



A loose Benders decomposition algorithm for approximating two-stage mixed-integer recourse models

Niels van der Laan¹ · Ward Romeijnders¹

Received: 3 December 2018 / Accepted: 28 August 2020 / Published online: 7 September 2020
© The Author(s) 2020

Abstract

We propose a new class of convex approximations for two-stage mixed-integer recourse models, the so-called generalized alpha-approximations. The advantage of these convex approximations over existing ones is that they are more suitable for efficient computations. Indeed, we construct a loose Benders decomposition algorithm that solves large problem instances in reasonable time. To guarantee the performance of the resulting solution, we derive corresponding error bounds that depend on the total variations of the probability density functions of the random variables in the model. The error bounds converge to zero if these total variations converge to zero. We empirically assess our solution method on several test instances, including the SIZES and SSLP instances from SIPLIB. We show that our method finds near-optimal solutions if the variability of the random parameters in the model is large. Moreover, our method outperforms existing methods in terms of computation time, especially for large problem instances.

Keywords Stochastic programming · Mixed-integer recourse · Convex approximations · Error bounds

Mathematics Subject Classification 90C15 · 90C11

1 Introduction

Consider the two-stage mixed-integer recourse model with random right-hand side

$$\eta^* := \min_x \{cx + Q(x) : Ax = b, x \in X \subseteq \mathbb{R}_+^{n_1}\}, \quad (1)$$

✉ Niels van der Laan
n.van.der.laan@rug.nl

¹ Department of Operations, University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands

where the recourse function Q is defined as

$$Q(x) := \mathbb{E}_\omega \left[\min_y \{qy : Wy = \omega - Tx, y \in Y \subseteq \mathbb{R}_+^{n_2}\} \right], \quad x \in X. \quad (2)$$

This model represents a two-stage decision problem under uncertainty. In the first stage, a decision x has to be made here-and-now, subject to deterministic constraints $Ax = b$ and random goal constraints $Tx = \omega$. Here, ω is a continuous or discrete random vector whose probability distribution is known. In the second stage, the realization of ω becomes known and any infeasibilities with respect to $Tx = \omega$ have to be repaired. This is modelled by the second-stage problem

$$v(\omega, x) := \min_y \{qy : Wy = \omega - Tx, y \in Y \subseteq \mathbb{R}_+^{n_2}\}. \quad (3)$$

The objective in this two-stage recourse model is to minimize the sum of immediate costs cx and expected second-stage costs $Q(x) = \mathbb{E}_\omega[v(\omega, x)]$, $x \in X$.

Frequently, integrality restrictions are imposed on the first- and second-stage decisions. That is, X and Y are of the form $X = \mathbb{Z}_+^{p_1} \times \mathbb{R}_+^{n_1-p_1}$ and $Y = \mathbb{Z}_+^{p_2} \times \mathbb{R}_+^{n_2-p_2}$. Such restrictions arise naturally when modelling real-life problems, for example to model on/off decisions or batch size restrictions. The resulting model is called a mixed-integer recourse (MIR) model. Such models have many practical applications in for example energy, telecommunication, production planning, and environmental control, see e.g. [10,37].

While MIR models are highly relevant in practice, they are notoriously difficult to solve. The reason is that Q is in general non-convex if integrality restrictions are imposed on the second-stage decision variables y , see [20]. Therefore, standard techniques for convex optimization cannot be used to solve these models. In contrast, if $Y = \mathbb{R}_+^{n_2}$, then Q is convex and efficient solution methods are available, most notably the L-shaped method in [32] and variants thereof.

Because of the non-convexity of Q , traditional solution methods for MIR models typically combine ideas from deterministic mixed-integer programming and stochastic continuous programming, see e.g. [4,8,9,15,17,19,29,31,38], and the survey papers [24,28,30]. We, however, use a fundamentally different approach to deal with the non-convex recourse function Q . Instead of solving the original MIR model in (1), we solve an approximating problem in which Q is replaced by a *convex approximation* \hat{Q} of Q . This convex approximation \hat{Q} does not have to be a lower bound of Q . The resulting approximating optimization problem is given by

$$\hat{\eta} := \min_x \{cx + \hat{Q}(x) : Ax = b, x \in X\}. \quad (4)$$

Because \hat{Q} is convex, we can use efficient techniques from convex optimization to solve the optimization problem in (4). Indeed, this optimization problem is convex if all first-stage variables $x \in X$ are continuous, and it is a MIP with a convex objective if some of these first-stage variables are integer. Thus, compared to traditional solution methods, we are able to solve similar-sized problems much faster, and solve

larger problem instances. In fact, we demonstrate this in our numerical experiments on problem instances available from [29] and SIPLIB [2], and on randomly generated problem instances.

Obviously, the optimal solution \hat{x} of the approximating problem in (4) is not necessarily optimal for the original MIR model in (1). That is why we guarantee the quality of the approximating solution \hat{x} , by deriving an error bound on

$$\|Q - \hat{Q}\|_{\infty} := \sup_x |Q(x) - \hat{Q}(x)|.$$

This error bound directly gives us an upper bound on the optimality gap of \hat{x} :

$$c\hat{x} + Q(\hat{x}) - \eta^* \leq 2\|Q - \hat{Q}\|_{\infty},$$

see [25].

Convex approximations and corresponding error bounds have been derived for many different classes of models. The idea to use convex approximations \hat{Q} for the non-convex mixed-integer recourse function Q dates back to [33], in which van der Vlerk proposes to use α -approximations for the special case of simple integer recourse (SIR) models. These α -approximations are obtained by perturbing the probability distribution of the random vector ω . Klein Haneveld et al. [14] derive an error bound for the α -approximations that depends on the total variations of the marginal density functions of the random variables in the SIR model.

More convex approximations have been described for more general classes of problems. For example, in [34,35] van der Vlerk extends the α -approximations to a class of MIR models with a single recourse constraint, and to integer recourse models with a totally unimodular (TU) recourse matrix, respectively. Furthermore, Romeijnnders et al. [26] derive an error bound for the latter approximation, and they derive a tighter error bound for the *shifted LP-relaxation* approximation for the same class of problems. The quality of the convex approximations for TU integer recourse models is assessed empirically in [22], and it turns out that they perform well if the variability of the random parameters in the model is large enough.

Romeijnnders et al. [23] generalize the shifted LP-relaxation to general two-stage MIR models, and they derive a corresponding *asymptotic error bound*. This error bound converges to zero if the variability of the random parameters in the model increases. Romeijnnders and van der Laan [21] derive similar error bounds for convex approximations that fit into a specific framework. In this framework, convex approximations are defined using *pseudo-valid cutting planes* for the second-stage feasible regions. Their idea is to only use cutting planes which are affine in the first-stage decision variables, so that the corresponding expected value function is convex. In general, the approximations are not exact, since pseudo-valid cutting planes cut away feasible solutions and are allowed to be overly conservative. Nevertheless, a similar error bound as for the shifted LP-relaxation has been derived if the pseudo-valid cutting planes are *tight*. That is, if they are exact on an entire grid of first-stage solutions.

Both the shifted LP-relaxation of [23] and the cutting plane framework of [21], however, cannot be applied directly to efficiently solve MIR models in general. This is

because the shifted LP-relaxation is very difficult to compute in general, as discussed in [23], and furthermore, the asymptotic error bound in the cutting plane framework of [21] only applies for tight pseudo-valid cutting planes, which are only available in special cases, e.g. for SIR models. That is why we propose an alternative class of convex approximations for general two-stage MIR models, the so-called generalized α -approximations. They are derived by exploiting properties of Gomory relaxations [11] of the second-stage mixed-integer programming problems.

Contrary to the shifted LP-relaxation, the generalized α -approximations, denoted by \hat{Q}_α , can be solved efficiently. In fact, we develop a so-called loose Benders decomposition algorithm to solve the approximating model in (4) with $\hat{Q} = \hat{Q}_\alpha$. Our Benders decomposition is called loose, because we derive optimality cuts for \hat{Q}_α which are in general not tight at the current solution. While these loose optimality cuts are in general not sufficient to find the optimal solution \hat{x} of the approximating model in (4), we prove that they are tight enough in the sense that a similar performance guarantee applies to the solution obtained by the loose Benders decomposition as to \hat{x} .

Summarizing, our main contributions are as follows.

- We propose a new class of convex approximations for general two-stage MIR models, which are based on Gomory relaxations of the second-stage problems. These generalized α -approximations can be solved efficiently, and a similar error bound as for the shifted LP-relaxation applies to the generalized α -approximations.
- We derive a loose Benders decomposition algorithm to (approximately) solve the approximating model with the generalized α -approximations. This is the first efficient algorithm for solving non-trivial convex approximations of general two-stage MIR models.
- We prove that the solution obtained by the loose Benders decomposition algorithm has a similar performance guarantee as the exact solution to the generalized α -approximations.
- We carry out extensive numerical experiments on 41 test instances from the literature and on 240 randomly generated instances, and show that using our loose Benders decomposition algorithm we obtain good solutions within reasonable time, also for large problem instances, in particular when the variability of the random parameters in the model is large.

The remainder of this paper is organized as follows. In Sect. 2, we discuss the shifted LP-relaxation of [23] and its corresponding error bound. In Sect. 3, we present the generalized α -approximations and an efficient algorithm for solving the corresponding approximating problem. Section 4 contains the proof of the performance guarantee for the loose Benders decomposition algorithm. In Sect. 5, we report on numerical experiments to evaluate the performance of our algorithm. Finally, we conclude in Sect. 6.

Throughout, we make the following assumptions. Assumptions (A2)–(A4) guarantee that $Q(x)$ is finite for all $x \in X$ such that $Ax = b$.

- (A1) The first-stage feasible region $\mathcal{X} := \{x \in X : Ax = b\}$ is bounded.
- (A2) The recourse is relatively complete: for all $\omega \in \mathbb{R}^m$ and $x \in \mathcal{X}$, there exists a $y \in Y$ such that $Wy = \omega - Tx$, so that $v(\omega, x) < \infty$.

- (A3) The recourse is sufficiently expensive: $v(\omega, x) > -\infty$ for all $\omega \in \mathbb{R}^m$ and $x \in \mathcal{X}$.
- (A4) $\mathbb{E}[|\omega_i|]$ is finite for all $i = 1, \dots, m$.
- (A5) The recourse matrix W is integer.

2 Existing convex approximations of mixed-integer recourse functions

In this section, we review the shifted LP-relaxation approximation of [23] and its corresponding error bound. First, however, we review results on asymptotic periodicity in mixed-integer linear programming in Sect. 2.1. We do so since these results are not only used to derive the shifted LP-relaxation in Sect. 2.2, but also to derive the generalized α -approximations in Sect. 3.1.

2.1 Asymptotic periodicity in mixed-integer programming

In order to derive convex approximations of the MIR function Q we analyze the value function v of the second-stage problem, defined as

$$v(\omega, x) = \min_y \left\{ qy : Wy = \omega - Tx, y \in \mathbb{Z}_+^{p_2} \times \mathbb{R}_+^{n_2-p_2} \right\}. \quad (5)$$

In particular, LP-duality implies that the LP-relaxation $v_{\text{LP}}(\omega, x)$ of $v(\omega, x)$ is polyhedral in the right-hand side vector $\omega - Tx$:

$$v_{\text{LP}}(\omega, x) = \max_{k=1, \dots, K} \lambda^k (\omega - Tx),$$

where λ^k , $k = 1 \dots, K$, are the extreme points of the dual feasible region $\{\lambda : \lambda W \leq q\}$. Romeijnnders et al. [23] derive a similar characterization of $v(\omega, x)$ in terms of linear and periodic functions by exploiting so-called Gomory relaxations. We briefly discuss these Gomory relaxations, before stating the characterization of $v(\omega, x)$ in Lemma 1.

The Gomory relaxation of $v(\omega, x)$ is defined for any dual feasible basis matrix of $v_{\text{LP}}(\omega, x)$. Let B denote such a matrix and let N be such that $W \equiv (B \ N)$, meaning equality up to a permutation of the columns. Let y_B and y_N denote the second-stage variables corresponding to the columns in B and N , respectively, and q_B and q_N their corresponding cost parameters. The Gomory relaxation v_B is obtained by *relaxing the non-negativity constraints* of the basic variables y_B . Romeijnnders et al. [23] derive the following expression for v_B :

$$v_B(\omega, x) = q_B B^{-1}(\omega - Tx) + \psi_B(\omega - Tx),$$

where

$$\psi_B(s) := \min \left\{ \bar{q}_N y_N : \begin{array}{l} B^{-1}(s - N y_N) \in \mathbb{Z}^{p_B} \times \mathbb{R}^{n_B}, \\ y_N \in \mathbb{Z}_+^{p_N} \times \mathbb{R}_+^{n_N} \end{array} \right\}, \quad s \in \mathbb{R}^m, \quad (6)$$

and $\bar{q}_N = q_N - q_B(B^{-1})N$. Moreover, they show that

$$v(\omega, x) = v_B(\omega, x) = q_B B^{-1}(\omega - Tx) + \psi_B(\omega - Tx)$$

if $\omega - Tx \in \Lambda := \{t : B^{-1}t \geq 0\}$, and if the distance of $\omega - Tx$ to the boundary of Λ is sufficiently large, see Definition 1. We say that $v(\omega, x)$ is asymptotically periodic, since ψ_B is a B -periodic function, i.e. $\psi_B(s + Bl) = \psi_B(s)$ for every $s \in \mathbb{R}^m$ and $l \in \mathbb{Z}^m$, see [23].

