \in \Omega$. In this case, because $f(x, \omega) \geq 0$ for all $x \in X, \omega \in \Omega$, we immediately have that assumptions **(B1)** and **(B2)** are satisfied.

Another interesting special case of two-stage CCMPR is a penalty-based model. We refer to this class of problems as two-stage CCMP with simple recovery. In this case, the recovery model takes the form:

$$\bar{f}(x, \omega) = \min_{y, u} \{q_\omega^\top y + w_\omega^\top u : T_\omega x + W_\omega y + D_\omega u \geq h_\omega, y \in \mathbb{R}_+^{n_2}, u \in \mathbb{R}_+^{n'_2}\}, \quad (7)$$

where $T_\omega, W_\omega, h_\omega$, and q_ω are the data associated with the *normal* operation problem under outcome ω . The vector of decision variables $u \in \mathbb{R}_+^{n'_2}$ can be interpreted as slack variables that are introduced to ensure that (7) always has a feasible solution. Thus, the full set of recovery variables is $\bar{y} = (y, u)$, with dimension $\bar{n}_2 = n_2 + n'_2$. The use of the slack variables u is penalized in the objective with the nonnegative cost vector w_ω . D_ω is a $d \times n'_2$ matrix. For example, feasibility of (7) can be guaranteed by taking $n'_2 = d$ and D_ω to be a $d \times d$ identity matrix. To simplify notation, we let $\bar{W}_\omega = (W_\omega, D_\omega)$ and $\bar{q}_\omega = (q_\omega, w_\omega)$. Hence, if a constraint is violated in the normal model, then the corresponding slack variable equals the shortfall. If D_ω is chosen such that (7) is feasible for any $x \in X, \omega \in \Omega$, then assumption **(B1)** is satisfied. We next show that assumption **(B2)** is also satisfied, and therefore this is a special case of the two-stage CCMPR model (4). For any $x \in X \setminus P(\omega)$, $f(x, \omega) = +\infty$, so the assumption trivially holds. Now consider any $\hat{x} \in P(\omega)$ and \hat{y} , the optimal second stage solution with objective $f(\hat{x}, \omega)$. So we have $T_\omega \hat{x} + W_\omega \hat{y} \geq h_\omega$. Then according to (7), the vector $(\hat{y}, 0)$ is a feasible solution (7). Hence, $f(\hat{x}, \omega) = q_\omega^\top \hat{y}_\omega = \bar{q}_\omega^\top (\hat{y}_\omega, 0) \geq \bar{f}(x, \omega)$, since $(\hat{y}, 0)$ is a feasible solution to the recovery problem. In Section 4 we will report computational experiments with two-stage CCMPR on this special case.

2.2.2 Two-stage consistency

Takriti and Ahmed [27] observe that in two-stage stochastic programs with an objective to minimize a weighted sum of the expected cost and some measure of the variability of costs, it is possible to obtain a model in which a second-stage solution obtained when solving a deterministic equivalent of the two-stage problem is *not* an optimal solution of the second-stage problem. They argue that this inconsistency makes such models inappropriate, and in the two-stage stochastic programming setting they provide conditions on the variability measure that assure this inconsistency does not occur.

In our two-stage model, for a given first-stage solution $x \in X$ and scenario $\omega \in \Omega$, the key question is whether we will operate in the normal model of (1), or in the recovery model (3). The two-stage model introduces decision variables \mathbb{I}_ω to distinguish these cases. However, the constraints (5b) only enforce that if $\mathbb{I}_\omega = 0$ then the normal operation is feasible, and do not enforce the converse that $\mathbb{I}_\omega = 0$ whenever the normal operation is feasible. Thus, the CCMPR model allows the possibility to operate some scenarios in recovery mode even when they could feasibly be operated with the normal recourse model. Thus, when we are actually solving the second-stage problem for a given $x \in X$ and $\omega \in \Omega$, it may be ambiguous which model should be solved. If the normal model is infeasible, then it is clear that we must solve the recovery model. However, if the normal model is feasible, then we need to have a policy to determine whether this is one of the outcomes that is selected to be operated in the recovery mode.

Model (5) determines which outcomes will operate according to the recovery model with a threshold policy. The decision maker first attempts to solve the normal operation problem (1). If it is infeasible, the recovery model is implemented. Otherwise, the decision-maker compares the cost of the normal operation $f(x, \omega)$ to a threshold v^* . If the cost exceeds v^* , the decision-maker chooses to operate in recovery mode, otherwise the decision-maker implements the optimal normal operation decision. The value of v^* can be obtained from the optimal solution of the CCMPR model (5) by setting $v^* = \max\{f(x, k) : z_k = 0\}$. By construction, this policy of operation in the second-stage is consistent with the solution obtained in the CCMPR model, and for the given sample yields a solution that is feasible to the normal model in at least $1 - \epsilon$ fraction of the scenarios. This policy generalizes the traditional use of a chance constraint, in which the “recovery” model amounts to just ignoring the scenario, and this “recovery” option is only used when the normal model is infeasible. If we adopt the convention that an infeasible problem has infinite cost, then the traditional chance constraint model operates in recovery mode only when the cost of the normal model is infinite. Our model instead operates in recovery mode only when the cost of normal operation exceeds a finite value, v^* .

3 Decomposition algorithm for solving two-stage CCMP with Recovery

In this section we propose a decomposition algorithm for two-stage CCMP. We begin this subsection by analyzing the structure of optimal solutions for (5).

Proposition 1. *There exists an optimal solution (x^*, z^*) of (5) in which $\sum_{k=1}^m z_k^* = p$.*

Proof. This follows immediately from assumption **(B2)**. □ □

In other words, there exists an optimal solution where we operate under the “normal” mode for exactly $m - p$ scenarios. Thus, we can replace (5c) with the constraint $\sum_{k=1}^m z_k = p$.

To introduce the branch-and-cut algorithm, we first define the following sets which will be approximated via cuts:

$$F = \{x \in \mathbb{R}^{n_1}, z \in \mathbb{B}^m : \sum_{k=1}^m z_k = p, z_k = 0 \Rightarrow x \in P_k, k \in K\},$$

$$Z = \{(x, z, \theta) \in F \times \mathbb{R}_+ : \theta \geq \frac{1}{m} \sum_{k=1}^m ((1 - z_k)f(x, k) + z_k \bar{f}(x, k))\}. \quad (8)$$

With this notation, solving the problem $\min\{c^\top x + \theta : x \in X, (x, z, \theta) \in Z\}$, where θ is the variable which is used to represent the expected second-stage costs (from both normal and recovery operations), gives an optimal solution (x^*, z^*) to (5) satisfying Proposition 1.

The branch-and-cut algorithm we propose works with the following master problem:

$$\text{RP}(K_0, K_1) = \min_{x, z, \theta} c^\top x + \theta \quad (9a)$$

$$\text{s.t. } \sum_{k=1}^m z_k = p \quad (9b)$$

$$(x, z) \in \mathcal{C} \quad (9c)$$

$$(x, z, \theta) \in \mathcal{D} \quad (9d)$$

$$x \in X, z \in [0, 1]^m, \theta \geq 0 \quad (9e)$$

$$z_k = 0, k \in K_0; z_k = 1, k \in K_1, \quad (9f)$$

where the sets K_0 and K_1 satisfy $K_0 \cap K_1 = \emptyset$ and represent the sets of variables z_k fixed to zero and to one, respectively, during the branch and bound process. The set \mathcal{C} is a polyhedral outer approximation of F , which is defined by feasibility cuts of the following form:

$$\alpha_1^\top x + \delta_1^\top z \geq \beta_1. \quad (10)$$

Here α_1 and δ_1 are n_1 and m dimensional coefficient vectors, respectively, and $\beta_1 \in \mathbb{R}$. The set \mathcal{D} is a polyhedral outer approximation of Z , which is defined by optimality cuts of the form:

$$\theta + \delta^\top z \geq \beta - \alpha^\top x, \quad (11)$$

where α and δ are n_1 and m dimensional coefficient vectors, respectively, and $\beta \in \mathbb{R}$.

