A Primal-Dual Lifting Scheme for Two-Stage Robust Optimization

Angelos Georghiou[1], Angelos Tsoukalas[2], and Wolfram Wiesemann[3]

[1] *Desautels Faculty of Management, McGill University, Canada*

[2] *Olayan School of Business, American University of Beirut, Lebanon*

[3] *Imperial College Business School, Imperial College London, United Kingdom*

April 8, 2019

Abstract

Two-stage robust optimization problems, in which decisions are taken both in anticipation of and in response to the observation of an unknown parameter vector from within an uncertainty set, are notoriously challenging. In this paper, we develop convergent hierarchies of primal (conservative) and dual (progressive) bounds for these problems that trade off the competing goals of tractability and optimality: While the coarsest bounds recover a tractable but suboptimal affine decision rule approximation of the two-stage robust optimization problem, the refined bounds lift extreme points of the uncertainty set until an exact but intractable extreme point reformulation of the problem is obtained. Based on these bounds, we propose a primal-dual lifting scheme for the solution of two-stage robust optimization problems that accommodates for discrete here-and-now decisions, infeasible problem instances as well as the absence of a relatively complete recourse. The incumbent solutions in each step of our algorithm afford rigorous error bounds, and they can be interpreted as piecewise affine decision rules. We illustrate the performance of our algorithm on illustrative examples and on an inventory management problem.

Keywords: robust optimization; two-stage problems; decision rules; error bounds.

1 Introduction

In the last two decades, robust optimization has emerged as a powerful methodology for immunizing mathematical programs against uncertainty in the problem data. Many dynamic optimization problems can be naturally formulated as *two-stage robust optimization problems* of the form

$$\begin{array}{lll} \text{minimize} & \boldsymbol{q}^\top \boldsymbol{x} & \\ \text{subject to} & \boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x} + \boldsymbol{W} \boldsymbol{y}(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi}) & \forall \boldsymbol{\xi} \in \Xi \\ & \boldsymbol{x} \in \mathbb{R}^{n_1}, \;\; \boldsymbol{y} : \Xi \mapsto \mathbb{R}^{n_2}, & \end{array} \tag{$\mathcal{P}$}$$

where $\boldsymbol{q} \in \mathbb{R}^{n_1}$ and $\boldsymbol{W} \in \mathbb{R}^{m \times n_2}$, while $\boldsymbol{T} : \Xi \overset{A}{\mapsto} \mathbb{R}^{m \times n_1}$ and $\boldsymbol{h} : \Xi \overset{A}{\mapsto} \mathbb{R}^m$ are affine functions of the uncertain parameter vector $\boldsymbol{\xi}$, which is only known to reside in the nonempty and bounded uncertainty set $\Xi = \{\boldsymbol{\xi} \in \mathbb{R}^k \, : \, \boldsymbol{F}\boldsymbol{\xi} \leq \boldsymbol{g}\}$, $\boldsymbol{F} \in \mathbb{R}^{l \times k}$ and $\boldsymbol{g} \in \mathbb{R}^l$. Problem $\mathcal{P}$ determines a first-stage decision $\boldsymbol{x}$, which does not depend on the realization of $\boldsymbol{\xi}$, as well as a second-stage policy $\boldsymbol{y}(\boldsymbol{\xi})$, which can adapt to the realization of $\boldsymbol{\xi}$, that are immunized against all parameter realizations $\boldsymbol{\xi} \in \Xi$. Without loss of generality, we may assume that the objective function only involves the first-stage decision $\boldsymbol{x}$ as we can move the second-stage cost to the constraints via an epigraph reformulation.

The two-stage robust optimization problem $\mathcal{P}$ has been used in diverse application domains, ranging from network design and operations, such as network flow problems [5] and vehicle routing [35, 54], to railway shunting and timetabling [45], energy systems [46, 47, 50, 57], the strategic [1] and operative [4, 44] aspects of operations management as well as healthcare [42]. It is also frequently used to determine approximately optimal solutions to more generic (but at the same time computationally more challenging) multi-stage robust optimization problems [26]. For a detailed review of the applications of problem $\mathcal{P}$, we refer the reader to [10, 27, 60].