Definition 1 Let $\Lambda \subset \mathbb{R}^m$ be a closed convex set and let $d \in \mathbb{R}$ with $d > 0$ be given. Then, we define $\Lambda(d)$ as

$$\Lambda(d) := \{s \in \Lambda : \mathcal{B}(s, d) \subset \Lambda\},$$

where $\mathcal{B}(s, d) := \{t \in \mathbb{R}^m : \|t - s\|_2 \leq d\}$ is the closed ball centered at s with radius d . We can interpret $\Lambda(d)$ as the set of points in Λ with at least Euclidean distance d to the boundary of Λ .

Lemma 1 [23, Theorem 2.9] *Consider the mixed-integer programming problem*

$$v(\omega, x) := \min_y \left\{ qy : Wy = \omega - Tx, y \in \mathbb{Z}_+^{p_2} \times \mathbb{R}_+^{n_2 - p_2} \right\},$$

where W is an integer matrix, and $v(\omega, x)$ is finite for all $\omega \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$. Then, there exist dual feasible basis matrices B^k of v_{LP} , $k = 1, \dots, K$, closed convex polyhedral cones $\Lambda^k := \{t \in \mathbb{R}^m : (B^k)^{-1}t \geq 0\}$, distances d_k , B^k -periodic functions ψ^k , and constants w_k such that we have the following:

- (i) $\bigcup_{k=1}^K \Lambda^k = \mathbb{R}^m$.
- (ii) $(\text{int } \Lambda^k) \cap (\text{int } \Lambda^l) = \emptyset$ for every $k, l \in \{1, \dots, K\}$ with $k \neq l$.

(iii) If $\omega - Tx \in \Lambda^k(d_k)$, then

$$v(\omega, x) = v_{LP}(\omega, x) + \psi^k(\omega - Tx) = q_{B^k}(B^k)^{-1}(\omega - Tx) + \psi^k(\omega - Tx),$$

where $\psi^l \equiv \psi^k$ if $q_{B^k}(B^k)^{-1} = q_{B^l}(B^l)^{-1}$.

- (iv) $0 \leq \psi^k(\omega - Tx) \leq w_k$ for all $\omega \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$, $k = 1, \dots, K$.

Lemma 1 (iii) shows that if $\omega - Tx \in \Lambda^k(d_k)$ for some $k = 1, \dots, K$, then $v(\omega, x)$ is equal to the sum of the LP-relaxation $v_{LP}(\omega, x)$ and $\psi^k(\omega - Tx)$. Hence, $\psi^k(\omega - Tx)$ can be interpreted as the additional costs resulting from the integrality restrictions on the decision variables y . For a discussion on how to obtain d_k or how to represent $\Lambda^k(d_k)$ using a system of linear inequalities we refer to [23].

2.2 The shifted LP-relaxation approximation

Lemma 1 shows why the second-stage value function is not convex in x . On regions of its domain it is the sum of a linear function $q_{B^k}(B^k)^{-1}(\omega - Tx)$ and a periodic function $\psi^k(\omega - Tx)$. Clearly the periodic part is causing v to be non-convex. That is why the shifted LP-relaxation is obtained by replacing this periodic part $\psi^k(\omega - Tx)$ by a constant Γ_k for every $k = 1, \dots, K$, with Γ_k defined as

$$\Gamma_k := p_k^{-m} \int_0^{p_k} \dots \int_0^{p_k} \psi^k(x) dx_1 \dots dx_m, \quad (7)$$

where $p_k = |\det B^k|$. The K constants Γ_k can be interpreted as the averages of the periodic functions ψ^k . The shifted LP-relaxation approximation is obtained by taking the pointwise maximum over all dual feasible basis matrices B^k , $k = 1, \dots, K$.

Definition 2 Define the *shifted LP-relaxation approximation* \tilde{Q} of the MIR function Q as $\tilde{Q}(x) = \mathbb{E}_\omega[\tilde{v}(\omega, x)]$, where

$$\tilde{v}(\omega, x) := \max_{k=1, \dots, K} \{q_{B^k}(B^k)^{-1}(\omega - Tx) + \Gamma_k\},$$

where B^k , $k = 1, \dots, K$, are the dual feasible basis matrices of Lemma 1, and Γ_k is defined in (7).

Romeijnders et al. [23] derive a total variation error bound on the approximation error $\|Q - \tilde{Q}\|_\infty$ of the shifted LP-relaxation \tilde{Q} . This error bound is expressed in terms of the *total variations* of the one-dimensional conditional probability density functions (pdf) of the random vector ω .

Definition 3 Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a real-valued function, and let $I \subset \mathbb{R}$ be an interval. Let $\Pi(I)$ denote the set of all finite ordered sets $P = \{x_1, \dots, x_{N+1}\}$ with $x_1 < \dots < x_{N+1}$ in I . Then, the *total variation* of f on I , denoted by $|\Delta|f(I)$, is defined as

$$|\Delta|f(I) = \sup_{P \in \Pi(I)} V_f(P),$$

where

$$V_f(P) = \sum_{i=1}^N |f(x_{i+1}) - f(x_i)|.$$

We write $|\Delta|f := |\Delta|f(\mathbb{R})$.

Definition 4 For every $i = 1, \dots, m$ and $x_{-i} \in \mathbb{R}^{m-1}$, define the i -th conditional density function $f_i(\cdot|x_{-i})$ of the m -dimensional joint pdf f as

$$f_i(x_i|x_{-i}) = \frac{f(x)}{f_{-i}(x_{-i})},$$

where f_{-i} is the joint pdf of the $(m-1)$ -dimensional random vector ω_{-i} , which is equal to ω without its i -th component. Define \mathcal{H}^m as the set of all m -dimensional joint pdf f such that $f_i(\cdot|x_{-i})$ is of bounded variation for all $i = 1, \dots, m$ and $x_{-i} \in \mathbb{R}^{m-1}$.

Theorem 1 [23, Theorem 5.1] *There exists a constant $C > 0$ such that for every continuous random vector ω with joint pdf $f \in \mathcal{H}^m$,*

$$\sup_x |Q(x) - \tilde{Q}(x)| \leq C \sum_{i=1}^m \mathbb{E}_{\omega_{-i}} [|\Delta| f_i(\cdot|\omega_{-i})]. \quad (8)$$

In general, the error bound in Theorem 1 may be large, in particular since the constant $C > 0$ may be large. Nevertheless, the theorem shows that if the total variations of the one-dimensional conditional pdf are small, then \tilde{Q} is a good approximation of Q . For example, if the components ω_i of ω follow independent normal distributions, with mean μ_i and variance σ_i^2 , for $i = 1, \dots, m$, then the error bound in (8) simplifies to $C' \sum_{i=1}^m \sigma_i^{-1}$ for some $C' > 0$; see [21, Example 2] for details. Observe that the error bound goes to zero if $\sigma_i \rightarrow \infty$ for all $i = 1, \dots, m$. Thus, the error of using the shifted LP-relaxation approximation decreases if the variability of the random parameters in the model increases.

3 Loose Benders decomposition algorithm for two-stage mixed-integer recourse models

3.1 Generalized α -approximations

To derive the generalized α -approximations \hat{Q}_α , we first derive a convex approximation \hat{v}_α of the second-stage value function v defined in (5), and we define $\hat{Q}_\alpha(x) := \mathbb{E}_\omega[\hat{v}_\alpha(\omega, x)]$. Similar as for the shifted LP-relaxation \tilde{v} , we use Lemma 1, i.e. for $\omega - Tx \in \Lambda^k(d_k)$,

$$v(\omega, x) = v_{B^k}(\omega, x) = q_{B^k}(B^k)^{-1}(\omega - Tx) + \psi^k(\omega - Tx).$$

However, instead of replacing $\psi^k(\omega - Tx)$ by its average Γ_k , as is done for the shifted LP-relaxation, we replace $\psi^k(\omega - Tx)$ by $\psi^k(\omega - \alpha)$ for some $\alpha \in \mathbb{R}^m$ to obtain the generalized α -approximation \hat{v}_α^k of v_{B^k} defined as

$$\hat{v}_\alpha^k(\omega, x) = q_{B^k}(B^k)^{-1}(\omega - Tx) + \psi^k(\omega - \alpha),$$

and we define the generalized α -approximation \hat{v}_α of v as

$$\hat{v}_\alpha(\omega, x) = \max_k \hat{v}_\alpha^k(\omega, x).$$

The difference compared to the shifted LP-relaxation seems small: the constants Γ_k are replaced by $\psi^k(\omega - \alpha)$. From a computational point of view, however, this difference is significant. This is because the constants Γ_k are the *averages* over ψ^k , and in general need to be obtained by computing a multi-dimensional integral of a mixed-integer value function. For a fixed ω and α , however, the value of $\psi^k(\omega - \alpha)$ is obtained by solving a *single* mixed-integer programming problem of the same size as the second-stage problem. In fact, we need to solve the Gomory relaxation discussed in Sect. 2.1, which can be done in polynomial time if all second-stage variables are integer [11].

Definition 5 For $\alpha \in \mathbb{R}^m$, we define the *generalized α -approximation* \hat{Q}_α of Q as

$$\hat{Q}_\alpha(x) := \mathbb{E}_\omega \left[\max_{k=1, \dots, K} \{ \lambda^k(\omega - Tx) + \psi^k(\omega - \alpha) \} \right], \quad x \in \mathbb{R}^{n_1},$$

with $\lambda^k := q_{B^k}(B^k)^{-1}$ and $\psi^k := \psi_{B^k}$, where $B^k, k = 1, \dots, K$, are the dual feasible basis matrices of Lemma 1.

Remark 1 The generalized α -approximations are a generalization of the α -approximations defined for *TU integer recourse models* [26], which arise if $y \in \mathbb{Z}_+^p$ and if the second-stage constraints are of the form $Wy \geq \omega - Tx$, where W is a TU matrix. Indeed, for these models, we have $\psi^k(s) = \lambda^k(\lceil s \rceil - s)$, see [23], and thus

$$\hat{Q}_\alpha(x) = \mathbb{E}_\omega \left[\max_{k=1, \dots, K} \lambda^k(\lceil \omega - \alpha \rceil + \alpha - Tx) \right],$$

which is the expression for the α -approximations for TU integer recourse models in [26].

It turns out that we can derive a similar error bound for the generalized α -approximations as for the shifted LP-relaxation.

Theorem 2 *There exists a constant $C > 0$, such that for every $\alpha \in \mathbb{R}^m$ and for every continuous random vector ω with joint pdf $f \in \mathcal{H}^m$,*

$$\sup_x |Q(x) - \hat{Q}_\alpha(x)| \leq C \sum_{i=1}^m \mathbb{E}_{\omega_{-i}} [|\Delta| f_i(\cdot | \omega_{-i})].$$

Proof See “Appendix”. □

Theorem 2 states that the approximation error of \hat{Q}_α goes to zero as the total variations $\mathbb{E}_{\omega_{-i}} [|\Delta| f_i(\cdot | \omega_{-i})]$, $i = 1, \dots, m$, all go to zero. This error bound is

independent of α , i.e. \hat{Q}_α is a good approximation of the true MIR function Q for any $\alpha \in \mathbb{R}^m$.

An interesting difference between the generalized α -approximations and the shifted LP-relaxation of Sect. 2.2 is that the approximating value function \hat{v}_α is not convex in ω for fixed $x \in \mathbb{R}^{n_1}$. Indeed, \hat{v}_α is only convex in x for every fixed ω , but this is sufficient to guarantee that the generalized α -approximation \hat{Q}_α is convex. In contrast, the value function \tilde{v} of the shifted LP-relaxation is convex in both ω and x . We illustrate these properties in Example 1. In this example, we also illustrate the difference between the generalized α -approximations and the pseudo-valid cutting plane approximation of [21].