The key of the decomposition approach is to derive strong valid inequalities for the sets F and Z . A class of strong valid feasibility cuts has been proposed by Luedtke [13] based on the so-called mixing set. We use the same class of cuts in the present paper. However, Luedtke [13] does not consider second-stage costs, and therefore does not require any optimality cuts. In this paper, we focus on obtaining strong optimality cuts for two-stage CCMPRs.

We first describe a naive way to obtain valid optimality cuts (11). Let (\hat{x}, \hat{z}) be such that $\hat{x} \in X$, $\hat{z} \in \mathbb{B}^m$, and \hat{z} satisfies (9b). If there exists $k \in K$ with $\hat{z}_k = 0$ and $\hat{x} \notin P_k$, then this solution violates the logical condition (5b), and hence we seek and add a feasibility cut. Otherwise, for each $k \in K$ with $\hat{z}_k = 0$, we solve the corresponding normal operation subproblem (which is now feasible by assumption):

$$\min_{y \in \mathbb{R}_+^{n_2}} \{q_k^\top y : W_k y \geq h_k - T_k \hat{x}\} = \max_{\pi \in \mathbb{R}_+^d} \{\pi^\top (h_k - T_k \hat{x}) : \pi^\top W_k \leq q_k\} \quad (12)$$

and let π_k be an optimal dual solution. Also let Π_k be the set of dual extreme points in (12). In addition, for each $k \in K$ such that $\hat{z}_k = 1$ we solve the recovery problem

$$\min_{\bar{y} \in \mathbb{R}_+^{n_2}} \{\bar{q}_k^\top \bar{y} : \bar{W}_k \bar{y} \geq \bar{h}_k - \bar{T}_k \hat{x}\} = \max_{\bar{\pi} \in \mathbb{R}_+^d} \{\bar{\pi}^\top (\bar{h}_k - \bar{T}_k \hat{x}) : \bar{\pi}^\top \bar{W}_k \leq \bar{q}_k\} \quad (13)$$

and let $\bar{\pi}_k$ be an optimal dual solution (let $\bar{\pi}_k = 0$ for problem (2) without recovery). Also let $\bar{\Pi}_k$ be the set of dual extreme points in (13). Then, if we let $S(\hat{z}) = \{k \in K : \hat{z}_k = 0\}$ and $\bar{S}(\hat{z}) = K \setminus S(\hat{z})$, we obtain the following optimality cut, which is valid for Z :

$$\theta + \sum_{k \in S(\hat{z})} M_k z_k + \sum_{k \in \bar{S}(\hat{z})} M_k (1 - z_k) \geq \frac{1}{m} \left(\sum_{k \in S(\hat{z})} \pi_k^\top (h_k - T_k x) + \sum_{k \in \bar{S}(\hat{z})} \bar{\pi}_k^\top (\bar{h}_k - \bar{T}_k x) \right), \quad (14)$$

where $M_k, k \in K$ is assumed to be large enough so that inequality (14) is redundant whenever $z_k = 1$ for some $k \in S(\hat{z})$ or $z_k = 0$ for some $k \in \bar{S}(\hat{z})$ (let $M_k = 0$ for $k \in \bar{S}(\hat{z})$ for problem (2) without recovery). We refer to inequality (14) as the big-M optimality cut. To see its validity, observe that for the solution (\hat{x}, \hat{z}) , this cut gives a correct lower bound on the second-stage costs (normal

and recovery). For any other solution, we must have at least one $k \in S(\hat{z})$ with $z_k = 1$ or $k \in \bar{S}(\hat{z})$ with $z_k = 0$, and hence inequality (14) is valid for large enough M_k . Note that additional assumptions may be necessary to ensure the existence of such M_k . Once again, our goal in this paper is to avoid the use of such big-M cuts.

An alternative approach, recently proposed by Zeng et al. [31] for problems without recovery, is to use a *bilinear* cut of the form:

$$\theta \geq \frac{1}{m} \left(\sum_{k \in S(\hat{z})} (1 - z_k) \pi_k^\top (h_k - T_k x) + \sum_{k \in \bar{S}(\hat{z})} z_k \bar{\pi}_k^\top (\bar{h}_k - \bar{T}_k x) \right). \quad (15)$$

To use this cut in a branch-and-cut algorithm, the bilinear terms ($z_k x_j$ for $k \in K$, $j = 1, \dots, n_1$) need to be linearized by adding additional variables s_{kj} and inequalities to enforce $s_{kj} = z_k x_j$. We experimented with this approach but on our test instances we found its performance to be comparable to the performance of the basic method using the cuts (14), and so we do not explore this approach further in this work. We think, however, that investigating the integration of these different techniques would be an interesting subject of future study.

3.1 Strong optimality cuts for two-stage CCMPR

In this section we derive valid optimality cuts that are stronger than the big-M optimality cuts (14). First, we define a secondary subproblem associated with the normal recourse problem for a given $\alpha \in \mathbb{R}^{n_1}$ and $k \in K$:

$$v_k(\alpha) = \min\{f(x, k) + \alpha^\top x : x \in P_k\} \quad (16)$$

so that, by definition,

$$f(x, k) \geq v_k(\alpha) - \alpha^\top x, \quad \forall x \in P_k. \quad (17)$$

Problem (16) always has a feasible solution from assumption **(A2)**. Let $\mathbf{dom} v_k = \{\alpha \in \mathbb{R}^{n_1} : v_k(\alpha) > -\infty\}$ be the domain of v_k .

Similarly, we define a secondary subproblem associated with the recovery problem for a given $\alpha \in \mathbb{R}^{n_1}$ and $k \in K$:

$$\bar{v}_k(\alpha) = \min\{\bar{f}(x, k) + \alpha^\top x : x \in X\} \quad (18)$$

so that:

$$\bar{f}(x, k) \geq \bar{v}_k(\alpha) - \alpha^\top x, \quad \forall x \in X. \quad (19)$$

Problem (18) always has a feasible solution from assumption **(A2)**. Let $\mathbf{dom} \bar{v}_k$ be the domain of \bar{v}_k .

We make the following additional assumption:

B3: There exists $D \subseteq \mathbb{R}^{n_1}$ such that $\mathbf{dom} v_k = \mathbf{dom} \bar{v}_k = D$ for all $k \in K$.

Assumption **(B3)** is satisfied with $D = \mathbb{R}^{n_1}$ if X is bounded. In our example application in Section 4, this assumption is satisfied with $D = \mathbb{R}_+^{n_1}$.

Next, for each $k \in K$, we define the following set:

$$Z_k = \{x \in X, z_k \in \mathbb{B}, \eta_k \in \mathbb{R}_+ : \eta_k \geq (1 - z_k)f(x, k) + z_k\bar{f}(x, k)\},$$

where η_k represents the objective function value of the second-stage problem for scenario k . Using the relationship $\theta = (1/m) \sum_{k \in K} \eta_k$, valid inequalities for the sets Z_k , $k \in K$ can be used to obtain valid inequalities for the set Z (optimality cuts).