The two-stage robust optimization problem $\mathcal{P}$ is convex, but it contains infinitely many decision variables and constraints. In fact, problem $\mathcal{P}$ is NP-hard [36], and the instances that can be solved in polynomial time are both rare and restrictive [2, 18, 35]. As a result, much of the existing research has focused on developing tractable conservative approximations to problem $\mathcal{P}$. Early attempts in this direction have proposed to restrict the second-stage decision $\boldsymbol{y}$ to affine [36, 43], segregated affine [24, 25, 32], piecewise affine [31] and algebraic as well as trigonometric polynomial functions [19] (so-called *decision rules*) of the parameters $\boldsymbol{\xi}$. Decision rules have recently been extended to incorporate both continuous and discrete second-stage decisions, either by partitioning the uncertainty set Ξ into hyperrectangles [34, 56] or by resorting to a semi-infinite solution scheme [14]. By themselves, decision rule approximations only provide a conservative bound on the optimal value of the two-stage robust optimization problem $\mathcal{P}$. To estimate the incurred suboptimality, decision rules are often combined with progressive bounds that emerge from replacing the uncertainty set Ξ in $\mathcal{P}$ with a finite subset of the parameter realizations. Scenario subsets that lead to good progressive bounds can be obtained from the Lagrange multipliers associated with the optimal solution of the decision rule problem [12, 37]. The decision rule approaches naturally extend to robust optimization problems with more than two stages. The suboptimality of decision rules has been investigated in [15, 18, 43]. For a survey of the decision rule literature, see [26].

Instead of relying on decision rules, the two-stage robust optimization problem $\mathcal{P}$ can also be conservatively approximated by its K-adaptability problem or via its copositive reformulation. The K-adaptability problem selects K candidate second-stage decisions here-and-now (that is, before observing the realization of $\boldsymbol{\xi}$) and implements the best of these decisions after the realization of $\boldsymbol{\xi}$ is known. Different K-adaptability solution schemes have been proposed in [11, 39, 53], and their suboptimality has been analyzed in [17]. The copositive reformulation of problem $\mathcal{P}$, on the other hand, is exact and convex, but it is typically difficult to solve. Tractable conservative approximations to this reformulation can be obtained via semidefinite programming [38, 59].

All of the solution schemes reviewed so far have in common that they provide a conservative approximation to the two-stage robust optimization problem $\mathcal{P}$. Problem $\mathcal{P}$ can be solved *exactly* through an iterative approximation of its worst-case second-stage cost function or its uncertainty set. An iterative approximation of the second-stage cost function can be obtained through a variant of Benders' decomposition [20, 41, 55, 62]. To this end, problem $\mathcal{P}$ is decomposed into a convex master problem involving the first-stage decisions and an outer (progressive) approximation of the worst-case second-stage cost, as well as a non-convex subproblem that provides cuts for the cost function. The Benders' decomposition scheme has been extended to the multi-stage version of problem $\mathcal{P}$ in [30]. Alternatively, the papers [6] and [61] propose a column-and-constraint generation scheme, which is based on semi-infinite programming techniques, to iteratively approximate the uncertainty set in problem $\mathcal{P}$. Here, the convex master problem is a relaxation of problem $\mathcal{P}$ that involves finitely many realizations $\boldsymbol{\xi} \in \Xi$, and the non-convex subproblem identifies parameter realizations $\boldsymbol{\xi} \in \Xi$ to be added to the master problem. Both the Benders' decomposition approaches and the semi-infinite programming schemes rely on sequences of progressive approximations to determine an optimal solution to problem $\mathcal{P}$ in finite time. In contrast, the iterative solution schemes presented in [13, 49] extend the uncertainty set partitioning approaches of [34, 56] to construct a sequence of conservative approximations to problem $\mathcal{P}$. Here, the convex master problem determines constant or affine decision rules for each set of the partition, and the subproblem identifies a refined partition for the master problem. Both approaches extend to integer decisions and more than two stages. It has been shown, however, that even asymptotic convergence to an optimal solution of problem $\mathcal{P}$ cannot be guaranteed in general [13].

Instead of approximating the second-stage decisions, the second-stage cost function or the uncertainty set of the two-stage robust optimization problem $\mathcal{P}$, it has been proposed in [63] to solve $\mathcal{P}$ through an iterative reformulation of the problem itself. The authors use Fourier-Motzkin elimination to reduce the number of second-stage decisions in problem $\mathcal{P}$ at the expense of additional constraints. This results in a hierarchy of increasingly accurate conservative approximations of $\mathcal{P}$ which converge to a static robust optimization problem that is equivalent to $\mathcal{P}$.