Example 1 Consider the second-stage mixed-integer value function v defined as

$$v(\omega, x) := \min\{y_1 + 2y_2 + 2y_3 : y_1 + y_2 - y_3 = \omega - x, y_1 \in \mathbb{Z}_+, y_2, y_3 \in \mathbb{R}_+\}.$$

We use this example, since it also appears in [23] and [21], allowing us to compare the generalized α -approximations to the shifted LP-relaxation and a pseudo-valid cutting plane approximation.

The LP-relaxation of v has two dual feasible basis matrices $B^1 = [-1]$ and $B^2 = [1]$. Thus, $K = 2$, and straightforward computations yield $\lambda^1 = -2$, $\lambda^2 = 1$, $\psi^1 \equiv 0$, and for every $s \in \mathbb{R}$,

$$\psi^2(s) = \begin{cases} s - \lfloor s \rfloor, & \text{if } s - \lfloor s \rfloor \leq 3/4, \\ 3 - 3(s - \lfloor s \rfloor), & \text{if } s - \lfloor s \rfloor \geq 3/4. \end{cases}$$

It follows that $v_{LP}(\omega, x) = \max\{-2(\omega - x), \omega - x\}$. Moreover, for every $\alpha \in \mathbb{R}$, the approximating value function is defined as

$$\hat{v}_\alpha(\omega, x) = \max\{-2(\omega - x), \omega - x + \psi^2(\omega - \alpha)\}.$$

In contrast, the value function \tilde{v} of the shifted LP-relaxation equals

$$\tilde{v}(\omega, x) = \max\{-2(\omega - x), \omega - x + 3/8\},$$

since $\Gamma^2 := \int_0^1 \psi^2(s) ds = 3/8$. Finally, the tight pseudo-valid cutting plane approximation \hat{v}_c is given by

$$\hat{v}_c(\omega, x) = \min_{y \in \mathbb{R}_+^3} \{y_1 + 2y_2 + 2y_3 : y_1 + y_2 - y_3 = \omega - x, \\ y_1 - \frac{1}{1 + \lfloor \omega \rfloor - \omega} y_3 \leq \lfloor \omega \rfloor - x\}.$$

Figure 1a, b show $v(\omega, x)$, $\hat{v}_\alpha(\omega, x)$, $\tilde{v}(\omega, x)$, and $\hat{v}_c(\omega, x)$ as a function of x and ω , respectively. They illustrate that \hat{v}_α is convex in x for fixed ω , but not convex in ω for fixed x , respectively. Moreover, Fig. 1b illustrates a key difference between $\tilde{v}(\omega, x)$ and $\hat{v}_c(\omega, x)$. Whereas $\tilde{v}(\omega, x)$ is obtained by replacing the periodic part $\psi^2(\omega - x)$

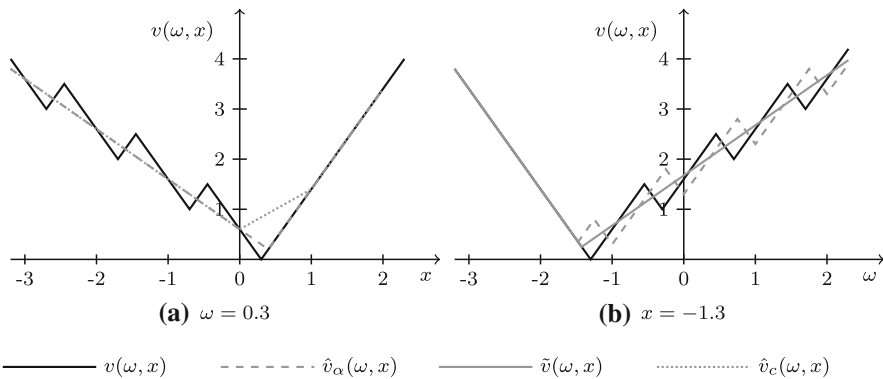


Fig. 1 The value function v and the approximating value functions \tilde{v} , \hat{v}_c , and \hat{v}_α , where $\alpha = 0$. The left figure shows $v(\omega, x)$, $\hat{v}_\alpha(\omega, x)$, and $\hat{v}_c(\omega, x)$, as a function of x , whereas the right figure shows $v(\omega, x)$, $\hat{v}_\alpha(\omega, x)$, and $\tilde{v}(\omega, x)$ as a function of ω

of $v(\omega, x)$ by its average Γ^2 , we obtain $\hat{v}_\alpha(\omega, x)$ by shifting $\psi^2(\omega - x)$ towards $\psi^2(\omega - \alpha)$.

Finally, Fig. 1a shows that $\hat{v}_\alpha(\omega, x) = \hat{v}_c(\omega, x)$ for a range of x values. However, the difference $\hat{v}_c(\omega, x) - \hat{v}_\alpha(\omega, x)$ is relatively large if $x \in (-0.3, 0.7)$. This is true in general for tight pseudo-valid cutting plane approximations: there always exist \hat{d}_k such that $\hat{v}_\alpha(\omega, x) = \hat{v}_c(\omega, x)$ if $\omega - Tx \in \Lambda^k(\hat{d}_k)$. However, the two approximations may differ if $\omega - Tx \notin \bigcup_k \Lambda^k(\hat{d}_k)$. \diamond

3.2 Benders decomposition for the generalized α -approximations

We solve the approximating problem

$$\hat{\eta}_\alpha := \min_x \{cx + \hat{Q}_\alpha(x) : Ax = b, x \in X\}, \quad (9)$$

using an SAA of \hat{Q}_α . Given a sample $\{\omega^1, \dots, \omega^S\}$ of size S from the distribution of ω , the SAA \hat{Q}_α^S of \hat{Q}_α is defined as

$$\hat{Q}_\alpha^S(x) := \frac{1}{S} \sum_{s=1}^S \hat{v}_\alpha(\omega^s, x), \quad x \in X, \quad (10)$$

and the corresponding SAA problem as

$$\hat{\eta}_\alpha^S := \min_x \{cx + \hat{Q}_\alpha^S(x) : Ax = b, x \in X\}. \quad (11)$$

Since \hat{Q}_α^S is convex, we can solve the SAA problem using Benders decomposition [6], i.e. using an L-shaped algorithm [32]. In iteration τ of this algorithm, we solve (11)

with \hat{Q}_α^S replaced by a convex polyhedral *outer approximation* $\hat{Q}_{\text{out}}^\tau \leq \hat{Q}_\alpha^S$. This problem is called the *master problem*

$$\hat{\eta}_\tau := \min_x \{cx + \hat{Q}_{\text{out}}^\tau(x) : Ax = b, x \in X\}, \quad (12)$$

and its optimal solution x^τ is referred to as the *current solution* at iteration τ . If $\hat{Q}_{\text{out}}^\tau(x^\tau) = \hat{Q}_\alpha^S(x^\tau)$, then the current solution x^τ is optimal for the original SAA problem in (11). If not, then we strengthen the outer approximation using an *optimality cut* for \hat{Q}_α^S of the form

$$\hat{Q}_\alpha^S(x) \geq \beta_{\tau+1}x + \delta_{\tau+1}, \quad \forall x \in X,$$

which is tight at x^τ , i.e. $\hat{Q}_\alpha^S(x^\tau) = \beta_{\tau+1}x^\tau + \delta_{\tau+1}$. The outer approximation $\hat{Q}_{\text{out}}^\tau$ in iteration τ is the pointwise maximum over all optimality cuts derived in previous iterations:

$$\hat{Q}_{\text{out}}^\tau(x) = \max_{t=1, \dots, \tau} \{\beta_t x + \delta_t\}, \quad x \in X.$$

The challenge for the generalized α -approximations, however, is to compute tight optimality cuts of \hat{Q}_α^S . Since

$$\hat{Q}_\alpha^S(x^\tau) = \frac{1}{S} \sum_{s=1}^S \max_{k=1, \dots, K} \{\lambda^k(\omega^s - Tx^\tau) + \psi^k(\omega^s - \alpha)\}, \quad x \in X,$$

such tight optimality cuts can be obtained by identifying for each scenario s the maximizing index k_s^τ at x^τ , defined as

$$k_s^\tau \in \arg \max_{k=1, \dots, K} \{\lambda^k(\omega^s - Tx^\tau) + \psi^k(\omega^s - \alpha)\}, \quad s = 1, \dots, S. \quad (13)$$

However, this is computationally too expensive, since we need to compute $\psi^k(\omega^s - \alpha)$ for all $k = 1, \dots, K$, and K grows *exponentially* in the size of the second-stage problem. Indeed, K is at least as large as the number of dual vertices of the feasible region of the LP-relaxation v_{LP} of v , of which there are exponentially many.

3.3 Loose Benders decomposition for the generalized α -approximations

To overcome this computational challenge we propose to use approximate indices \hat{k}_s^τ defined as

$$\hat{k}_s^\tau \in \arg \max_{k=1, \dots, K} \lambda^k(\omega^s - Tx^\tau), \quad s = 1, \dots, S. \quad (14)$$

These indices are *computationally tractable* since they correspond to the optimal basis matrix index of the LP-relaxation $v_{\text{LP}}(\omega^s, x^\tau)$, defined in (5). Hence, they can

be obtained by solving a single LP. Moreover, if the values of $\psi^k(\omega^s - \alpha)$ are equal for all $k = 1, \dots, K$, then the indices \hat{k}_s^τ are optimal in (13), i.e. $\hat{k}_s^\tau = k_s^\tau$. In general however, the indices k_s are suboptimal in (13), leading to the following definition.

Definition 6 Let x^τ be given and let \hat{k}_s^τ , $s = 1, \dots, S$, be as in (14). We define the *loose optimality cut* for \hat{Q}_α^S at x^τ as

$$\hat{Q}_\alpha^S(x) \geq \frac{1}{S} \sum_{s=1}^S \lambda^{\hat{k}_s^\tau} (\omega^s - Tx) + \psi^{\hat{k}_s^\tau} (\omega^s - \alpha), \quad x \in X.$$

We use these loose optimality cuts in our loose Benders decomposition algorithm, LBDA(α). In this algorithm the outer approximation $\hat{Q}_{\text{out}}^\tau$ of \hat{Q}_α^S is defined using our loose optimality cuts. The algorithm terminates with tolerance level $\varepsilon \geq 0$ if $\hat{Q}_{\text{out}}^\tau(x^\tau) \geq \hat{Q}_{\text{out}}^{\tau+1}(x^\tau) - \varepsilon$; then, LBDA(α) reports $\hat{x}_\alpha = x^\tau$ as solution.

A full description of LBDA(α) is given below. Note that the algorithm requires a lower bound L on \hat{Q}_α^S such that $\hat{Q}_\alpha^S(x) \geq L$ for all $x \in \mathcal{X}$. Such a lower bound L exists because of Assumptions (A1)–(A4).

Loose Benders decomposition algorithm (LBDA(α))

- 1: **Inputs** Parameters: A , b , c , T , q , W . Distribution of ω . Lower bound L on \hat{Q}_α^S . Shift parameter α . Tolerance ε . Sample size S .
 - 2: **Output** Near-optimal solution \hat{x}_α .
 - 3: **Initialization**
 - 4: Initialize $\tau = 0$ and $\hat{Q}_{\text{out}}^\tau \equiv L$.
 - 5: Obtain a sample $\{\omega^1, \dots, \omega^S\}$ of size S from the distribution of ω .
 - 6: **Iteration step**
 - 7: Solve $\min_x \{cx + \hat{Q}_{\text{out}}^\tau(x) : Ax = b, x \in X\}$.
 - 8: Denote the optimal solution by x^τ .
 - 9: **for** $s = 1, \dots, S$ **do**
 - 10: Solve $v_{\text{LP}}(\omega^s, x^\tau)$.
 - 11: Denote the optimal basis matrix index by \hat{k}_s^τ .
 - 12: Solve $v_{B^{\hat{k}_s^\tau}}(\omega^s - \alpha, 0)$.
 - 13: Compute $\psi^{\hat{k}_s^\tau}(\omega^s - \alpha) = v_{B^{\hat{k}_s^\tau}}(\omega^s - \alpha, 0) - \lambda^{\hat{k}_s^\tau}(\omega^s - \alpha)$.
 - 14: **end for**
 - 15: $\beta_{\tau+1} \leftarrow -\frac{1}{S} \sum_{s=1}^S \lambda^{\hat{k}_s^\tau} T$.
 - 16: $\delta_{\tau+1} \leftarrow \frac{1}{S} \sum_{s=1}^S \left\{ \lambda^{\hat{k}_s^\tau} \omega^s + \psi^{\hat{k}_s^\tau}(\omega^s - \alpha) \right\}$.
 - 17: $\hat{Q}_{\text{out}}^{\tau+1}(x) := \max\{\hat{Q}_{\text{out}}^\tau(x), \beta_{\tau+1}x + \delta_{\tau+1}\}$.
 - 18: **Stopping criterion**
 - 19: **if** $\hat{Q}_{\text{out}}^\tau(x^\tau) \geq \hat{Q}_{\text{out}}^{\tau+1}(x^\tau) - \varepsilon$ **then**
 - 20: **return** $\hat{x}_\alpha := x^\tau$.
 - 21: **stop.**
 - 22: **else**
 - 23: $\tau \leftarrow \tau + 1$. Go to line 7.
 - 24: **end if**
-

The solution \hat{x}_α of LBDA(α) is not necessarily ε -optimal for the SAA problem in (11), since LBDA(α) uses the loose optimality cuts of Definition 6. If the optimality cuts were tight, then $\hat{Q}_{\text{out}}^{\tau+1}(x^\tau) = \hat{Q}_\alpha^S(x^\tau)$ for every iteration τ , and thus the algorithm would terminate if $\hat{Q}_{\text{out}}^\tau(x^\tau) \geq \hat{Q}_\alpha^S(x^\tau) - \varepsilon$. We, however, show that our

loose optimality cuts are *asymptotically tight* at x^τ : the difference $\hat{Q}_{\text{out}}^{\tau+1}(x^\tau) - \hat{Q}_\alpha^S(x^\tau)$ converges to zero if the total variations of the one-dimensional conditional pdf of the random vector ω go to zero and as $S \rightarrow \infty$, see Proposition 1. In this case, we are able to prove that the LBDA(α) solution \hat{x}_α is near-optimal; see Theorem 3 below for a bound on the optimality gap $c\hat{x}_\alpha + Q(\hat{x}_\alpha) - \eta^*$. This performance guarantee is independent of α , and thus it applies if we take e.g. $\alpha = 0$. We further explore selection of α in Sect. 5. The proof of the performance guarantee is postponed to Sect. 4.2.