Proposition 2. *Let $k \in K$, $\pi \in \Pi_k$, and $\alpha = T_k^\top \pi$. Then $\bar{v}_k(\alpha) > -\infty$ and the following inequality is valid for Z_k :*

$$\eta_k + (\pi^\top h_k - \bar{v}_k(\alpha))z_k \geq \pi^\top h_k - \alpha^\top x. \quad (20)$$

Proof. Let $(x, z_k, \eta_k) \in Z_k$. If $z_k = 0$, then

$$\eta_k \geq f(x, k) \geq \pi^\top h_k - \pi^\top T_k x = \pi^\top h_k - \alpha^\top x$$

by weak duality. It also follows then that

$$v_k(\alpha) \geq \min\{\pi^\top h_k - \alpha^\top x + \alpha^\top x : x \in P_k\} = \pi^\top h_k,$$

which shows that $\alpha \in \mathbf{dom} v_k = D$, and thus $\bar{v}_k(\alpha) > -\infty$ by assumption **(B3)**. Finally, if $z_k = 1$, then $\eta_k \geq \bar{f}(x, k) \geq \bar{v}_k(\alpha) - \alpha^\top x$ by (19). \square \square

We can obtain an analogous cut from dual solutions of the recovery problem.

Proposition 3. *Let $k \in K$, $\bar{\pi} \in \bar{\Pi}_k$, and $\alpha = \bar{T}_k^\top \bar{\pi}$ ($\bar{\pi} = \alpha = 0$ for problem (2) without recovery). Then $v_k(\alpha) > -\infty$ and the following inequality is valid for Z_k :*

$$\eta_k + (\bar{\pi}^\top \bar{h}_k - v_k(\alpha))(1 - z_k) \geq \bar{\pi}^\top \bar{h}_k - \alpha^\top x. \quad (21)$$

Proof. Analogous to the proof of proposition 2. \square \square

We next discuss how we obtain the dual solutions used in inequality (20) or (21). Suppose that we have a solution (\hat{x}, \hat{z}) and that $\hat{z}_k = 0$ for some scenario $k \in K$. Then, we attempt to solve the normal operation problem (12). If it is infeasible (i.e., $\hat{x} \notin P_k$), we must add a feasibility cut. If it is feasible, then we choose π in (20) to be an optimal dual solution. Then, by construction, when $\hat{z}_k = 0$, (20) enforces $\eta_k \geq f(\hat{x}, k)$. If $\hat{z}_k = 1$ for some scenario $k \in K$, then we solve the recovery problem (13) and choose $\bar{\pi}$ in (21) to be an optimal dual solution. Again, this choice enforces $\eta_k \geq \bar{f}(\hat{x}, k)$ when $\hat{z}_k = 1$.

We can use inequalities (20) and (21) in a multi-cut implementation of a Benders-type decomposition algorithm. For a single cut implementation, we have the following corollary.

Corollary 4. Let $S \subseteq K$, $\pi_k \in \Pi_k$ and $\alpha_k = T_k^\top \pi_k$ for $k \in S$, and $\bar{\pi}_k \in \bar{\Pi}_k$ and $\alpha_k = \bar{T}_k^\top \bar{\pi}_k$ ($\bar{\pi}_k = \alpha_k = 0$ for problem (2) without recovery) for $k \in K \setminus S$. Then, the following inequality is valid for Z :

$$\begin{aligned} \theta + \frac{1}{m} \sum_{k \in S} (\pi_k^\top h_k - \bar{v}_k(\alpha_k)) z_k + \frac{1}{m} \sum_{k \in K \setminus S} (\bar{\pi}_k^\top \bar{h}_k - v_k(\alpha_k))(1 - z_k) \\ \geq \frac{1}{m} \sum_{k \in S} \pi_k^\top h_k + \frac{1}{m} \sum_{k \in K \setminus S} \bar{\pi}_k^\top \bar{h}_k - \frac{1}{m} \sum_{k=1}^m \alpha_k^\top x. \end{aligned} \quad (22)$$

For a given a solution (\hat{x}, \hat{z}) with $\hat{x} \in X$ and $\hat{z} \in \mathbb{B}^m$, and such that the logical constraints (5b) are satisfied, we choose $S = \{k \in K : \hat{z}_k = 0\}$ and for $k \in S$ choose π_k to be an optimal dual solution of (12), and for $k \in K \setminus S$ choose $\bar{\pi}_k$ to be an optimal dual solution of (13).

Next we derive another class of optimality cuts for two-stage CCMPR.

Theorem 5. Let $S \subset K$ have $|S| = m - p$, $\pi_k \in \Pi_k$ and $\alpha_k = T_k^\top \pi_k$ for $k \in S$, and $\bar{\pi}_k \in \bar{\Pi}_k$ and $\alpha_k = \bar{T}_k^\top \bar{\pi}_k$ ($\bar{\pi}_k = \alpha_k = 0$ for problem (2) without recovery) for $k \in K \setminus S$. Also define $v_*(\alpha_k) = \min\{v_{k'}(\alpha_k) : k' \in K \setminus S\}$ and $\bar{v}_*(\alpha_k) = \min\{\bar{v}_{k'}(\alpha_k) : k' \in S\}$. Then, the following inequality is valid for Z :

$$\begin{aligned} \theta \geq \frac{1}{m} \sum_{k \in S} (\pi_k^\top h_k - \alpha_k^\top x + (v_*(\alpha_k) - \pi_k^\top h_k) z_k) \\ + \frac{1}{m} \sum_{k \in K \setminus S} (\bar{\pi}_k^\top \bar{h}_k - \alpha_k^\top x + (\bar{v}_*(\alpha_k) - \bar{\pi}_k^\top \bar{h}_k)(1 - z_k)). \end{aligned} \quad (23)$$

Proof. Let $(x, z, \theta) \in Z$. First, note, from **(B3)**, that $v_*(\alpha_k)$ and $\bar{v}_*(\alpha_k)$ are well-defined since $\alpha_k \in D$ for $k \in K$. We prove the two inequalities:

$$\sum_{k=1}^m f(x, k)(1 - z_k) \geq \sum_{k \in S} (\pi_k^\top h_k - \alpha_k^\top x + (v_*(\alpha_k) - \pi_k^\top h_k) z_k), \quad (24)$$

$$\sum_{k=1}^m \bar{f}(x, k) z_k \geq \sum_{k \in K \setminus S} (\bar{\pi}_k^\top \bar{h}_k - \alpha_k^\top x + (\bar{v}_*(\alpha_k) - \bar{\pi}_k^\top \bar{h}_k)(1 - z_k)) \quad (25)$$

which establishes the claim from the definition of Z in (8).