In this paper, we develop an alternative solution scheme for the two-stage robust optimization problem $\mathcal{P}$ that aims to provide an attractive trade-off between the conflicting objectives of optimality and tractability. We summarize our key contributions as follows.

1. We develop convergent hierarchies of primal (conservative) and dual (progressive) bounds to the two-stage robust optimization problem $\mathcal{P}$. Our bounds combine affine decision rules with an extreme point enumeration to trade off the conflicting goals of tractability and optimality. While the primal bounds apply to any bounded polyhedral uncertainty set, the dual bounds require information about the sum of outer products of the extreme points of the uncertainty set, which can be computed in closed form for several classes of common uncertainty sets.

2. We propose a primal-dual lifting scheme that is inspired by polyhedral combinatorics. Our solution approach accommodates for discrete here-and-now decisions, infeasible problem instances as well as the absence of a relatively complete recourse. The initial bounds are based on affine decision rules and can thus be computed efficiently.

3. We highlight the intimate relationship between our bounds and piecewise affine decision rules over simplicial decompositions of the uncertainty set Ξ in problem $\mathcal{P}$.

We believe that the proposed approach fills a gap in the literature: Both the Benders' decomposition schemes and the semi-infinite programming approaches reviewed above typically consider instances of problem $\mathcal{P}$ with right-hand uncertainty and a relatively complete recourse. Moreover, while the master problems in these approaches provide tractable progressive bounds, the subproblems providing the conservative bounds constitute bi-affine or mixed-integer optimization problems that are difficult to solve. Although the adaptive uncertainty partitioning approaches can provide tractable conservative and progressive bounds, they do not offer a convergence guarantee. To our best knowledge, the Fourier-Motzkin elimination scheme presented in [63] is the only approach that satisfies all of the properties outlined above. We will show in our numerical experiments that although both the Fourier-Motzkin elimination scheme and our approach are outperformed by the column-and-constraint generation scheme [61] on instances of an inventory management problem with relatively complete recourse, our approach outperforms both alternatives on a variant of the problem that lacks a relatively complete recourse. We thus believe that our lifting approach provides a promising complementary method to solve two-stage robust optimization problems.

The remainder of the paper is structured as follows. Section 2 presents our hierarchies of primal and dual bounds to problem $\mathcal{P}$, and Section 3 employs these bounds to develop a primal-dual lifting scheme for the solution of problem $\mathcal{P}$. We discuss the relationship between our bounds and piecewise affine decision rules in Section 4, and we report on numerical experiments in Section 5. The appendix shows how to compute the sum of outer products of the extreme points of an uncertainty set, which is required for our dual bound, for several classes of common uncertainty sets.

Remark 1 (Discrete Here-and-Now Decisions)**.** *Throughout the paper we will discuss how our algorithm extends to a variant of problem $\mathcal{P}$ that contains continuous and discrete first-stage decisions:*

$$\begin{array}{lll} \text{minimize} & \boldsymbol{q}_{\mathrm{c}}^\top \boldsymbol{x}_{\mathrm{c}} + \boldsymbol{q}_{\mathrm{d}}^\top \boldsymbol{x}_{\mathrm{d}} & \\ \text{subject to} & \boldsymbol{T}_{\mathrm{c}}(\boldsymbol{\xi})\,\boldsymbol{x}_{\mathrm{c}} + \boldsymbol{T}_{\mathrm{d}}(\boldsymbol{\xi})\,\boldsymbol{x}_{\mathrm{d}} + \boldsymbol{W}\boldsymbol{y}(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi}) \qquad \forall \boldsymbol{\xi} \in \Xi & \\ & \boldsymbol{x}_{\mathrm{c}} \in \mathbb{R}^{nc_1}, \;\; \boldsymbol{x}_{\mathrm{d}} \in \mathbb{Z}^{nd_1}, \;\; \boldsymbol{y} : \Xi \mapsto \mathbb{R}^{n_2} & \end{array} \tag{$\mathcal{P}_{\mathrm{d}}$}$$

We will see that our bounds and the lifting scheme naturally extend to this more generic problem.