Theorem 3 *Consider the two-stage mixed-integer recourse model*

$$\eta^* = \min_x \{cx + Q(x) : Ax = b, x \in X\}.$$

Let \hat{x}_α denote the solution by LBDA(α) with tolerance ε and sample size S . Then, there exists a constant $C > 0$ such that for every continuous random vector ω with joint pdf $f \in \mathcal{H}^m$

$$c\hat{x}_\alpha + Q(\hat{x}_\alpha) - \eta^* \leq \varepsilon + C \sum_{i=1}^m \mathbb{E}_{\omega_{-i}} [|\Delta| f_i(\cdot | \omega_{-i})],$$

w.p. 1 as $S \rightarrow \infty$.

Theorem 3 implies that the optimality gap of \hat{x}_α converges to the prespecified tolerance ε as the total variations of the underlying one-dimensional conditional pdf go to zero.

3.4 Implementation details of LBDA(α)

LBDA(α) can be implemented efficiently if the input size of the second-stage problem $v(\omega, x)$ is moderate. During each iteration τ , we have to solve the LP-relaxation $v_{\text{LP}}(\omega^s, x^\tau)$ and the Gomory relaxation $v_{B^{\hat{k}_s^\tau}}(\omega^s - \alpha, 0)$ for each scenario $s = 1, \dots, S$, in order to generate a loose optimality cut. If the input size of $v(\omega, x)$ is not too large, then $v_{\text{LP}}(\omega^s, x^\tau)$ and $v_{B^{\hat{k}_s^\tau}}(\omega^s - \alpha, 0)$ can be solved in reasonable time using standard LP and MIP solvers, respectively. Moreover, the master problem can be solved efficiently using a standard LP solver, or MIP solver if some of the first-stage decision variables are integer. Improved implementations of LBDA(α) using a multicut approach [7] and regularization techniques [27] are possible. Furthermore, the subproblems $v_{B^{\hat{k}_s^\tau}}(\omega^s - \alpha, 0)$, $s = 1, \dots, S$ can be solved in parallel, and it may be beneficial to solve these subproblems inexactly in the first phase of the algorithm.

In Sect. 5, we exploit that LBDA(α) can be run multiple times, using different values of α . The performance guarantee of LBDA(α) is independent of α and thus applies to every candidate solution that we obtain in this way. Moreover, we use in- and out-of-sample evaluation to select the best candidate solution. In our numerical experiments, we investigate several schemes for selecting the values of α . Running LBDA(α) multiple times can be done very efficiently by using parallelization and a *common warm start*. That is, we first apply the L-shaped algorithm of [32] to the

LP-relaxation of the original problem (1), in which the integer restrictions on the second-stage variables y are relaxed. Next, we run LBDA(α) for each value of α , and we keep the optimality cuts generated for the LP-relaxation Q_{LP} of Q , which are also valid for \hat{Q}_α , for any value of α .

4 Performance guarantee of LBDA(α)

4.1 Convergence of sampling and loose optimality cuts

The performance guarantee of LBDA(α) does not follow directly from our error bound for the generalized α -approximations. The reason is that LBDA(α) uses sampling and the loose optimality cuts of Definition 6 to solve the corresponding approximating problem. We consider these aspects in Sects. 4.1.1 and 4.1.2, respectively. In particular, we prove consistency of the SAA and asymptotic tightness of our loose optimality cuts.

4.1.1 Consistency of the sample average approximation

Intuitively, the SAA becomes better as the sample size S increases. Indeed, we show in Lemma 2 that the SAA \hat{Q}_α^S of \hat{Q}_α converges uniformly to \hat{Q}_α on \mathcal{X} w.p. 1 as $S \rightarrow \infty$.

Lemma 2 *Consider the generalized α -approximation \hat{Q}_α and its sample average approximation \hat{Q}_α^S . Then,*

$$\sup_{x \in \mathcal{X}} |\hat{Q}_\alpha(x) - \hat{Q}_\alpha^S(x)| \rightarrow 0$$

w.p. 1 as $S \rightarrow \infty$, where $\mathcal{X} = \{x \in \mathbb{R}_+^{n_1} : Ax \leq b\}$.

Proof See “Appendix”. □

Corollary 1 $\sup_{x \in \mathcal{X}} |Q(x) - \hat{Q}_\alpha^S(x)| \leq C \sum_{i=1}^m \mathbb{E}_{\omega_{-i}} [|\Delta| f_i(\cdot | \omega_{-i})]$ w.p. 1 as $S \rightarrow \infty$.

Proof Since

$$\sup_{x \in \mathcal{X}} |Q(x) - \hat{Q}_\alpha^S(x)| \leq \sup_{x \in \mathcal{X}} |Q(x) - \hat{Q}_\alpha(x)| + \sup_{x \in \mathcal{X}} |\hat{Q}_\alpha(x) - \hat{Q}_\alpha^S(x)|,$$

the result follows directly by combining Theorem 2 and Lemma 2. □

Corollary 1 implies a similar error bound on the difference between the optimal values η^* and $\hat{\eta}_\alpha^S$ of the original MIR problem (1) and the SAA problem (11), respectively, see Corollary 2.

Corollary 2 Consider the optimal values η^* and $\hat{\eta}_\alpha^S$ of the MIR problem (1) and the SAA problem (11). Then,

$$\hat{\eta}_\alpha^S - \eta^* \leq C \sum_{i=1}^m \mathbb{E}_{\omega_{-i}} [|\Delta| f_i(\cdot | \omega_{-i})]$$

w.p. 1 as $S \rightarrow \infty$.

Proof Since the optimal solution x^* of (1) is feasible but not necessarily optimal in (11), we have

$$\hat{\eta}_\alpha^S - \eta^* \leq cx^* + \hat{Q}_\alpha^S(x^*) - \eta^* = \hat{Q}_\alpha^S(x^*) - Q(x^*) \leq \sup_{x \in \mathcal{X}} |Q(x) - \hat{Q}_\alpha^S(x)|.$$

The result follows directly from Corollary 1. \square

4.1.2 Asymptotic tightness of loose optimality cuts

If we derive a loose optimality cut $\hat{Q}_\alpha^S(x) \geq \beta_{\tau+1}x + \delta_{\tau+1}$ for \hat{Q}_α^S at x^τ in LBDA(α), then the gap $\hat{Q}_\alpha^S(x^\tau) - (\beta_{\tau+1}x^\tau + \delta_{\tau+1})$ may be positive, since the cut is not necessarily tight at x^τ . However, we derive a bound on this gap by considering the index function

$$\hat{k}(\omega, x) \in \arg \max_{k=1, \dots, K} \lambda^k(\omega - Tx), \quad \omega \in \Omega, \quad x \in X.$$

Note that $\hat{k}(\omega^s, x^\tau) = \hat{k}_s^\tau$, i.e. the index function $\hat{k}(\omega, x)$ is used to derive our loose optimality cuts. Based on this index function, we define the lower bounding function L_α^S of \hat{Q}_α^S , given by

$$L_\alpha^S(x) := \frac{1}{S} \sum_{s=1}^S \left[\lambda^{\hat{k}(\omega^s, x)}(\omega^s - Tx) + \psi^{\hat{k}(\omega^s, x)}(\omega^s - \alpha) \right], \quad x \in X, \quad (15)$$

and we note that $L_\alpha^S(x^\tau)$ is the value of the loose optimality cut at x^τ , i.e. $L_\alpha^S(x^\tau) = \beta_{\tau+1}x^\tau + \delta_{\tau+1}$. The function L_α^S is a lower bound of \hat{Q}_α^S since its corresponding value function, defined as

$$v_\alpha(\omega, x) := \lambda^{\hat{k}(\omega, x)}(\omega - Tx) + \psi^{\hat{k}(\omega, x)}(\omega - \alpha), \quad x \in X, \quad (16)$$

is a lower bound of $\hat{v}_\alpha(\omega, x)$ for all $\omega \in \Omega$ and $x \in X$.

Proposition 1 contains a uniform total variation bound on the difference between L_α^S and \hat{Q}_α^S . In order to derive this result, we first analyze the difference between v_α and \hat{v}_α . It turns out that v_α equals \hat{v}_α on large parts of the domain, see Lemma 3.

Lemma 3 Consider the value functions v_α and \hat{v}_α , with v_α defined in (16) and \hat{v}_α defined as

$$\hat{v}_\alpha(\omega, x) = \max_{k=1, \dots, K} \left\{ \lambda^k(\omega - Tx) + \psi^k(\omega - \alpha) \right\}. \quad (17)$$

Let Λ^k , $k = 1, \dots, K$, denote the closed convex cones from Lemma 1. Then, there exist vectors $\sigma_k \in \Lambda^k$ and a constant $R > 0$ such that

- (i) $0 \leq \hat{v}_\alpha(\omega, x) - v_\alpha(\omega, x) \leq R$ for all $\omega \in \mathbb{R}^m$ and $x \in \mathbb{R}^{n_1}$, and
- (ii) $\omega - Tx \in \sigma_k + \Lambda^k \implies v_\alpha(\omega, x) = \hat{v}_\alpha(\omega, x) = \lambda^k(\omega - Tx) + \psi^k(\omega - \alpha)$.

Proof See “Appendix”. \square

Proposition 1 Consider the SAA of the generalized α -approximation \hat{Q}_α^S and its lower bound L_α^S , defined in (15). There exists a constant $C > 0$ such that for every continuous random vector ω with joint pdf $f \in \mathcal{H}^m$,

$$\sup_{x \in \mathcal{X}} \left| \hat{Q}_\alpha^S(x) - L_\alpha^S(x) \right| \leq C \sum_{i=1}^m \mathbb{E}_{\omega_{-i}} \left[|\Delta| f_i(\cdot | \omega_{-i}) \right]$$

w.p. 1 as $S \rightarrow \infty$.

Proof Define $\Delta(\omega, x) := \hat{v}_\alpha(\omega, x) - v_\alpha(\omega, x)$, so that

$$\hat{Q}_\alpha^S(x) - L_\alpha^S(x) = \frac{1}{S} \sum_{s=1}^S \Delta(\omega^s, x).$$

We derive an upper bound on $\Delta(\omega, x)$, independent of x . We then apply the strong law of large numbers (SLLN) to obtain the desired result.