Let $U = \{k \in K : z_k = 0\}$. By (9b), we have $|U| = m - p$, and so $|S \setminus U| = |U \setminus S|$. Let $\sigma : U \setminus S \rightarrow S \setminus U$ be a one-to-one mapping between $U \setminus S$

and $S \setminus U$, i.e., $\{\sigma_k : k \in U \setminus S\} = S \setminus U$. Then, using (17) we obtain

$$\begin{aligned}
\sum_{k=1}^m f(x, k)(1 - z_k) &= \sum_{k \in U} f(x, k) \\
&= \sum_{k \in U \cap S} f(x, k) + \sum_{k \in U \setminus S} f(x, k) \\
&\geq \sum_{k \in U \cap S} (\pi_k^\top h_k - \alpha_k^\top x) + \sum_{k \in U \setminus S} f(x, k) \\
&\geq \sum_{k \in U \cap S} (\pi_k^\top h_k - \alpha_k^\top x) + \sum_{k \in U \setminus S} (v_k(\alpha_{\sigma_k}) - \alpha_{\sigma_k}^\top x) \\
&\geq \sum_{k \in U \cap S} (\pi_k^\top h_k - \alpha_k^\top x) + \sum_{k \in U \setminus S} (v_*(\alpha_{\sigma_k}) - \alpha_{\sigma_k}^\top x) \quad (26) \\
&= \sum_{k \in U \cap S} (\pi_k^\top h_k - \alpha_k^\top x) + \sum_{k \in S \setminus U} (v_*(\alpha_k) - \alpha_k^\top x) \quad (27) \\
&= \sum_{k \in S} (\pi_k^\top h_k - \alpha_k^\top x + (v_*(\alpha_k) - \pi_k^\top h_k)z_k),
\end{aligned}$$

where (26) follows because $U \setminus S \subseteq K \setminus S$ and therefore $v_k(\alpha_{\sigma_k}) \geq v_*(\alpha_{\sigma_k})$ and (27) follows from the definition of one-to-one mapping σ . This establishes (24).

The arguments for (25) are similar. Let $\bar{U} = \{k \in K : z_k = 1\}$ so that $|\bar{U}| = p$ and so $|S \setminus \bar{U}| = |\bar{U} \setminus S|$. Let $\bar{\sigma} : \bar{U} \setminus S \rightarrow S \setminus \bar{U}$ be a one-to-one mapping between $\bar{U} \setminus S$ and $S \setminus \bar{U}$. Then, using (19), we obtain

$$\begin{aligned}
\sum_{k=1}^m \bar{f}(x, k)z_k &= \sum_{k \in \bar{U}} \bar{f}(x, k) \\
&\geq \sum_{k \in \bar{U} \cap S} (\bar{\pi}_k^\top \bar{h}_k - \alpha_k^\top x) + \sum_{k \in \bar{U} \setminus S} \bar{f}(x, k) \\
&\geq \sum_{k \in \bar{U} \cap S} (\bar{\pi}_k^\top \bar{h}_k - \alpha_k^\top x) + \sum_{k \in \bar{U} \setminus S} (\bar{v}_k(\alpha_{\bar{\sigma}_k}) - \alpha_{\bar{\sigma}_k}^\top x) \\
&\geq \sum_{k \in \bar{U} \cap S} (\bar{\pi}_k^\top \bar{h}_k - \alpha_k^\top x) + \sum_{k \in \bar{U} \setminus S} (\bar{v}_*(\alpha_{\bar{\sigma}_k}) - \alpha_{\bar{\sigma}_k}^\top x) \\
&= \sum_{k \in \bar{U} \cap S} (\bar{\pi}_k^\top \bar{h}_k - \alpha_k^\top x) + \sum_{k \in S \setminus \bar{U}} (\bar{v}_*(\alpha_k) - \alpha_k^\top x) \\
&= \sum_{k \in K \setminus S} (\bar{\pi}_k^\top \bar{h}_k - \alpha_k^\top x + (\bar{v}_*(\alpha_k) - \bar{\pi}_k^\top \bar{h}_k)(1 - z_k))
\end{aligned}$$

which establishes (25). \square \square

Given a solution (\hat{x}, \hat{z}) with $\hat{x} \in X$ and $\hat{z} \in \mathbb{B}^m$, and such that the logical constraints (5b) are satisfied, in Theorem 5 we use $S = \{k \in K : \hat{z}_k = 0\}$, and for $k \in S$ we choose π_k to be an optimal dual solution of (12), and for $k \in K \setminus S$ we

choose $\bar{\pi}_k$ to be an optimal dual solution of (13). It is easy to see that if $(\hat{x}, \hat{z}) \in F$, with this choice, (23) enforces $\theta \geq \frac{1}{m} \sum_{k \in K} (f(\hat{x}, k)(1 - \hat{z}_k) + \bar{f}(\hat{x}, k)\hat{z}_k)$.

Based on numerical experiments presented in §4.2, inequality (23) provides a stronger lower bound than optimality cut (22) with faster convergence rate, however, at the cost of solving $2p(m-p)$ single scenario subproblems in order to obtain the values $v_*(\alpha_k)$ for each $k \in S$ and $\bar{v}_*(\alpha_k)$ for each $k \in K \setminus S$. Hence, a strategy is to combine the optimality cuts (22) and (23).

Note that for the special case without recovery given by (2), the last term in inequality (23) is eliminated. In this case, we need to solve only $p(m-p)$ secondary subproblems to obtain the values $v_*(\alpha_k)$ for $k \in S$.

3.2 Strong optimality cut for random right-hand sides problem

Inequalities (22) and (23) are valid for two-stage CCMPR for which the randomness appears only in the right-hand-side vectors $h_\omega, \bar{h}_\omega, \omega \in \Omega$. However, we can take advantage of the special structure of this class of problems to obtain optimality cuts with less effort.

For (12) and (13) with $T_k = T, W_k = W, \bar{T}_k = \bar{T}$ and $\bar{W}_k = \bar{W}$ for all $k \in K$, the corresponding dual subproblems share the same polyhedron for all $k \in K$ and hence the same dual extreme point sets Π for the dual of (12) and $\bar{\Pi}$ for the dual of (13). Furthermore, for each $\pi \in \Pi$ and $k \in K$, we have

$$f(x, k) \geq \pi^\top h_k - \pi^\top T x, \quad \forall x \in P_k$$

and for each $\bar{\pi} \in \bar{\Pi}$ and $k \in K$

$$\bar{f}(x, k) \geq \bar{\pi}^\top \bar{h}_k - \bar{\pi}^\top \bar{T} x, \quad \forall x \in X.$$

Note that the second stage value approximation function generated by the same dual extreme point $\pi \in \Pi$ and $\bar{\pi} \in \bar{\Pi}$ for different scenarios are parallel planes.

Let $S \subseteq K$ have $|S| = m - p$ (e.g., $S = \{k \in K : \hat{z}_k = 0\}$ for some solution \hat{z}), and let $G_k(\pi) = \pi^\top h_k$ for $k \in S$. In addition, define $G_*(\pi) = \min\{G_k(\pi) : k \in K \setminus S\}$, for $\pi \in \Pi$. Also let $\bar{G}_k(\bar{\pi}) = \bar{\pi}^\top \bar{h}_k$ for $k \in K \setminus S$. Similarly, define $\bar{G}_*(\bar{\pi}) = \min\{\bar{G}_k(\bar{\pi}) : k \in S\}$, for $\bar{\pi} \in \bar{\Pi}$. For each $k \in S$, let π_k be an optimal dual solution to (12) and for each $k \in K \setminus S$, let $\bar{\pi}_k$ be an optimal dual solution to (13) ($\bar{\pi}_k = 0$ for problem (2) without recovery). Then the proposed optimality cut is :

$$\begin{aligned} \theta \geq & \frac{1}{m} \sum_{k \in S} (G_k(\pi_k) - \pi_k^\top T x + (G_*(\pi_k) - G_k(\pi_k))z_k) \\ & + \frac{1}{m} \sum_{k \in K \setminus S} (\bar{G}_k(\bar{\pi}_k) - \bar{\pi}_k^\top \bar{T} x + (\bar{G}_*(\bar{\pi}_k) - \bar{G}_k(\bar{\pi}_k))(1 - z_k)). \end{aligned} \quad (28)$$