Notation. For a finite-dimensional set Ω, we denote by $\operatorname{ext}\Omega$, $\operatorname{conv}\Omega$ and $\operatorname{cl}\Omega$ the set of extreme points, the convex hull and the closure of Ω, respectively. We define by $\{f : \mathbb{R}^n \overset{A}{\mapsto} \mathbb{R}^m\}$ the set of all affine functions from $\mathbb{R}^n$ to $\mathbb{R}^m$. The vector of ones and the identity matrix are denoted by $\mathbf{e}$ and $\boldsymbol{I}$, respectively, and their dimensions will be clear from the context. We use $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \operatorname{tr}(\boldsymbol{A}^\top \boldsymbol{B})$ to denote the trace inner product between two symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

2 A Duality Scheme for Two-Stage Robust Optimization

Problem $\mathcal{P}$ is a convex but challenging optimization problem as it involves infinitely many decision variables and constraints. This can be partially resolved if we replace the uncertainty set Ξ in $\mathcal{P}$ with its extreme points $\operatorname{ext}\Xi$:

$$
\begin{array}{lll}
\text{minimize} & \boldsymbol{q}^\top \boldsymbol{x} & \\
\text{subject to} & \boldsymbol{T}(\boldsymbol{\xi})\,\boldsymbol{x} + \boldsymbol{W}\boldsymbol{y}(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi}) & \forall \boldsymbol{\xi} \in \operatorname{ext}\Xi \\
& \boldsymbol{x} \in \mathbb{R}^{n_1}, \;\; \boldsymbol{y} : \operatorname{ext}\Xi \mapsto \mathbb{R}^{n_2}, &
\end{array}
\tag{$\mathcal{P}'$}
$$

In the following, we refer to this problem as the *extreme point reformulation* $\mathcal{P}'$.

Proposition 1 (Extreme Point Reformulation)**.** *Problems $\mathcal{P}$ and $\mathcal{P}'$ are equivalent in the following sense: Any feasible solution to one problem can be transformed into a feasible solution to the other problem that attains the same objective value.*

Proof of Proposition 1. One readily verifies that for every feasible solution $\boldsymbol{x} \in \mathbb{R}^{n_1}$ and $\boldsymbol{y} : \Xi \mapsto \mathbb{R}^{n_2}$ to the robust optimization problem $\mathcal{P}$, the restriction of $\boldsymbol{y}$ to $\operatorname{ext}\Xi$ provides a feasible solution to the extreme point reformulation $\mathcal{P}'$ that attains the same objective value.

Let $\boldsymbol{x} \in \mathbb{R}^{n_1}$ and $\boldsymbol{y}' : \operatorname{ext}\Xi \mapsto \mathbb{R}^{n_2}$ be a feasible solution to the extreme point reformulation $\mathcal{P}'$. By construction, there is a function $\lambda : \Xi \times \operatorname{ext}\Xi \mapsto \mathbb{R}_+$ that satisfies

$$
\sum_{\boldsymbol{\xi}' \in \operatorname{ext}\Xi} \lambda(\boldsymbol{\xi}, \boldsymbol{\xi}') = 1 \qquad \text{and} \qquad \sum_{\boldsymbol{\xi}' \in \operatorname{ext}\Xi} \lambda(\boldsymbol{\xi}, \boldsymbol{\xi}') \cdot \boldsymbol{\xi}' = \boldsymbol{\xi} \qquad \forall \boldsymbol{\xi} \in \Xi.
$$

Hence, $(\boldsymbol{x}, \boldsymbol{y})$ with $\boldsymbol{y}(\boldsymbol{\xi}) := \sum_{\boldsymbol{\xi}' \in \operatorname{ext}\Xi} \lambda(\boldsymbol{\xi}, \boldsymbol{\xi}') \cdot \boldsymbol{y}'(\boldsymbol{\xi}')$, $\boldsymbol{\xi} \in \Xi$, is feasible in problem $\mathcal{P}$ since