If we define $\mathcal{M} = \cup_{k=1}^K (\sigma_k + \Lambda_k)$, where σ_k and Λ^k , $k = 1, \dots, K$, are the vectors and closed convex cones from Lemma 3, then

$$\Delta(\omega, x) \leq \xi(\omega, x) := \begin{cases} R, & \text{if } \omega - Tx \notin \mathcal{M}, \\ 0, & \text{if } \omega - Tx \in \mathcal{M}, \end{cases}$$

where R is the upper bound on $\hat{v}_\alpha(\omega, x) - v_\alpha(\omega, x)$ from Lemma 3. Moreover, we can derive a bound on $\mathbb{P}[\omega - Tx \in \mathcal{M}]$. Unfortunately, the random variable $\xi(\omega, x)$ depends on x . Therefore, we cannot apply the SLLN to

$$\sup_{x \in \mathcal{X}} \frac{1}{S} \sum_{s=1}^S \xi(\omega^s, x)$$

if \mathcal{X} is infinite. To resolve this, we use that \mathcal{X} is bounded. Let D denote the diameter of $T\mathcal{X}$, i.e., $\|Tx - Tx'\| \leq D$ for all $x, x' \in \mathcal{X}$. Define $\mathcal{M}' \subset \mathcal{M}$ as $\mathcal{M}' :=$

$\bigcup_{k=1}^K (\sigma_k + \Lambda^k)(D)$. Fix an arbitrary $\bar{x} \in X$. Note that for all $x \in \mathcal{X}$,

$$\begin{aligned} \omega - T\bar{x} \in \mathcal{M}' &\implies \exists k : \omega - T\bar{x} \in (\sigma_k + \Lambda^k)(D) \\ &\implies \exists k : \omega - Tx \in (\sigma_k + \Lambda^k) \implies \omega - Tx \in \mathcal{M}. \end{aligned}$$

We obtain

$$\Delta(\omega, x) \leq \xi(\omega) := \begin{cases} R & \text{if } \omega - T\bar{x} \notin \mathcal{M}' \\ 0 & \text{if } \omega - T\bar{x} \in \mathcal{M}' \end{cases}$$

Note that $\bar{\xi}(\omega)$ only depends on a fixed $x^* \in \mathcal{X}$ and is independent of x . By the SLLN,

$$\frac{1}{S} \sum_{s=1}^S \bar{\xi}(\omega^s) \rightarrow R \mathbb{P}[\omega - T\bar{x} \notin \mathcal{M}'],$$

w.p. 1 as $S \rightarrow \infty$.

By [23, Lemma 3.9], $\mathbb{R}^m \setminus \mathcal{M}'$ can be covered by finitely many hyperslices, that is,

$$\mathbb{R}^m \setminus \mathcal{M}' \subset \bigcup_{j=1}^J H_j,$$

where the hyperslices H_j are defined as

$$H_j := \{x \in \mathbb{R}^m : 0 \leq a_j^T x \leq \delta_j\},$$

for some a_j^T and δ_j . By [23, Theorem 4.6], there exists a constant $\beta > 0$ such that

$$\mathbb{P}[\omega - Tx^* \notin \mathcal{M}'] \leq \beta \sum_{i=1}^m \mathbb{E}_{\omega_{-i}} [|\Delta| f_i(\cdot | \omega_{-i})].$$

The result now follows from

$$\begin{aligned} \sup_{x \in \mathcal{X}} \left| \hat{Q}_\alpha^S(x) - L_\alpha^S(x) \right| &= \sup_{x \in \mathcal{X}} \frac{1}{S} \sum_{s=1}^S \Delta(\omega^s, x) \\ &\leq \frac{1}{S} \sum_{s=1}^S \bar{\xi}(\omega^s) \rightarrow R \mathbb{P}[\omega - Tx^* \notin \mathcal{M}'] \\ &\leq C \sum_{i=1}^m \mathbb{E}_{\omega_{-i}} [|\Delta| f_i(\cdot | \omega_{-i})], \end{aligned}$$

w.p. 1 as $S \rightarrow \infty$, where $C = R\beta$. □

Proposition 1 shows that our loose optimality cuts are asymptotically tight, since for every iteration τ in LBDA(α) the loose optimality cut for \hat{Q}_α^S at x^τ is tight for L_α^S at x^τ . Hence, if the underlying total variations are small enough and if S is large enough, then the loose optimality cut is nearly tight for \hat{Q}_α^S at x^τ .

4.2 Error bound on the optimality gap of LBDA(α)

We are now ready to prove the performance guarantee of LBDA(α) in Theorem 3. Intuitively, we are able to derive this bound since (i) our loose optimality cuts are asymptotically tight, and thus \hat{x}_α is near-optimal for the SAA problem with \hat{Q}_α^S , (ii) the SAA \hat{Q}_α^S converges uniformly to \hat{Q}_α , and (iii) Theorem 2 contains a uniform error on the difference between \hat{Q}_α and Q .

Proof of Theorem 3 Let \hat{x}_α denote the solution returned by LBDA(α). Since $\hat{x}_\alpha := x^\tau$ is the current solution in the final iteration τ of the algorithm, it follows that \hat{x}_α is a minimizer of

$$\min_x \{cx + \hat{Q}_{\text{out}}^\tau(x) : Ax = b, x \in X\}.$$

Moreover, since $\hat{Q}_{\text{out}}^\tau \leq \hat{Q}_\alpha^S$, it follows that $c\hat{x}_\alpha + \hat{Q}_{\text{out}}^\tau(\hat{x}_\alpha) \leq \hat{\eta}_\alpha^S$, and thus, by rearranging terms and adding \hat{Q}_α^S on both sides,

$$c\hat{x}_\alpha + \hat{Q}_\alpha^S(\hat{x}_\alpha) - \hat{\eta}_\alpha^S \leq \hat{Q}_\alpha^S(\hat{x}_\alpha) - \hat{Q}_{\text{out}}^\tau(\hat{x}_\alpha). \quad (18)$$

The right-hand side of (18) represents an upper bound on the optimality gap of \hat{x}_α in the SAA problem (11). The termination criterion of LBDA(α) guarantees that this upper bound is not too large. Indeed, at termination it holds that

$$\hat{Q}_{\text{out}}^\tau(\hat{x}_\alpha) \geq \hat{Q}_{\text{out}}^{\tau+1}(\hat{x}_\alpha) - \varepsilon \geq L_\alpha^S(\hat{x}_\alpha) - \varepsilon,$$

and thus the upper bound in (18) reduces to

$$\begin{aligned} c\hat{x}_\alpha + \hat{Q}_\alpha^S(\hat{x}_\alpha) - \hat{\eta}_\alpha^S &\leq \hat{Q}_\alpha^S(\hat{x}_\alpha) - L_\alpha^S(\hat{x}_\alpha) + \varepsilon \\ &\leq \sup_{x \in \mathcal{X}} |\hat{Q}_\alpha^S(x) - L_\alpha^S(x)| + \varepsilon. \end{aligned} \quad (19)$$

In the end, however, we are not interested in the optimality gap of \hat{x}_α in the SAA problem (11) but in the optimality gap

$$c\hat{x}_\alpha + Q(\hat{x}_\alpha) - \eta^*$$

of \hat{x}_α in the original MIR problem (1). Adding and subtracting both $\hat{Q}_\alpha^S(\hat{x}_\alpha)$ and $\hat{\eta}_\alpha^S$, and using (19), yields

$$\begin{aligned}
c\hat{x}_\alpha + Q(\hat{x}_\alpha) - \eta^* &= (c\hat{x}_\alpha + Q(\hat{x}_\alpha) - \hat{\eta}_\alpha^S) + (Q(\hat{x}_\alpha) - \hat{Q}_\alpha^S(\hat{x}_\alpha)) + (\hat{\eta}_\alpha^S - \eta^*) \\
&\leq \sup_{x \in \mathcal{X}} |\hat{Q}_\alpha^S(x) - L_\alpha^S(x)| + \varepsilon + \sup_{x \in \mathcal{X}} |Q(x) - \hat{Q}_\alpha^S(x)| + \hat{\eta}_\alpha^S - \eta^*.
\end{aligned}$$

Applying Corollaries 1 and 2, and Proposition 1, we conclude that there exists a constant $C > 0$ such that

$$c\hat{x}_\alpha + Q(\hat{x}_\alpha) - \eta^* \leq C \sum_{i=1}^m \mathbb{E}_{\omega_{-i}} [|\Delta| f_i(\cdot | \omega_{-i})] + \varepsilon,$$

w.p. 1 as $S \rightarrow \infty$. □

The performance guarantee for LBDA(α) in Theorem 3 is a worst-case bound. For many problem instances, the actual performance may be much better. In Sect. 5, we assess the performance of LBDA(α) empirically on a wide range of test instances.

5 Numerical experiments

We test the performance of LBDA(α) on problem instances from the literature and on randomly generated instances, see Sects. 5.2 and 5.3, respectively. In particular, we consider (variations of) an investment planning problem in [29], and two classes of problem instances available from SIPLIB [2], namely the SIZES problem [13] and the stochastic server location problem (SSLP) [18]. First, however, we describe the set-up of our numerical experiments in Sect. 5.1.

5.1 Set-up of numerical experiments

In our numerical experiments, we compare LBDA(α) to several benchmark methods, in terms of costs, relative optimality gaps, and computations times. Since the performance of LBDA(α) depends on α , we investigate four different approaches to select α .

First, we take α equal to the zero vector. Second, we take $\alpha = \alpha^* := Tx^*$, where x^* is the optimal solution of the original problem. Obviously, for large problem instances x^* is unknown, however, we expect that α^* is a good choice of α since the generalized α -approximations are obtained by replacing Tx by α in the Gomory relaxations, and thus \hat{Q}_{α^*} is a good approximation of Q near the true optimal solution x^* . We test this for the smaller problem instances for which x^* is known.

Since α^* , however, typically cannot be computed for larger problem instances, we also propose an iterative scheme, in which we first obtain \hat{x}_{α_0} by running LBDA(α_0), where α_0 is the zero vector. Next, we run LBDA(α_1), where $\alpha_1 := T\hat{x}_{\alpha_0}$. We extend this scheme to 100 iterations by recursively defining $\alpha_{k+1} := T\hat{x}_{\alpha_k}$, $k = 1, \dots, 100$. We then select the best value of α in terms of expected costs, denoted by $\alpha^\#$. Finally, we apply LBDA(α) multiple times using 100 different values of α , drawn from an multivariate uniform distribution on $[0, 100]^m$, and we denote the best value of α in terms of the expected costs by α^+ .

In order to compare these approaches, note that the expected costs $cx + Q(x)$ of a candidate solution x can be computed exactly if the random vector ω has a finite number of realizations, as is the case for the SIZES, SSLP, and investment planning problems that we consider. Therefore, if the optimal value η^* is known, then the relative optimality gap $\rho(x)$, defined as

$$\rho(x) = \frac{cx + Q(x) - \eta^*}{|\eta^*|} * 100\%,$$

can be computed exactly, and otherwise bounds on $\rho(x)$ can be computed.

In contrast, for our randomly generated instances, we assume that ω is continuously distributed. For these instances, we use the multiple replications procedure (MRP) [16] with Latin hypercube sampling [5] to obtain 95% confidence upper bounds on $\rho(x)$. Moreover, we compare the performance of LBDA(α) to a range of benchmark solutions, using out-of-sample estimation of $cx + Q(x)$, with a sample size of 10^5 , which guarantees that the standard errors of our results are sufficiently small. The benchmark and LBDA(α) solutions are computed using a sample of size $S = 1000$.

The first benchmark solution \bar{x}_S is obtained by solving the deterministic equivalent formulation (DEF) of the corresponding SAA of the original problem (1). The DEF is a large-scale MIP, which, typically, cannot be solved in reasonable time by standard MIP solvers. Hence we also solve the DEF using a smaller sample size $S' = 100$, resulting in the second benchmark solution $\bar{x}_{S'}$.

We obtain three additional benchmark solutions by solving the generalized α -approximations exactly for $\alpha = 0$, $\alpha = \alpha^*$ and $\alpha = \alpha^+$, that is, we find the optimal solution x_α^* of the approximating problem (4) with $\hat{Q} = \hat{Q}_\alpha^S$. We do so by solving the approximation second-stage problems

$$\max_{k=1, \dots, K} \{\lambda^k(\omega - Tx) + \psi^k(\omega - \alpha)\}$$

by enumeration over $k = 1, \dots, K$. For this reason, x_α^* can only be computed in reasonable time for small problem instances.

Finally, we consider two trivial benchmark solutions, which we expect to outperform significantly. First, we relax the integer restrictions on the second-stage variables in the SAA of the original problem (1), resulting in the benchmark solution x_{LP} . Second, we solve the Jensen approximation, which replaces the distribution of ω by a degenerate distribution at $\mu = \mathbb{E}_\omega[\omega]$, and denote the optimal solution by x_μ .

We run our experiments on a single Intel Xeon E5 2680v3 core @2.5GHz with Gurobi 7.0.2. To ensure a fair comparison between solutions, we use common random numbers where possible and we limit the computation time of each algorithm to two hours.