Theorem 6. *Inequality (28) is valid for Z in the case when the randomness occurs only in the right-hand-side.*

Proof. Based on the definition of $G_k(\pi_k)$, we have:

$$f(x, k) \geq G_k(\pi_k) - \pi_k^\top T x, \quad \forall x \in P_k.$$

In addition, we have:

$$\bar{f}(x, k) \geq \bar{G}_k(\bar{\pi}_k) - \bar{\pi}_k^\top \bar{T} x, \quad \forall x \in \bar{P}_k.$$

Let $\alpha_k = T^\top \pi_k$ for $k \in S$, and $\alpha_k = \bar{T}^\top \bar{\pi}_k$ for $k \in K \setminus S$. Then, the rest of the proof is identical to the validity of (23), where the role of $v_k(\alpha)$ in (17) is replaced by $G_k(\pi_k)$, and the role of $\bar{v}_k(\bar{\alpha})$ in (19) is replaced by $\bar{G}_k(\bar{\pi}_k)$. \square \square

Deriving inequality (28) requires only $O(mp)$ vector multiplications in contrast to $O(mp)$ optimization subproblems, so it requires less computational effort.

3.3 Decomposition algorithm for two-stage CCMPRs

In this subsection, we present a branch-and-cut based decomposition algorithm. The algorithm is described in Algorithm 1, and the optimality cut generation procedure $\text{OptCuts}(\hat{x}, \hat{z}, \hat{\theta}, \mathcal{D})$ is given in Algorithm 2. The feasibility cut separation procedure $\text{SepCuts}(\hat{x}, \hat{z}, \mathcal{C})$ used in Algorithm 1 is the same as that in [13], so we do not discuss the details of $\text{SepCuts}(\hat{x}, \hat{z}, \mathcal{C})$.

Theorem 7. *Algorithm 1 converges to an optimal solution of (5) after finitely many iterations.*

Proof. First, as shown in [13], the number of feasibility cuts (10) is finite, and (10) always cuts off solutions where $z_k = 0$ but $x \notin P_k$. In addition, there are finitely many optimality cuts (22), (23) and (28), because there are finitely many $\pi_k \in \Pi_k, \bar{\pi}_k \in \bar{\Pi}_k$ (and hence α_k) to consider for all $k \in K$. Also, optimality cuts (22), (23) and (28) do not cut off any $(x, z, \theta) \in Z$ with $\sum_{k \in K} z_k = m - p$.

Next, we show that by using (22), (23) and (28), the algorithm converges to an optimal solution (x^*, z^*) . If for a current solution (\hat{x}, \hat{z}) , $\hat{z} = z^*$, then (22), (23) and (28) reduce to Benders optimality cuts, which correctly capture the cost approximation function for (\hat{x}, \hat{z}) . Otherwise, (22), (23) and (28) provide a lower bound on θ^* . Hence, the convergence of the algorithm directly follows from the convergence result of Benders decomposition algorithm. Finally, since the algorithm uses a branch-and cut procedure to solve the master problem, it processes a finite number of nodes. Thus, it terminates finitely. \square \square

4 Application and computational experiments

In this section we test our proposed algorithm on a resource planning problem first with no recovery (2) and then with recovery (5). We implemented our approach with C using CPLEX 12.4. The subroutines $\text{SepCuts}(\hat{x}, \hat{z}, \mathcal{C})$ and $\text{OptCuts}(\hat{x}, \hat{z}, \hat{\theta}, \mathcal{D})$ were implemented using the CPLEX lazy constraint callback

Algorithm 1: Decomposition algorithm for two-stage CCMPRs

```
1  $t \leftarrow 0, K_0(0) \leftarrow \emptyset, K_1(0) \leftarrow \emptyset, \mathcal{C} \leftarrow \mathbb{R}^{n_1 \times m}, \mathcal{D} \leftarrow \mathbb{R}^{n_1 \times m \times 1}, \text{OPEN} \leftarrow \{0\}, Ub \leftarrow$   
    $+\infty, Lb \leftarrow -\infty;$   
2 while OPEN  $\neq \emptyset$  do  
3   Step 1 : Choose  $l \in \text{OPEN}$  and let OPEN  $\leftarrow \text{OPEN} \setminus \{l\}$  ;  
4   Step 2 : Process node  $l$ ;  
5   while CUTFOUND and RCUTFOUND  $\neq \text{TRUE}$  and  $Lb < Ub$ ; do  
6     solve (9) ;  
7     if (9) is infeasible then  
8       CUTFOUND  $\leftarrow \text{FALSE}$  ;  
9     else  
10      Let  $(\hat{x}, \hat{z}, \hat{\theta})$  be an optimal solution to (9);  
11       $Lb \leftarrow \text{RP}(K_0(l), K_1(l))$ ;  
12      if  $\hat{z} \in \{0, 1\}^m$  then  
13        CUTFOUND  $\leftarrow \text{SepCuts}(\hat{x}, \hat{z}, \mathcal{C})$  ;  
14        if CUTFOUND  $\neq \text{TRUE}$  then  
15          CUTFOUND  $\leftarrow \text{OptCuts}(\hat{x}, \hat{z}, \hat{\theta}, \mathcal{D})$  ;  
16        end  
17        if CUTFOUND  $\neq \text{TRUE}$ , then update  $Ub \leftarrow Lb$   
18      end  
19    end  
20  end  
21  Step 3 : Branch if necessary;  
22  if  $Lb < Ub$  then  
23    Choose  $k \in K$  such that  $\hat{z}_k \in (0, 1)$ ;  
24     $K_0(t+1) \leftarrow K_0(l) \cup \{k\}, K_1(t+1) \leftarrow K_1(l)$  ;  
25     $K_0(t+2) \leftarrow K_0(l), K_1(t+2) \leftarrow K_1(l) \cup \{k\}$  ;  
26     $t \leftarrow t+2$ ;  
27    OPEN  $\leftarrow \text{OPEN} \cup \{t+1, t+2\}$   
28  end  
29 end
```

Algorithm 2: Procedure: OptCuts($\hat{x}, \hat{z}, \hat{\theta}, \mathcal{D}$)

Input: $\hat{\sigma} = (\hat{x}, \hat{z}, \hat{\theta})$.
Output: If a valid optimality cut for (5) is found which is violated by $(\hat{x}, \hat{z}, \hat{\theta})$, then add this cut to \mathcal{D} , and returns TRUE. Otherwise returns FALSE.

```
1 for all  $k \in K$  with  $\hat{z}_k = 0$  do  
2   solve (12) to obtain  $f(\hat{x}, k)$  and a dual optimal solution  $\pi_k$ ;  
3 end  
4 for all  $k \in K$  with  $\hat{z}_k = 1$  do  
5   solve (13) to obtain  $\bar{f}(\hat{x}, k)$  and a dual optimal solution  $\bar{\pi}_k$  if there is recovery.  
   Otherwise,  $\bar{\pi}_k = 0$ . ;  
6 end  
7 if  $\hat{\theta} < \frac{1}{m} \sum_{k=1}^m f(\hat{x}, k)(1 - \hat{z}_k) + \frac{1}{m} \sum_{k=1}^m \bar{f}(\hat{x}, k)\hat{z}_k$  then  
8   CUTFOUND  $\leftarrow \text{TRUE}$ ;  
9   Generate an optimality cut: if the randomness appears only in the right  
   hand-side, then use (28). Otherwise, use (22) and (23) with  $\alpha_k = T_k^\top \pi_k$  for  $k$  with  
    $\hat{z}_k = 0$ , and  $\alpha_k = \bar{T}_k^\top \bar{\pi}_k$  for  $k$  with  $\hat{z}_k = 1$  ( $\alpha_k = 0$  if there is no recovery);  
10 else  
11   CUTFOUND  $\leftarrow \text{FALSE}$   
12 end
```

function. These subroutines are called whenever CPLEX finds an integer candidate solution to the master problem (9). All the tests were run on a Windows XP operating system with 2.30 GHz Intel QuadCore processor 2356 (2 cores) with 2GB RAM. Both cores were used for testing the deterministic equivalent formulation and only a single core was used for the decomposition algorithm. A time limit of one hour and a tree memory limit of 500 MB were enforced.