$$
\begin{aligned}
\boldsymbol{T}(\boldsymbol{\xi})\,\boldsymbol{x} + \boldsymbol{W}\boldsymbol{y}(\boldsymbol{\xi}) \;&=\; \sum_{\boldsymbol{\xi}' \in \operatorname{ext}\Xi} \lambda(\boldsymbol{\xi}, \boldsymbol{\xi}') \cdot \left[\boldsymbol{T}(\boldsymbol{\xi}')\,\boldsymbol{x} + \boldsymbol{W}\boldsymbol{y}'(\boldsymbol{\xi}')\right] \\
&\geq\; \sum_{\boldsymbol{\xi}' \in \operatorname{ext}\Xi} \lambda(\boldsymbol{\xi}, \boldsymbol{\xi}') \cdot \boldsymbol{h}(\boldsymbol{\xi}') \;=\; \boldsymbol{h}(\boldsymbol{\xi}) \qquad \forall \boldsymbol{\xi} \in \Xi,
\end{aligned}
$$

where the first identity follows from the definition of $(\boldsymbol{x}, \boldsymbol{y})$ and the fact that the mapping $\boldsymbol{T}$ is affine, the inequality follows from the feasibility of $(\boldsymbol{x}, \boldsymbol{y}')$ in the extreme point reformulation $\mathcal{P}'$, and the second identity holds since $\sum_{\boldsymbol{\xi}' \in \operatorname{ext}\Xi} \lambda(\boldsymbol{\xi}, \boldsymbol{\xi}') \cdot \boldsymbol{\xi}' = \boldsymbol{\xi}$. The statement now follows from the fact that $(\boldsymbol{x}, \boldsymbol{y})$ attains the same objective value in $\mathcal{P}$ as $(\boldsymbol{x}, \boldsymbol{y}')$ does in $\mathcal{P}'$. $\square$

Note that the proof of Proposition 1 does not rely on the characteristics of the first-stage decision $\boldsymbol{x}$. Hence, the statement immediately extends to the generalized two-stage robust optimization problem $\mathcal{P}_{\mathrm{d}}$ from Remark 1 that contains both continuous and discrete here-and-now decisions.

The extreme point reformulation $\mathcal{P}'$ is finite-dimensional, but it remains computationally burdensome since the number of decision variables and constraints scales with the number of extreme points of Ξ. By the dual upper bound theorem [51, § 26.3], this number satisfies

$$|\mathrm{ext}\,\Xi| \;\leq\; \binom{l - \lceil k/2 \rceil}{\lfloor k/2 \rfloor} + \binom{l - 1 - \lceil (k-1)/2 \rceil}{\lfloor (k-1)/2 \rfloor},$$

and the bound is attained by dual cyclic polytopes. This superpolynomial growth is to be expected as problem $\mathcal{P}$ is NP-hard [36, Theorem 3.4]. We note that for a fixed number k of uncertain problem parameters, the dual upper bound theorem implies that $|\mathrm{ext}\,\Xi| \in \mathcal{O}(l^{\lfloor k/2 \rfloor})$. Moreover, the *expected* number of vertices of uniformly sampled polyhedra is much lower than this upper bound [22].

We emphasize that the statement of Proposition 1 does not hold if problem $\mathcal{P}$ exhibits random recourse, that is, if $\boldsymbol{W}$ in $\mathcal{P}$ depends on $\boldsymbol{\xi}$. This has also been observed in [6].

Example 1 (**Random Recourse**)**.** *Consider the following instance of $\mathcal{P}$ with random recourse:*

$$\begin{array}{lll} \text{minimize} & 0 & \\ \text{subject to} & \xi \cdot y(\xi) \geq 1 & \forall \xi \in [-1, 1] \\ & y : [-1, 1] \mapsto \mathbb{R} & \end{array}$$

This problem is infeasible since any second-stage decision y violates the constraint $0 \cdot y(0) \geq 1$, but its extreme point formulation is solved by any y satisfying $y(-1) \leq -1$ and $y(1) \geq 1$. □

The next two subsections develop families of primal and dual bounds on the optimal value of the two-stage robust optimization problem $\mathcal{P}$ that combine the extreme point reformulation $\mathcal{P}'$ with affine decision rules.

2.1 Hierarchy of Primal Bounds

A popular conservative approximation to the two-stage robust optimization problem $\mathcal{P}$ replaces the second-stage decision $\boldsymbol{y} : \Xi \mapsto \mathbb{R}^{n_2}$ with an affine decision rule $\boldsymbol{y} : \Xi \overset{A}{\mapsto} \mathbb{R}^{n_2}$. The resulting problem has finitely many decision variables (the intercept and the slopes of $\boldsymbol{y}$) but infinitely many constraints. Classical robust optimization techniques then employ linear programming duality to reformulate this semi-infinite problem as a linear program that scales polynomially in the size (n_1, n_2, m, k, l) of the input data [36]. The affine decision rule approximation performs surprisingly well on practical problems [43], and it is even optimal in some problem classes [2, 18, 35]. In general, however, affine decision rules are suboptimal even in seemingly benign problems [31, Example 4.5].