Table 1 The SIZES problem

Instance	Computation time (s) (optimality gap)				
	DEF	LBDA(α)			
	x^*	$\alpha = 0$	$\alpha = \alpha^*$	$\alpha = \alpha^\#$	$\alpha = \alpha^+$
SIZES3	0.2 (0.0%)	0.2 (0.35%)	0.3 (0.20%)	24.8 (0.32%)	23.8 (0.20%)
SIZES5	1.5 (0.0%)	0.3 (0.44%)	0.6 (0.26%)	34.0 (0.35%)	34.0 (0.14%)
SIZES10	312.1 (0.0%)	0.7 (0.47%)	0.9 (0.08%)	68.6 (0.10%)	53.8 (0.11%)

5.2 Test instances from the literature

5.2.1 The SIZES problem

We first consider all instances of the SIZES test problem suite [13] from SIPLIB. These instances have mixed-binary variables in both stages, and differ in the number of scenarios, namely 3, 5, and 10. The DEF of the largest instance has 341 constraints and 825 variables, of which 110 are binary. We refer to [2] for further details. In Table 1, we report the outcomes of LBDA(α) and solving the DEF.

We observe from Table 1 that LBDA(α) performs very well for all choices of α . Indeed, on every instance, the optimality gaps of all LBDA(α) solutions are below 0.5%, and below 0.2% for $\alpha = \alpha^+$. Moreover, LBDA(α) runs very fast for all instances: for a single value of α , the computation time of LBDA(α) is always below one second.

Another observation is that LBDA($\alpha^\#$) and LBDA(α^+) consistently outperform LBDA(0), at the expense of additional computation time. Nevertheless, the computation times of LBDA(α) for $\alpha^\#$ and α^+ are still moderate, and they scale much better to larger instances than solving the DEF. In particular, the time taken to solve the DEF grows exponentially as the sample size S increases, whereas the computation times of LBDA(α) are approximately linear in S . Finally, LBDA(α) performs very well if $\alpha = \alpha^*$. However, since α^* is not known in practice, it is useful to observe that LBDA(α^+) achieves similar performance.

5.2.2 The stochastic server location problem

The SSLP instances are more challenging in terms of input size than the SIZES instances. Indeed, the DEF of the largest instance has over 1,000,000 binary decision variables and 120,000 constraints. Their first- and second-stage problems are pure binary and mixed-binary, respectively, and ω follows a discrete distribution. A full problem description can be found in [18], in which the instances are solved using the D^2 algorithm of [31]. See also [1] and [12] for more recent computational experiments on these test instances using other exact approaches. In Table 2 we report the best known running time for each SSLP instance over all these exact approaches, along with the outcomes of LBDA(α).

Table 2 The stochastic server location problem

Instance	Computation time (s) (optimality gap)				
	Exact approaches				
	x^*	LBDA(α)			
		$\alpha = 0$	$\alpha = \alpha^*$	$\alpha = \alpha^\#$	$\alpha = \alpha^+$
SSLP_5_25_50	0.5 ¹ (0.0%)	0.2 (0.0%)	0.2 (0.0%)	2.8 (0.0%)	2.8 (0.0%)
SSLP_5_25_100	1.0 ¹ (0.0%)	0.2 (0.0%)	0.2 (0.0%)	5.3 (0.0%)	5.3 (0.0%)
SSLP_15_45_5	4.3 ² (0.0%)	1.2 (0.45%)	2.1 (0.0%)	79.3 (0.45%)	13.5 (0.0%)
SSLP_15_45_10	35.3 ² (0.0%)	2.9 (0.0%)	2.7 (0.0%)	101.5 (0.0%)	10.8 (0.0%)
SSLP_15_45_15	92.7 ² (0.0%)	4.6 (0.0%)	5.6 (0.0%)	270.7 (0.0%)	105.4 (0.0%)
SSLP_10_50_50	65.7 ² (0.0%)	0.6 (0.0%)	1.1 (0.0%)	102.0 (0.0%)	48.3 (0.0%)
SSLP_10_50_100	153.3 ² (0.0%)	1.9 (0.0%)	1.9 (0.0%)	137.0 (0.0%)	121.5 (0.0%)
SSLP_10_50_500	1033.0 ³ (0.0%)	8.7 (0.0%)	8.6 (0.0%)	525.7 (0.0%)	482.1 (0.0%)
SSLP_10_50_1000	2691.0 ² (0.0%)	13.5 (0.0%)	58.7 (0.0%)	5529.5 (0.0%)	903.5 (0.0%)
SSLP_10_50_2000	4952.0 ² (0.0%)	55.7 (0.0%)	73.6 (0.0%)	6020.8 (0.0%)	1698.9 (0.0%)

¹See [18]. ²See [1]. ³See [12]

Strikingly, LBDA(α) was able to solve all instances to optimality for $\alpha = \alpha^+$ and $\alpha = \alpha^*$. Moreover, LBDA(0) and LBDA($\alpha^\#$) solved all instances except SSLP_15_45_5, on which both achieved an optimality gap of 0.45%. In terms of computation time, LBDA(0) is clearly preferred to LBDA($\alpha^\#$) and LBDA(α^+), while achieving similar results. Finally, although directly comparing LBDA(α) to the other approaches is not completely fair since the algorithms were run on different machines, it is clear that LBDA(α) is generally faster than exact approaches. Indeed, LBDA(0) solved all instances in under one minute, and eight out of ten instances were solved in less than ten seconds, whereas the fastest exact approach required at least one minute for six out of ten instances, and over one hour for the largest instance.

5.2.3 An investment planning problem

We consider the following problem by Schultz et al. [29],

$$\min_x \{-3/2x_1 - 4x_2 + \mathbb{E}_\omega[v(\omega, x)] : x \in [0, 5]^2\},$$

where

$$v(\omega, x) = \min_{y \in Y} \{-16y_1 - 19y_2 - 23y_3 - 28y_4 : 2y_1 + 3y_2 + 4y_3 + 5y_4 \leq \omega_1 - x_1 \\ 6y_1 + y_2 + 3y_3 + y_4 \leq \omega_2 - x_2\},$$

and where the second-stage decision variables are binary, i.e. $Y = \{0, 1\}^4$, and the random vector $\omega = (\omega_1, \omega_2)$ follows a discrete distribution which assigns equal probabilities to $S = 441$ equidistant lattice points of $[5, 15]^2$. Schultz et al. consider a second variant of this problem by choosing the technology matrix T as

$$T = H := \begin{pmatrix} 2/3 & 1/3 \\ 1/3 & 2/3 \end{pmatrix},$$

whereas in the original formulation, T is the identity matrix I_2 . For both variants, we consider $S \in \{4, 9, 36, 121, 441, 1681, 10201\}$ and $Y = \mathbb{Z}_+^4$, in addition to $Y = \{0, 1\}^4$, as is done in [19]. For each of the resulting 28 instances, Table 3 shows the results of LBDA(α) and solving the DEF. Note that if Gurobi could not solve an instance within two hours, then we report the gaps to \hat{x}^* , the best solution that Gurobi was able to find.

Overall, LBDA(α) performs well on the instances in Table 3. In particular, for $\alpha = \alpha^\#$ and $\alpha = \alpha^+$, LBDA(α) achieves gaps that are below 2% on 22 and 23 out of 28 instances, respectively. In general, LBDA(α) achieves better results if S is larger. For example, if $Y = \{0, 1\}^4$ and $T = H$, then the gaps are strictly decreasing in S . This is in line with the performance guarantee of LBDA(α) in Theorem 3: if S is larger, then the distributions of ω_1 and ω_2 more closely resemble a continuous uniform distribution on $[5, 15]$, which has small total variation, i.e. the error bound in Theorem 3 is small.

Table 3 An investment planning problem

Instance			Computation time (s) (gap to \hat{x}^*)				
			DEF	LBDA(α)			
Y	T	S	\hat{x}^*	$\alpha = 0$	$\alpha = T\hat{x}^*$	$\alpha = \alpha^\#$	$\alpha = \alpha^+$
\mathbb{Z}_+^4	H	4	0	0.0 (0.0%)	0.0 (0.0%)	0.4 (0.0%)	0.4 (0.0%)
		9	0	0.0 (12.9%)	0.0 (0.0%)	0.7 (2.2%)	0.7 (0.1%)
		36	0.5	0.0 (5.0%)	0.0 (5.8%)	3.0 (0.5%)	2.9 (0.0%)
		121	8.2	0.1 (4.0%)	0.1 (4.0%)	7.6 (4.0%)	7.7 (4.0%)
		441	7200 ¹	0.3 (3.1%)	0.2 (3.1%)	27.6 (3.1%)	27.8 (3.1%)
		1681	7200 ¹	0.9 (0.0%)	0.9 (0.0%)	104.4 (0.0%)	105.6 (0.0%)
		10201	7397 ¹	5.5 (−0.2%)	5.4 (−0.2%)	631.8 (−0.2%)	639.1 (−0.2%)
\mathbb{Z}_+^4	I_2	4	0	0.0 (0.0%)	0.0 (0.0%)	0.3 (0.0%)	0.3 (0.0%)
		9	0	0.0 (0.9%)	0.0 (1.0%)	0.8 (0.9%)	0.8 (0.0%)
		36	0	0.0 (4.3%)	0.0 (0.9%)	2.9 (0.9%)	2.9 (0.0%)
		121	0.5	0.1 (4.0%)	0.1 (2.8%)	9.8 (0.0%)	9.9 (0.0%)
		441	20.5	0.3 (0.9%)	0.3 (1.5%)	36.5 (0.2%)	36.1 (0.0%)
		1681	7200 ¹	1.3 (0.7%)	1.2 (0.1%)	136.1 (0.0%)	138.2 (0.0%)
		10201	7269 ¹	8.2 (0.3%)	7.9 (0.0%)	864.5 (−0.1%)	883.0 (−0.1%)
$\{0, 1\}^4$	H	4	0	0.0 (8.8%)	0.0 (8.8%)	0.2 (8.8%)	0.2 (8.8%)
		9	0	0.0 (5.0%)	0.0 (5.0%)	0.4 (5.0%)	0.5 (5.0%)
		36	0.7	0.0 (1.6%)	0.0 (1.6%)	1.9 (1.6%)	2.0 (1.6%)
		121	32.5	0.1 (1.8%)	0.1 (1.8%)	6.7 (1.8%)	6.7 (1.8%)
		441	7200 ¹	0.3 (2.1%)	0.3 (2.1%)	24.2 (2.1%)	24.3 (2.1%)
		1681	7200 ¹	0.9 (0.3%)	0.9 (0.3%)	92.0 (0.3%)	91.9 (0.3%)
		10201	7204 ¹	5.0 (0.1%)	5.1 (0.1%)	554.5 (0.1%)	553.4 (0.1%)
$\{0, 1\}^4$	I_2	4	0	0.0 (1.9%)	0.0 (14.8%)	0.3 (1.9%)	0.3 (0.1%)
		9	0	0.0 (1.4%)	0.0 (0.0%)	0.6 (1.4%)	0.7 (0.1%)
		36	0	0.0 (5.0%)	0.0 (2.6%)	3.4 (2.0%)	3.3 (1.0%)
		121	0.9	0.1 (2.1%)	0.1 (4.7%)	10.7 (1.5%)	11.1 (0.4%)
		441	118.3	0.4 (2.5%)	0.4 (0.3%)	41.6 (1.3%)	42.5 (0.4%)
		1681	7204 ¹	1.8 (0.9%)	1.8 (1.1%)	162.1 (0.9%)	166.5 (0.3%)
		10201	7293 ¹	10.5 (1.1%)	10.9 (0.9%)	1032.7 (0.8%)	1055.1 (0.0%)

¹ DEF could not be solved in time: \hat{x}^* denotes best solution found by Gurobi.

On some instances, LBDA(α) does not consistently perform well for all values of α . For example, if $Y = \mathbb{Z}_+^4$, $T = H$, and $S = 9$, then LBDA(α^+) and LBDA(0) achieve gaps of 0.1% and 12.9%, respectively. Furthermore, the instances with $Y = \{0, 1\}^4$, $T = H$, and $S \in \{4, 9\}$ turn out to be difficult instances for LBDA(α): for all choices of α , the resulting gaps are 8.8% and 5%, respectively.

In fact, for every choice of α , LBDA(α) achieves identical gaps if $Y = \{0, 1\}^4$ and $T = H$, but the gaps are much smaller if $S \geq 36$, e.g. if $S = 10201$, then the gaps

are 0.1%. In contrast, there are large differences between the different choices of α if $Y = \{0, 1\}^4$ and $T = I$. For these instances, $\alpha = \alpha^\#$ and $\alpha = \alpha^+$ outperform the other choices of α . However, similar as for the SIZES instances in Sect. 5.2.1, there is a trade-off between performance and computation times, since $\text{LBDA}(\alpha^+)$ and $\text{LBDA}(\alpha^\#)$ require 100 $\text{LBDA}(\alpha)$ runs, which is computationally more demanding than $\text{LBDA}(0)$. Nevertheless, on every instance, the computation times of $\text{LBDA}(\alpha)$ for $\alpha \in \{\alpha^\#, \alpha^+\}$ are below 20 minutes, and below 3 minutes if $S \leq 1681$.