4.1 Two-Stage CCMP without Recovery

Here we test our approach on a resource planning problem adapted from [13]. It consists of a set of resources (e.g., server types), denoted by $i \in I := \{1, \dots, n_1\}$, which can be used to meet demands of a set of customer types, denoted by $j \in J := \{1, \dots, r\}$.

The problem can be stated as:

$$\begin{aligned} \min_{x \in \mathbb{R}_+^{n_1}, z \in \mathbb{B}^m} \quad & c^\top x + \frac{1}{m} \sum_{k=1}^m ((1 - z_k)f(x, k) + z_k \bar{f}(x, k)) \\ \text{s.t.} \quad & z_k = 0 \Rightarrow x \in P_k, k \in K \\ & \sum_{k=1}^m z_k \leq p, \end{aligned}$$

where

$$\begin{aligned} P_k = \{x \in \mathbb{R}_+^{n_1} : \exists y \in \mathbb{R}_+^{n_1 \times r}, \\ \sum_{j=1}^r y_{ij} \leq \rho_{ki} x_i, \forall i \in I, \sum_{i=1}^{n_1} \mu_{kij} y_{ij} \geq \xi_{kj}, \forall j \in J\}. \end{aligned}$$

Here the first stage vector x represents the number of servers to employ, and c is its cost. In this problem, ξ, ρ, μ are random vectors following a finite and discrete joint distribution represented by a set of m equally likely scenarios K , where $\rho_{ki} \in (0, 1]$ represents the utilization rate of server type $i \in I$, $\xi_{kj} \geq 0$ represents the demand of customer type $j \in J$, and $\mu_{kij} \geq 0$ represents the rate of serving customer type j with server type i under scenario $k \in K$. Furthermore, the second stage problem for $k \in K$ is stated as:

$$\begin{aligned} f(x, k) = \{ \min_{y \in \mathbb{R}_+^{n_1 \times r}} q_k^\top y : \sum_{j=1}^r y_{ij} \leq \rho_{ki} x_i, \forall i \in I, \\ \sum_{i=1}^{n_1} \mu_{kij} y_{ij} \geq \xi_{kj}, \forall j \in J\}, \end{aligned}$$

where y_{ij} is the second-stage decision variable representing the number of server type i allocated to customer j , and q_{kij} is the unit allocation cost under scenario k . In this section, we assume there is no recovery model, and so $\bar{f}(x, k) \equiv 0$. In Section 4.2, we consider the problem with a recovery model.

We generate the parameters c , ρ_k , and μ_k following the scheme in [13] for the same type of resource planning problems. To generate the random demands ξ_k , we first generate a base demand $\bar{\xi}_j$ which follow $N(200, 20)$ for all $j \in J$. Then we let ξ_{kj} follow $N(\bar{\xi}_j, 0.1 \times \bar{\xi}_j)$ for all $k \in K$. We let the second stage cost $q_{kij} = \rho_{ki}$, for all $k \in K$ and $j \in J$, which guarantees that the second stage costs associated with the highly reliable servers are more expensive.

In Table 1, we summarize our experiments on problems where only the demand (right-hand side) is uncertain. We compare our “Strong” decomposition algorithm which uses the optimality cuts (28), against two other approaches: the deterministic equivalent problem (DEP) (6) and the “Basic” decomposition approach which uses the strong feasibility cuts (10) and the big-M optimality cuts (14). To obtain a valid big-M in inequality (14), we let $M_k = \sum_{j=1}^r \pi_{kj} \xi_{kj}$, where π_k is the dual vector for the subproblem (12) for $k \in K$ at the current optimality cut generation iteration.

In all the tables, each row reports the average of three instances under the same settings: the number of server and customer types (n_1, r) , the risk level ϵ , and the number of scenarios m . The Time column reports the average solution time in seconds, and the Gap column reports the average percentage end gap of all instances for each setting, given by $(Ub - Lb)/Ub$, where Ub and Lb are the best upper and lower bounds returned by the algorithm, respectively. The # column reports how many of the three instances are solved to optimality within the time limit. We do not include Time and # columns in a table, if an algorithm reached the time limit for all instances tested. The asterisk (*) indicates that there is at least one instance where the tree memory limit is reached. The dash (-) indicates that no instance is solved to optimality, and that no feasible solution is found by the algorithm within the time limit.

Table 1: Result for instances with random demand.

Instances		DEP	Basic	Strong		
(m, ϵ)	(n_1, r)	Gap (%)	Gap (%)	#	Time	Gap (%)
(2000, 0.05)	(5,10)	4.60	2.34	3	166	0
	(10,20)	-	2.93	3	483	0
	(15,30)	-	2.69	3	1106	0
(2500, 0.05)	(5,10)	4.64	2.61	3	279	0
	(10,20)	-	3.08	3	711	0
	(15,30)	-	2.88	2	1819	0.09
(2000, 0.1)	(5,10)	7.1	5.46	3	723	0
	(10,20)	-	5.99	3	1069	0
	(15,30)	-	6.27	3	1032	0
(2500, 0.1)	(5,10)	7.63	5.32	3	641	0
	(10,20)	-	5.79	3	1198	0
	(15,30)	-	6.03	2	2112	0.02

From Table 1 we see that the deterministic equivalent formulation is not able to solve any instances to optimality within the time limit. Moreover, it even fails

to find a feasible solution for instances of larger sizes. 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 instances within the time limit. The end 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 end gap is less than 0.1%.

The only difference between the basic and strong decomposition algorithm is the type of optimality cuts used. To illustrate the benefit of the optimality cuts we propose, we report the number of nodes explored (Node) and the optimality cuts (Cut) added to the master problem in the basic and strong decomposition algorithm in Table 2 for the instances in Table 1. Observe that the strong decomposition algorithm requires significantly fewer nodes than the basic decomposition algorithm. The number of optimality cuts added for the strong decomposition algorithm is also generally smaller than that for the basic decomposition algorithm. Hence, from Tables 1 and 2, we conclude that the additional computational effort to generate (28) pays off.

Table 2: Number of optimality cuts and nodes for instances with random demand.