In the following we present a family of conservative approximations to $\mathcal{P}$ that includes the highly tractable but usually suboptimal affine decision rules and the optimal but typically intractable extreme point reformulation $\mathcal{P}'$ as special cases. We parameterize our approximation by the *scenario set* $\Xi_S \subseteq \operatorname{ext}\Xi$, which gives rise to a complementary *affine set* $\Xi_A \subseteq \Xi$ defined by $\Xi_A = \operatorname{conv}([\operatorname{ext}\Xi] \setminus \Xi_S)$:

$$
\begin{array}{lll}
\text{minimize} & \boldsymbol{q}^\top \boldsymbol{x} & \\
\text{subject to} & \boldsymbol{T}(\boldsymbol{\xi})\,\boldsymbol{x} + \boldsymbol{W}\boldsymbol{y}_S(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi}) & \forall \boldsymbol{\xi} \in \Xi_S \\
& \boldsymbol{T}(\boldsymbol{\xi})\,\boldsymbol{x} + \boldsymbol{W}\boldsymbol{y}_A(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi}) & \forall \boldsymbol{\xi} \in \Xi_A \\
& \boldsymbol{x} \in \mathbb{R}^{n_1}, \;\; \boldsymbol{y}_S : \Xi_S \mapsto \mathbb{R}^{n_2}, \;\; \boldsymbol{y}_A : \Xi_A \overset{A}{\mapsto} \mathbb{R}^{n_2} &
\end{array}
\qquad (\overline{\mathcal{P}}(\Xi_S))
$$

Problem $\overline{\mathcal{P}}(\Xi_S)$ optimizes over individual second-stage decisions $\boldsymbol{y}_S : \Xi_S \mapsto \mathbb{R}^{n_2}$ for all realizations $\boldsymbol{\xi}$ in the scenario set Ξ_S and over an affine decision rule $\boldsymbol{y}_A : \Xi_A \overset{A}{\mapsto} \mathbb{R}^{n_2}$ for the affine set Ξ_A. The scenario set Ξ_S is a finite subset of the extreme points of Ξ, whereas the affine set Ξ_A is a polyhedral subset of Ξ. Robust optimization techniques allow us to reformulate problem $\overline{\mathcal{P}}(\Xi_S)$ as a linear program that scales polynomially in the parameters $(n_1, n_2, m, |\Xi_S|, k, \ell)$, where ℓ denotes the number of inequalities required to describe Ξ_A (see page 9 below). The choices $\Xi_S = \emptyset$ (and thus $\Xi_A = \Xi$) and $\Xi_S = \operatorname{ext}\Xi$ (and thus $\Xi_A = \emptyset$) recover the affine decision rule approximation and the extreme point reformulation $\mathcal{P}'$, respectively. The requirement that $\Xi_A = \operatorname{conv}([\operatorname{ext}\Xi] \setminus \Xi_S)$ is illustrated in Figure 1, and it ensures that

$$
\operatorname{conv}(\Xi_S \cup \Xi_A) \;=\; \operatorname{conv}(\Xi_S \cup \operatorname{conv}([\operatorname{ext}\Xi] \setminus \Xi_S)) \;=\; \operatorname{conv}(\Xi_S \cup ([\operatorname{ext}\Xi] \setminus \Xi_S)) \;=\; \operatorname{conv}(\operatorname{ext}\Xi) \;=\; \Xi.
$$

In the following, we use $\overline{\mathcal{P}}(\Xi_S)$ both to refer to the bounding problem and to its optimal objective value. We now show that $\overline{\mathcal{P}}(\Xi_S)$ bounds problem $\mathcal{P}$ from above.