5.3 Randomly generated test instances

We generate random MIR problems of the form (1), with $X = \mathbb{R}_+^{n_1}$ and

$$Q(x) = \mathbb{E}_\omega \left[\min_y \{qy : Wy \geq \omega - Tx, y \in \mathbb{Z}_+^p\} \right], \quad x \in X.$$

In addition, we assume that the components of the random vector $\omega \in \mathbb{R}^m$ follow independent normal distributions with mean 10 and standard deviation $\sigma \in \{0.1, 0.5, 1, 2, 4, 10\}$. The parameters c, q, T , and W are fixed, and their elements are drawn from discrete uniform distributions with supports contained in $[1, 5]$, $[5, 10]$, $[1, 6]$, and $[1, 6]$, respectively.

The reason that we consider multiple values of the standard deviation σ is that Theorem 3 implies that $\text{LBDA}(\alpha)$ performs better as σ increases. This is because the total variations of the one-dimensional conditional pdf are small if σ is large, and thus the error bound on the optimality gap achieved by $\text{LBDA}(\alpha)$ is also small.

To prevent noise in the outcomes of our experiments, we compute the average optimality gaps, costs, and computation times over 20 randomly generated test instances for each value of σ . We consider test instances of two different sizes, namely $n_1 = 10$, $p = 5$, $m = 5$ (small), and $n_1 = 100$, $p = 40$, $m = 20$ (large). Tables 4 and 5 display the results for the small and large versions, respectively.

From these results, we observe that $\text{LBDA}(\alpha)$ clearly outperforms the sampling solutions in terms of computation time and scalability to larger problem instances. In particular, we observe that the computation time of $\text{LBDA}(\alpha)$ is of the same order of magnitude as that of x_{LP} , while it performs significantly better in terms of optimality gaps and out-of-sample estimated expected costs. Undeniably, our results indicate that $\text{LBDA}(\alpha)$ can be implemented very efficiently and that it can handle large MIR problem instances.

In line with the performance guarantee in Theorem 3, $\text{LBDA}(\alpha)$ performs better for larger values of σ . For example, on the small instances, the optimality gaps achieved by $\text{LBDA}(0)$ are strictly decreasing in σ , and for $\sigma = 10.0$, $\text{LBDA}(\alpha^+)$ outperforms the sampling solution \bar{x}_S . A similar observation is true for the optimal solution of the generalized α -approximations \hat{Q}_α , as we would expect based on the error bound for \hat{Q}_α in Theorem 2. Observe, however, that even for large values of σ , the optimality gaps reported in Table 4 for $\text{LBDA}(\alpha)$ are relatively large, i.e. around 3–4%, and that the optimality gaps achieved by e.g. $\text{LBDA}(\alpha^+)$ are not strictly decreasing in σ . The reason is that the actual optimality gaps are likely much smaller than the upper bounds

Table 4 Randomly generated test instances ($n_1 = 10$, $p = 5$, $m = 5$)

σ	Computation time (s) (optimality gap)		x_α^*							LBDA(α)				
	Benchmarks		$\bar{x}_{S'}$	x_μ	x_{LP}	$\alpha = 0$	$\alpha = T \bar{x}_S$	$\alpha = \alpha^+$	$\alpha = 0$	$\alpha = T \bar{x}_S$	$\alpha = \alpha^\#$	$\alpha = \alpha^+$		
0.1	677.8 (0.3%)	0.1 (2.6%)	0.0 (102.5%)	2.3 (37.0%)	169.3 (28.3%)	83.2 (0.3%)	86.3 (11.8%)	4.0 (24.5%)	3.8 (0.4%)	389.4 (15.0%)	358.3 (3.2%)			
0.5	7125.0 ² (1.1%)	0.3 (2.3%)	0.0 (84.8%)	1.8 (28.1%)	169.6 (17.0%)	106.3 (1.1%)	96.6 (7.9%)	4.4 (12.1%)	2.7 (1.6%)	206.8 (5.8%)	200.9 (1.7%)			
1.0	7200 ¹ (1.9%)	1.0 (2.8%)	0.0 (71.9%)	1.7 (21.2%)	176.4 (10.2%)	121.5 (2.4%)	104.9 (7.0%)	2.9 (9.0%)	2.9 (9.0%)	351.5 (4.9%)	313.0 (2.3%)			
2.0	7200 ¹ (3.0%)	368.6 ⁴ (4.1%)	0.0 (60.0%)	1.9 (13.8%)	173.1 (5.6%)	150.7 (3.3%)	128.5 (4.9%)	3.3 (7.7%)	3.4 (5.0%)	277.5 (4.5%)	243.6 (3.6%)			
4.0	7200 ¹ (4.1%)	436.9 ⁴ (4.6%)	0.0 (65.2%)	2.0 (8.4%)	180.0 (4.8%)	125.0 (4.1%)	94.0 (4.6%)	3.4 (6.7%)	3.5 (5.2%)	303.6 (5.0%)	275.5 (4.2%)			
10.0	7200 ¹ (3.3%)	1119.2 ³ (4.0%)	0.0 (87.6%)	2.3 (4.2%)	173.8 (3.3%)	143.8 (3.3%)	126.8 (3.3%)	3.8 (3.7%)	4.1 (3.5%)	344.9 (3.3%)	303.3 (3.2%)			

¹ 0/20 instances solved in time² 1/20 instances solved in time³ 17/20 instances solved in time⁴ 19/20 instances solved in time

Table 5 Randomly generated test instances ($m_1 = 100, p = 40, m = 20$)

σ	Computation time (s) (gap to \bar{x}_S)						
	Benchmarks						
	\bar{x}_S	$\bar{x}_{S'}$	x_{μ}	x_{LP}	LBDA(α)		
					$\alpha = 0$	$\alpha = T \bar{x}_S$	$\alpha = \alpha^\#$
0.1	1519.9	1.7 (13.0%)	0.0 (142.8%)	25.4 (42.7%)	34.3 (31.7%)	31.2 (0.4%)	995.4 (6.7%)
0.5	7200 ¹	3.9 (9.9%)	0.0 (116.6%)	26.8 (33.3%)	33.4 (22.3%)	32.4 (2.0%)	1196.5 (4.0%)
1.0	7200 ¹	4.9 (8.0%)	0.0 (93.0%)	25.6 (25.0%)	31.2 (11.4%)	31.9 (3.0%)	1008.7 (4.1%)
2.0	7200 ¹	38.5 (5.7%)	0.0 (85.6%)	23.6 (14.9%)	30.3 (6.2%)	29.2 (4.1%)	1062.9 (3.2%)
4.0	7200 ¹	165.5 (4.0%)	0.0 (105.4%)	20.7 (6.3%)	28.9 (3.1%)	28.9 (2.2%)	1244.3 (1.6%)
10.0	7200 ¹	5692.5 ² (2.5%)	0.0 (132.6%)	18.4 (1.3%)	29.0 (1.0%)	28.9 (0.5%)	1502.4 (0.3%)

¹ 0/20 instances solved in time

² 7/20 instances solved in time

reported in Table 4. This is because they are computed using the MRP, which relies on solving multiple SAAs of the original problem. Since, however, Gurobi has difficulties solving the DEFs of these SAAs in reasonable time, especially for large values of σ , the bounds obtained using the MRP are typically not sharp.

Interestingly, x_{LP} also performs better as σ increases. An explanation is that our error bound for the generalized α -approximation implies that the MIR function Q becomes closer to a convex function as σ increases. Thus, since the LP-relaxation of Q is a convex lower bound of Q , its approximation error is expected to become smaller as σ increases. Note however, that unlike the generalized α -approximations, the approximation error of the LP-relaxation does not go to zero. Indeed, based on our results, LBDA(α) is clearly preferred to x_{LP} , since LBDA(α) consistently outperforms x_{LP} , at the expense of very little additional computation time.

Furthermore, the results in Tables 4 and 5 indicate that $\alpha = T\bar{x}_S$ is a good choice for LBDA(α). Indeed, if $\alpha = T\bar{x}_S$, then LBDA(α) and x_α^* perform similar to \bar{x}_S . For example, on the small instances, they achieve optimality gaps that are on average within 0.2% and 1.1% of \bar{x}_S , respectively. However, since \bar{x}_S is difficult to compute in practice, the use of LBDA($T\bar{x}_S$) is limited, whereas LBDA(α^+) and LBDA($\alpha^\#$) can be applied directly. Similar as for the instances in Sects. 5.2.1 and 5.2.3, they achieve significantly better results than LBDA(0), at the expense of higher computation times. In particular, on the large instances, LBDA($\alpha^\#$) performs 2 to 5 times as well as LBDA(0), and on the small instances, LBDA(α^+) achieves optimality gaps that are within 0.6% of \bar{x}_S for $\sigma \geq 0.5$. While the computation times of LBDA(α) increase for $\alpha = \alpha^+$ and $\alpha = \alpha^\#$ compared to $\alpha = 0$, they remain manageable: even the large instances are solved within 26 minutes.

Finally, observe from Table 4 that LBDA(α) generally achieves better or similar performance as x_α^* . In other words, the fact that LBDA(α) uses loose optimality cuts to solve the generalized α -approximations has no negative effect on the solution quality.

6 Conclusion

We consider two-stage mixed-integer recourse models with random right-hand side. Due to non-convexity of the recourse function, such models are extremely difficult to solve. We develop a tractable approximating model by using convex approximations of the recourse function. In particular, we propose a new class of convex approximations, the so-called generalized α -approximations, and we derive a corresponding error bound on the difference between these approximations and the true recourse function. In addition, we show that this error bound is small if the variability of the random parameters in the model is large. More precisely, the error bound for the generalized α -approximations goes to zero as the total variations of the one-dimensional conditional probability density functions of the random right-hand side vector in the model go to zero.

The advantage of the generalized α -approximations over existing convex approximations is that it can be solved efficiently. In fact, we describe a loose Benders decomposition algorithm, LBDA(α), which efficiently solves the corresponding approximating model. The quality of the candidate solution \hat{x}_α generated by

LBDA(α) in the original model is guaranteed by Theorem 3, which states an upper bound on the optimality gap of \hat{x}_α . This performance guarantee is similar to the error bound we prove for the generalized α -approximations. Indeed, we show that the optimality gap of \hat{x}_α is small if the variability of the random parameters in the model is large.

In addition to this theoretical guarantee on the solution quality, we assess LBDA(α) empirically on a range of test instances. In particular, we consider the SIZES and SSLP instances from SIPLIB, an investment planning problem by [29], and randomly generated instances. We find that LBDA(α) performs well in terms of computation times, scalability to larger problem instances, and solution quality. In particular, LBDA(α) is able to solve larger instances than traditional sampling techniques and its computation times scale more favourably in the input size of the instances. In terms of solution quality, LBDA(α) solves the SIZES and SSLP instances to near optimality and generally performs very well on the investment planning instances. Moreover, on the randomly generated instances, LBDA(α) performs similar to traditional sampling techniques and achieves small optimality gaps if the variability of the random parameters in the model is medium to large.

One avenue for future research is to derive sharper theoretical error bounds for the generalized α -approximations. While Theorem 3 provides conditions under which our solution method performs well, the quantitative error bound cannot be computed, as it depends on an unknown and potentially large constant C . A sharp tractable error bound would be an improvement over our current results. Another avenue is the extension of our solution method to more general mixed-integer recourse models, for example by allowing for randomness in the second-stage cost coefficients q , technology matrix T , or recourse matrix W .

Acknowledgements The research of Ward Romeijnnders has been supported by Grant 451-17-034 4043 from The Netherlands Organisation for Scientific Research (NWO).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix

Proof of Theorem 2 Our proof is similar to e.g. [23, Theorem 5.1] and [21, Theorem 2]. Here, we point out the differences. In particular, we show that there exist vectors σ_k , $k = 1, \dots, K$ and a constant $R > 0$, such that

- (i) if $\omega - Tx \in \sigma_k + \Lambda^k$, then $v(\omega, x) - \hat{v}_\alpha(\omega, x)$ is zero-mean B^k -periodic, and
- (ii) $|v(\omega, x) - \hat{v}_\alpha(\omega, x)| \leq R$,

where Λ^k , $k = 1, \dots, K$, are the closed convex cones from Lemma 1. Property (i) follows from Lemmas 1 (iii) and 3 (ii), and the fact that $\psi^k(\omega - Tx) - \psi^k(\omega - \alpha)$ is zero-mean B^k -periodic.