Instances		Basic		Strong	
(m, ε)	(n_1, r)	Node	Cut	Node	Cut
(2000, 0.05)	(5,10)	15257	624	67	17
	(10,20)	9266	474	3	58
	(15,30)	10166	203	104	9
(2500, 0.05)	(5,10)	9174	563	6	34
	(10,20)	9833	427	3	70
	(15,30)	8203	190	20	135
(2000, 0.1)	(5,10)	10097	581	853	181
	(10,20)	13139	263	76	160
	(15,30)	8967	196	61	100
(2500, 0.1)	(5,10)	8275	554	598	93
	(10,20)	11703	266	19	134
	(15,30)	5867	132	9	137

In Table 3, we report results for instances with random demands, second-stage costs, service and utilization rates. Since these instances are much more challenging, we consider smaller instances. We see that the deterministic equivalent formulation and the basic decomposition algorithm were not able to solve most of the instances. In addition, for the basic decomposition method, the memory used by the branch-and-cut tree exploded very fast as we added the big-M optimality cuts into the master problem, and so most of the instances terminated due to the tree memory limit. On the other hand, the strong decom-

position algorithm with proposed optimality cuts (23) gives the best results. It solves many more instances to optimality within the time limit. In our implementation we choose not to use inequalities (22) for the case of no recovery.

Table 3: Results for instances with random ρ, μ, ξ, q .

Instances		DEP			Basic			Strong		
(m, ε)	(n_1, r)	#	Time	Gap	#	Time	Gap	#	Time	Gap
(400, 0.05)	(5,10)	1	2625	0.62	0	1765*	1.55	3	809	0
	(7,14)	0	3600	1.21	1	1020*	0.50	3	1915	0
(500, 0.05)	(5,10)	0	3600	1.08	0	3600	1.63	3	1379	0
	(7,14)	0	3600	1.37	0	3186*	0.36	2	2006	0.006
(600, 0.05)	(5,10)	0	3600	1.17	0	3067*	1.04	3	1346	0
	(7,14)	0	3600	2.81	0	1352*	0.50	2	2731	0.07
(400, 0.1)	(5,10)	0	3600	1.50	0	3600	3.49	1	2633	0.21
	(7,14)	0	3600	4.09	1	860*	0.68	0	1689	0.68
(500, 0.1)	(5,10)	0	3600	4.52	0	3339*	4.15	1	2528*	0.35
	(7,14)	0	3600	5.73	0	2187*	1.11	1	1831	0.74
(600, 0.1)	(5,10)	0	3600	3.89	0	2639*	1.82	1	2041	0.48
	(7,14)	0	3600	6.32	2	1449	1.00	2	2061	0.56

Table 4 reports the number of nodes and optimality cuts for both of the decomposition algorithms for instances with random demands, second-stage costs, service and utilization rates. As before, the strong decomposition algorithm requires much fewer optimality cuts and generally fewer number of branch-and-cut nodes to find solutions that are of higher quality. Note that the branch-and-cut nodes reported appear to be smaller in some cases for the basic decomposition algorithm than the strong decomposition algorithm, but this is because the former algorithm terminates prematurely for most of the instances due to the memory limit. Tables 3 and 4 highlight the benefits of obtaining strong optimality cuts (23) despite their high computational requirements to solve $O(mp)$ single scenario subproblems. Note also that we can further take advantage of the special structure of the resource planning problem as suggested in [13] to solve these problems more effectively.

4.2 Two-Stage CCMP with Recovery

Here we introduce the recovery version of the probabilistic resource planning problem studied in §4.1, where the simple recovery operation is given by

$$\bar{f}(x, k) = \left\{ \begin{array}{l} \min_{y \in \mathbb{R}_+^{n_1 \times r}, u \in \mathbb{R}_+^r} q_k^\top y + w_k^\top u : \sum_{j=1}^r y_{ij} \leq \rho_{ki} x_i, \forall i \in I, \\ \sum_{i=1}^{n_1} \mu_{kij} y_{ij} + u_j \geq \xi_{kj}, \forall j \in J, \end{array} \right.$$

Table 4: Number of optimality cuts and nodes for instances with random ρ, μ, ξ, q .

Instances		Basic		Strong	
(m, ε)	(n_1, r)	Node	Cut	Node	Cut
(400, 0.05)	(5,10)	36117	825	2676	207
	(7,14)	20674	367	14619	301
(500, 0.05)	(5,10)	49404	1361	6681	243
	(7,14)	92401	442	41060	177
(600, 0.05)	(5,10)	33291	996	17977	246
	(7,14)	33280	321	116324	162
(400, 0.1)	(5,10)	44402	880	83383	300
	(7,14)	41287	160	27159	169
(500, 0.1)	(5,10)	31864	878	82179	216
	(7,14)	23154	346	12052	124
(600, 0.1)	(5,10)	41218	394	37493	146
	(7,14)	23743	297	152293	82

where u_j is a variable that represents the level of outsourcing to cover the shortage in servers due to high demand of customer type $j \in J$. We let w_k , the unit cost of outsourcing, follow $N(3, 0.2)$, which is higher than the unit costs of service q_k .

In Table 5, we report the results for instances with random demands. For this class of problems we use two types of optimality cuts (22) and (28) in the strong decomposition algorithm, and compare its performance against the deterministic equivalent formulation and the basic decomposition method with the big-M optimality cut. In column ‘Strengthened’, we collected the results for the decomposition algorithm using the optimality cuts (22) only. In column ‘Strong’, we report the results of the decomposition algorithm which uses the optimality cut (28) only. The ‘-’ in the gap column indicates that there is at least one instance where CPLEX failed to find any feasible solution within the time limit. In this case, we report the average end gap only for the instances for which it is available. As we can see, neither the deterministic equivalent nor the basic decomposition algorithm terminated with an acceptable gap. In addition, the basic decomposition algorithm performs worse than DEP for this class of problems. In contrast, the strengthened decomposition algorithm which uses (22) as optimality cuts results in much smaller end gaps, and solves a few instances to optimality. The strong decomposition algorithm which uses (28) does not perform as well on the problems with recovery as it does for the problems without recovery. It reaches the time limit for all instances, but it still provides the smallest average end gaps for every setting.

For the two-stage CCMPR with random ρ, μ, ξ, q , we generated the “expensive” optimality cut (23) once every $m \times 0.02$ calls to the $\text{OptCuts}(\hat{x}, \hat{z}, \hat{\theta}, \mathcal{D})$ function. For the remaining calls, we use the strengthened big-M optimality cut (22). For example, for an instance which has 1000 scenarios, inequality (23) was

Table 5: Result for 2-stage CCMPR with random demands only.

Instances		DEP	Basic	Strengthened			Strong
(m, ε)	(n_1, r)	Gap (%)	Gap (%)	#	Time	Gap (%)	Gap (%)
(800, 0.05)	(5,10)	2.16	4.26	1	2484	1.27	0.19
	(10,20)	5.28	10.29	0	3600	2.44	0.19
(1200, 0.05)	(5,10)	5.05	9.07	1	2432	1.32	0.21
	(10,20)	4.05(-)	8.46	0	3600	2.51	0.24
(800, 0.1)	(5,10)	5.33	10.61	1	2423	0.73	0.26
	(10,20)	5.28	10.79	0	3600	1.45	0.39
(1200, 0.1)	(5,10)	7.32	9.08	1	3566	0.85	0.85
	(10,20)	5.02(-)	14.28	0	3600	1.69	0.60

generated once every 20 calls to the procedure $\text{OptCuts}(\hat{x}, \hat{z}, \hat{\theta}, \mathcal{D})$. We compare the proposed algorithm against the deterministic equivalent formulation and the basic decomposition algorithm.