In order to prove (ii), we use a similar argument as in [23, Lemma 3.6] and [21, Proposition 2]. It suffices to show that there exists a constant R' such that $|\hat{v}_\alpha(\omega, x) - v_{LP}(\omega, x)| \leq R'$. We can take $R' = \max_k w_k$, where w_k are the upper bounds on ψ^k from Lemma 1, $k = 1, \dots, K$. \square

Proof of Lemma 2 Our line of proof is based on [3]. For any $\nu > 0$, consider a finite set \mathcal{X}_ν such that for all $x \in \mathcal{X}$, there exists an $x' \in \mathcal{X}_\nu$ such that $\|x - x'\| \leq \nu$. Such a set \mathcal{X}_ν exist due to Assumption 1. Let $x \in \mathcal{X}$ be given and let $x' \in \mathcal{X}_\nu$ be such that $\|x - x'\| \leq \nu$. Note that

$$|\hat{Q}_\alpha(x) - \hat{Q}_\alpha^S(x)| \leq |\hat{Q}_\alpha(x) - \hat{Q}_\alpha(x')| + |\hat{Q}_\alpha(x') - \hat{Q}_\alpha^S(x')| + |\hat{Q}_\alpha^S(x') - \hat{Q}_\alpha^S(x)|. \quad (20)$$

The first and third term on the right-hand side of (20) can be bounded by noting that both \hat{Q}_α and \hat{Q}_α^S are Lipschitz continuous. Denote Lipschitz constants of \hat{Q}_α and \hat{Q}_α^S by L_1 and L_2 , respectively. We obtain

$$|\hat{Q}_\alpha(x) - \hat{Q}_\alpha^S(x)| \leq (L_1 + L_2)\nu + |\hat{Q}_\alpha(x') - \hat{Q}_\alpha^S(x')|,$$

which gives

$$\sup_{x \in \mathcal{X}} |\hat{Q}_\alpha(x) - \hat{Q}_\alpha^S(x)| \leq (L_1 + L_2)\nu + \sup_{x' \in \mathcal{X}_\nu} |\hat{Q}_\alpha(x') - \hat{Q}_\alpha^S(x')|.$$

The first term $(L_1 + L_2)\nu$ can be made arbitrarily small by letting $\nu \rightarrow 0$. The result follows, because for fixed ν , the second term $\sup_{x' \in \mathcal{X}_\nu} |\hat{Q}_\alpha(x') - \hat{Q}_\alpha^S(x')|$ goes to zero w.p. 1 as $S \rightarrow \infty$. To see this, fix any $x' \in \mathcal{X}_\nu$, and consider the random variable

$$\xi := \max_{k=1, \dots, K} \{\lambda^k(\omega^S - Tx') + \psi^k(\omega^S - \alpha)\}.$$

Thus, by the SLLN, $\hat{Q}_\alpha^S(x') \rightarrow \hat{Q}_\alpha(x')$ w.p. 1 as $S \rightarrow \infty$. We can apply the SLLN since $\mathbb{E}[\xi]$ exists and is finite by Assumptions (A2)–(A4). The result follows because \mathcal{X}_ν is finite. \square

Proof of Lemma 3 It follows from the definitions of \hat{v}_α and v_α that $\hat{v}_\alpha \geq v_\alpha$. Moreover, we can take $R = \max_k w_k$, where w_k are the upper bounds on ψ^k , $k = 1, \dots, K$, from Lemma 1.

To prove (ii), note that by the Basis Decomposition Theorem in [36], there exist basis matrices B^k , $k = 1, \dots, K$, and closed convex cones $\Lambda^k = \{t : (B^k)^{-1}t \geq 0\}$ such that $\omega - Tx \in \Lambda^k$ implies $k(\omega, x) = k$, and thus

$$v_\alpha(\omega, x) = \lambda^k(\omega - Tx) + \psi^k(\omega - \alpha). \quad (21)$$

It remains to show that there exists $\sigma_k \in \Lambda^k$ such that $\hat{v}_\alpha(\omega, x) = \lambda^k(\omega - Tx) + \psi^k(\omega - \alpha)$ if $\omega - Tx \in \sigma_k + \Lambda^k$. Fix arbitrary $l \in \{1, \dots, K\}$. It suffices to show that there exist $\sigma_{kl} \in \Lambda^k$ such that $\omega - Tx \in \sigma_{kl} + \Lambda^k$ implies that

$$\lambda^k(\omega - Tx) + \psi^k(\omega - \alpha) \geq \lambda^l(\omega - Tx) + \psi^l(\omega - \alpha). \quad (22)$$

This is because

$$\bigcap_{l=1}^K (\sigma_{kl} + \Lambda^k) = \sigma_k + \Lambda^k$$

for some $\sigma_k \in \Lambda^k$. Hence, if $\omega - Tx \in \sigma_k + \Lambda^k$, then

$$\hat{v}_\alpha(\omega, x) = \max_{k=1, \dots, K} \left\{ \lambda^k(\omega - Tx) + \psi^k(\omega - \alpha) \right\} = \lambda^k(\omega - Tx) + \psi^k(\omega - \alpha).$$

To prove (22), note that if $\lambda^k = \lambda^l$, then by Lemma 1 (iii), $\psi^k(\omega - \alpha) = \psi^l(\omega - \alpha)$, so that (22) holds with equality. If $\lambda^k \neq \lambda^l$, then $\lambda^k s > \lambda^l s$ for any $s \in \text{int}(\Lambda^k)$. For sufficiently large $\gamma > 0$, we thus have

$$\gamma(\lambda^k s - \lambda^l s) \geq w_l.$$

If we take $\sigma_{kl} = \gamma s$, then (22) holds by observing that $\psi^k(\omega - \alpha) \geq 0$ and $\psi^l(\omega - \alpha) \leq w_l$. \square

References

1. Ahmed, S.: A scenario decomposition algorithm for 0–1 stochastic programs. *Oper. Res. Lett.* **41**(6), 565–569 (2013)
2. Ahmed, S., Garcia, R., Kong, N., Ntamo, L., Parija, G., Qiu, F., Sen, S.: SIPLIB: a stochastic integer programming test problem library. <https://www2.isye.gatech.edu/~sahmed/siplib> (2015)
3. Ahmed, S., Shapiro, A.: The sample average approximation method for stochastic programs with integer recourse. Preprint available from <https://www.optimization-online.org> (2002)
4. Bansal, M., Huang, K.L., Mehrotra, S.: Tight second stage formulations in two-stage stochastic mixed integer programs. *SIAM J. Optim.* **28**(1), 788–819 (2018)
5. Bayraksan, G., Morton, D.P.: Assessing solution quality in stochastic programs. *Math. Program.* **108**(2–3), 495–514 (2006)
6. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.* **4**(1), 238–252 (1962)
7. Birge, J.R., Louveaux, F.V.: A multicut algorithm for two-stage stochastic linear programs. *Eur. J. Oper. Res.* **34**(3), 384–392 (1988)
8. Carøe, C.C., Schultz, R.: Dual decomposition in stochastic integer programming. *Oper. Res. Lett.* **24**(1–2), 37–45 (1999)
9. Gade, D., Küçükyavuz, S., Sen, S.: Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. *Math. Program.* **144**(1–2), 39–64 (2014)
10. Gassmann, H.I., Ziemba, W.T. (eds.): *Stochastic programming: applications in finance, energy, planning and logistics*, World Scientific Series in Finance, vol. 4. World Scientific (2013)
11. Gomory, R.: Some polyhedra related to combinatorial problems. *Linear Algebra Appl.* **2**(4), 451–558 (1969)

12. Guo, G., Hackebeil, G., Ryan, S.M., Watson, J.P., Woodruff, D.L.: Integration of progressive hedging and dual decomposition in stochastic integer programs. *Oper. Res. Lett.* **43**(3), 311–316 (2015)
13. Jorjani, S., Scott, C.H., Woodruff, D.L.: Selection of an optimal subset of sizes. *Int. J. Prod. Res.* **37**(16), 3697–3710 (1999)
14. Klein Haneveld, W.K., Stougie, L., van der Vlerk, M.H.: Simple integer recourse models: convexity and convex approximations. *Math. Program.* **108**(2–3), 435–473 (2006)
15. Laporte, G., Louveaux, F.V.: The integer L-shaped method for stochastic integer programs with complete recourse. *Oper. Res. Lett.* **13**(3), 133–142 (1993)
16. Mak, W.K., Morton, D.P., Wood, R.K.: Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Oper. Res. Lett.* **24**(1–2), 47–56 (1999)
17. Ntairo, L.: Fenchel decomposition for stochastic mixed-integer programming. *J. Glob. Optim.* **55**(1), 141–163 (2013)
18. Ntairo, L., Sen, S.: The million-variable march for stochastic combinatorial optimization. *J. Glob. Optim.* **32**(3), 385–400 (2005)
19. Qi, Y., Sen, S.: The ancestral benders' cutting plane algorithm with multi-term disjunctions for mixed-integer recourse decisions in stochastic programming. *Math. Program.* **161**(1–2), 193–235 (2017)
20. Rinnooy Kan, A.H.G., Stougie, L.: Stochastic integer programming. In: Y. Ermoliev, R.J.B. Wets (eds.) *Numerical Techniques for Stochastic Optimization, Springer Series in Computational Mathematics*, vol. 10, pp. 201–213. Springer (1988)
21. Romeijnnders, W., van der Laan, N.: Pseudo-valid cutting planes for two-stage mixed-integer stochastic programs with right-hand-side uncertainty. *Operations Research* (2020)
22. Romeijnnders, W., Morton, D.P., van der Vlerk, M.H.: Assessing the quality of convex approximations for two-stage totally unimodular integer recourse models. *INFORMS J. Comput.* **29**(2), 211–231 (2017)
23. Romeijnnders, W., Schultz, R., van der Vlerk, M.H., Klein Haneveld, W.K.: A convex approximation for two-stage mixed-integer recourse models with a uniform error bound. *SIAM J. Optim.* **26**(1), 426–447 (2016)
24. Romeijnnders, W., Stougie, L., van der Vlerk, M.H.: Approximation in two-stage stochastic integer programming. *Surv. Oper. Res. Manag. Sci.* **19**(1), 17–33 (2014)
25. Romeijnnders, W., van der Vlerk, M.H., Klein Haneveld, W.K.: Convex approximations of totally unimodular integer recourse models: a uniform error bound. *SIAM J. Optim.* **25**(1), 130–158 (2015)
26. Romeijnnders, W., van der Vlerk, M.H., Klein Haneveld, W.K.: Total variation bounds on the expectation of periodic functions with applications to recourse approximations. *Math. Program.* **157**(1), 3–46 (2016)
27. Ruszczyński, A.: A regularized decomposition method for minimizing a sum of polyhedral functions. *Math. Program.* **35**(3), 309–333 (1986)
28. Schultz, R.: Stochastic programming with integer variables. *Math. Program.* **97**(1–2), 285–309 (2003)
29. Schultz, R., Stougie, L., van der Vlerk, M.H.: Solving stochastic programs with integer recourse by enumeration: a framework using Gröbner basis reductions. *Math. Program.* **83**(1–3), 229–252 (1998)
30. Sen, S.: Algorithms for stochastic mixed-integer programming models. *Handb. Oper. Res. Manag. Sci.* **12**, 515–558 (2005)
31. Sen, S., Hgle, J.L.: The C^3 theorem and a D^2 algorithm for large scale stochastic mixed-integer programming: set convexification. *Math. Program.* **104**(1), 1–20 (2005)
32. Van Slyke, R.M., Wets, R.: L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM J. Appl. Math.* **17**(4), 638–663 (1969)
33. van der Vlerk, M.H.: Stochastic programming with integer recourse. Ph.D. thesis, University of Groningen, The Netherlands (1995)
34. van der Vlerk, M.H.: Convex approximations for complete integer recourse models. *Math. Program.* **99**(2), 297–310 (2004)
35. Van der Vlerk, M.H.: Convex approximations for a class of mixed-integer recourse models. *Ann. Oper. Res.* **177**(1), 139–150 (2010)
36. Walkup, D.W., Wets, R.J.B.: Lifting projections of convex polyhedra. *Pac. J. Math.* **28**(2), 465–475 (1969)

37. Wallace, S.W., Ziemba, W.T. (eds.): Applications of Stochastic Programming. MPS-SIAM Series on Optimization (2005)
38. Zhang, M., Küçükyavuz, S.: Finitely convergent decomposition algorithms for two-stage stochastic pure integer programs. *SIAM J. Optim.* **24**(4), 1933–1951 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.