Table 6: Result for two-stage CCMPR with random ρ, μ, ξ, q .

Instances		DEP	Basic	Strong		
(m, ε)	(n_1, r)	Gap (%)	Gap (%)	#	Time	Gap (%)
(1000, 0.05)	(5,10)	6.94	6.11	0	3600	1.34
	(10,20)	6.91	8.51	2	1381	0.14
(1200, 0.05)	(5,10)	7.61	6.53	0	3600	1.67
	(10,20)	-	8.79	2	1413	0.20
(1000, 0.1)	(5,10)	10.16	11.05	0	3600	2.62
	(10,20)	7.17	14.33	1	1539*	0.60
(1200, 0.1)	(5,10)	12.63	11.76	0	3600	2.95
	(10,20)	-	14.28	0	2316*	0.80

As we can see from Table 6, for the general two-stage CCMPR problems, the deterministic equivalent formulation and the basic decomposition algorithm which utilizes the big-M optimality cuts both performed poorly on all instances due to large solution times and end gaps. These instances are difficult for the strong decomposition algorithm as well. In most instances our algorithm reaches the time limit, but the end gaps are less than 3% for all instances. In addition, there are some instances where this algorithm reaches the memory limit. Therefore, developing more efficient algorithms for the general two-stage CCMPR continues to be an interesting research question.

5 Conclusion

In this paper, we study a class of chance-constrained two-stage stochastic optimization problems where second-stage feasible recourse decisions incur addi-

tional cost. In addition, “recovery” decisions are made for the infeasible scenarios to obtain feasible solutions to a relaxed second-stage problem. We develop Benders-type decomposition algorithms with specialized optimality and feasibility cuts to solve this class of problems. Computational results on a chance-constrained resource planning problem indicate that our algorithms are highly effective in solving these problems compared to a mixed-integer programming reformulation and a basic decomposition method, especially for the cases where the randomness appears only on the right-hand-side. Even though our description assumes that the first-stage feasible region X is a polyhedron, our algorithm can be extended to the case where there are integer restrictions on the first stage variables. Similarly, we currently add optimality cuts when an integral z is found as an optimal solution to the master problem. An interesting extension is to add the optimality cuts at fractional z encountered during the branch-and-bound algorithm.

Acknowledgements

We are grateful to the two anonymous referees and Dave Morton for their comments on an earlier version.

References

- [1] Benders, J.: Partitioning procedures for solving mixed-variables programming problems **4**(1), 238–252 (1962)
- [2] Beraldi, P., Ruszczyński, A.: A branch and bound method for stochastic integer programs under probabilistic constraints. *Optim. Methods Softw.* **17**, 359–382 (2002)
- [3] Beraldi, P., Ruszczyński, A.: The probabilistic set covering problem. *Oper. Res.* **50**(6), 956–967 (2002)
- [4] Bertsimas, D., Brown, D.: Constructing uncertainty sets for robust linear optimization. *Oper. Res.* **57**(6), 1483–1495 (2009)
- [5] Birge, J.R., Louveaux, F.V.: A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research* **34**(3), 384–392 (1988)
- [6] Calafiore, G., Campi, M.: Uncertain convex programs: randomized solutions and confidence levels. *Math. Program.* **102**, 25–46 (2005)
- [7] Calafiore, G., Campi, M.: The scenario approach to robust control design. *IEEE Trans. Automat. Control* **51**, 742–753 (2006)
- [8] Campi, M., Garatti, S.: A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. *J. Optim. Theory Appl.* **148**, 257–280 (2011)

- [9] Charnes, A., Cooper, W.: Deterministic equivalents for optimizing and satisficing under chance constraints. *Oper. Res.* **11**, 18–39 (1963)
- [10] Charnes, A., Cooper, W., Symonds, G.: Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Manage. Sci.* **4**, 235–263 (1958)
- [11] Dentcheva, D., Prékopa, A., Ruszczyński, A.: Concavity and efficient points of discrete distributions in probabilistic programming. *Math. Program.* **89**, 55–77 (2000)
- [12] Küçükyavuz, S.: On mixing sets arising in chance-constrained programming. *Math. Program.* **132**, 31–56 (2012)
- [13] Luedtke, J.: A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Math. Program.* **146**, 219–244 (2014)
- [14] Luedtke, J., Ahmed, S.: A sample approximation approach for optimization with probabilistic constraints. *SIAM J. Optim.* **19**, 674–699 (2008)
- [15] Luedtke, J., Ahmed, S., Nemhauser, G.: An integer programming approach for linear programs with probabilistic constraints. *Math. Program.* **12**, 247–272 (2010)
- [16] Miller, B.L., Wagner, H.M.: Chance constrained programming with joint constraints. *Oper. Res.* **13**(6), 930–965 (1965)
- [17] Miller, N., Ruszczyński, A.: Risk-averse two-stage stochastic linear programming: Modeling and decomposition. *Operations Research* **59**(1), 125–132 (2011)
- [18] Nemirovski, A., Shapiro, A.: Scenario approximation of chance constraints. In: Calafiore, G., Dabbene, F. (eds.) *Probabilistic and Randomized Methods for Design Under Uncertainty*, pp. 3–48. Springer, London (2005)
- [19] Noyan, N.: Risk-averse two-stage stochastic programming with an application to disaster management. *Computers and Operations Research* **39**(3), 541–559 (2012)
- [20] Prékopa, A.: Contributions to the theory of stochastic programming. *Math. Program.* **4**, 202–221 (1973)
- [21] Prékopa, A.: Dual method for the solution of a one-stage stochastic programming problem with random RHS obeying a discrete probability distribution. *ZOR - Methods and Models of Operations Research* **34**, 441–461 (1990)
- [22] Ruszczyński, A.: Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. *Math. Program.* **93**, 195–215 (2002)

- [23] Saxena, A., Goyal, V., Lejeune, M.: MIP reformulations of the probabilistic set covering problem. *Math. Program.* **121**, 1–31 (2009)
- [24] Sen, S.: Relaxations for probabilistically constrained programs with discrete random variables. *Oper. Res. Lett.* **11**, 81–86 (1992)
- [25] Song, Y., Luedtke, J., Küçükyavuz, S.: Chance-constrained binary packing problems. To appear in *INFORMS Journal on Computing* (2013)
- [26] Song, Y., Luedtke, J.R.: Branch-and-cut approaches for chance-constrained formulations of reliable network design problems. *Math. Program. Comput.* **5**(4), 397–432 (2013)
- [27] Takriti, S., Ahmed, S.: On robust optimization of two-stage systems. *Math. Program.* **99**, 109–126 (2004)
- [28] Van Slyke, R., Wets, R.J.: L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM J. Appl. Math.* **17**, 638–663 (1969)
- [29] Wang, J., Shen, S.: 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)
- [30] Wang, S., Guan, Y., Wang, J.: A chance-constrained two-stage stochastic program for unit commitment with uncertain wind power output. *IEEE Transactions on Power Systems* **27**, 206–215 (2012)
- [31] Zeng, B., An, Y., Kuznia, L.: Chance constrained mixed integer program: Bilinear and linear formulations, and Benders decomposition. *Optimization Online* (2014). http://www.optimization-online.org/DB_FILE/2014/03/4295.pdf
- [32] Zhang, M., Küçükyavuz, S., Goel, S.: A branch-and-cut method for dynamic decision making under joint chance constraints. *Manage. Sci.* **60**(5), 1317–1333 (2014)