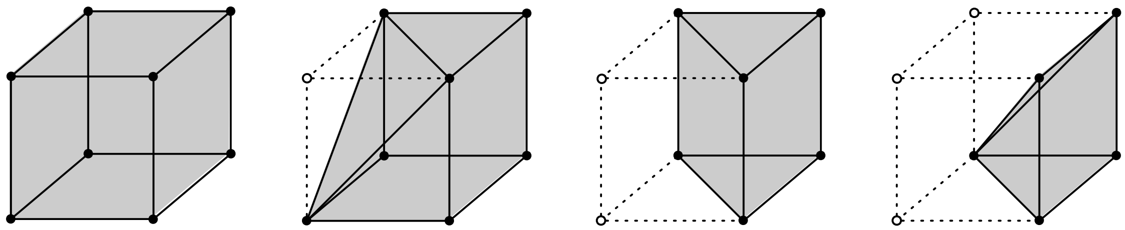

Figure 1. Different decompositions of a three-dimensional hypercube into scenario sets Ξ_S (hollow circles) and implied affine sets Ξ_A (shaded regions). In all cases, the convex hull of Ξ_S and Ξ_A recovers the original hypercube.

Proposition 2 (Primal Bound)**.** *Problem $\overline{\mathcal{P}}(\Xi_S)$ satisfies the following two properties.*

(i) Any feasible solution to $\overline{\mathcal{P}}(\Xi_S)$ can be transformed into a feasible solution to $\overline{\mathcal{P}}(\Xi'_S)$, $\Xi_S \subseteq \Xi'_S$, that attains the same objective value.

(ii) Any feasible solution to $\overline{\mathcal{P}}(\operatorname{ext}\Xi)$ can be transformed into a feasible solution to $\mathcal{P}$ that attains the same objective value, and vice versa.

Proof of Proposition 2. As for the first statement, let $(\boldsymbol{x}, \boldsymbol{y}_S, \boldsymbol{y}_A)$ be a feasible solution to $\overline{\mathcal{P}}(\Xi_S)$. In that case, $(\boldsymbol{x}, \boldsymbol{y}'_S, \boldsymbol{y}'_A)$ with $\boldsymbol{y}'_S(\boldsymbol{\xi}) := \boldsymbol{y}_S(\boldsymbol{\xi})$ for $\boldsymbol{\xi} \in \Xi_S$; $:= \boldsymbol{y}_A(\boldsymbol{\xi})$ for $\boldsymbol{\xi} \in \Xi'_S \setminus \Xi_S$ and $\boldsymbol{y}'_A(\boldsymbol{\xi}) := \boldsymbol{y}_A(\boldsymbol{\xi})$, $\boldsymbol{\xi} \in \Xi'_A$, is a feasible solution to $\mathcal{P}(\Xi'_S)$ that attains the same objective value.

The second statement directly follows from Proposition 1. $\square$

The second property states that $\overline{\mathcal{P}}(\operatorname{ext}\Xi)$ is equivalent to problem $\mathcal{P}$. The first property implies that $\overline{\mathcal{P}}(\Xi_S)$ provides a conservative approximation to $\overline{\mathcal{P}}(\Xi'_S)$ if $\Xi_S \subseteq \Xi'_S$, and in particular—by virtue of the second statement—to problem $\mathcal{P}$. We remark that the second statement also holds true for $\Xi_S \neq \operatorname{ext}\Xi$ if the implied affine set Ξ_A is a simplex [16, Theorem 1].

The proof of Proposition 2 does not exploit any properties of the first-stage decision $\boldsymbol{x}$, and hence the result immediately extends to the mixed-integer two-stage robust optimization problem $\mathcal{P}_{\mathrm{d}}$.

The size of problem $\overline{\mathcal{P}}(\Xi_S)$ depends on the number of scenarios $|\Xi_S|$ and the number of inequalities ℓ required to describe Ξ_A, both of which depend on the choice of the uncertainty set Ξ. If Ξ is the 1-norm ball in $\mathbb{R}^k$, $\Xi = \{\boldsymbol{\xi} \in \mathbb{R}^k \, : \, \|\boldsymbol{\xi}\|_1 \leq 1\}$, for example, then Ξ_S contains at most $|\operatorname{ext}\Xi| = 2k$ scenarios, and any set Ξ_A can be described efficiently by (subsets of) the projection of $\Xi' = \{(\boldsymbol{\xi}, \boldsymbol{\chi}) \in \mathbb{R}^k \times \mathbb{R}^k \, : \, \mathbf{e}^\top \boldsymbol{\chi} \leq 1, \;\; \boldsymbol{\chi} \geq \pm\boldsymbol{\xi}\}$ onto its first k components. If the uncertainty set is the ∞-norm ball in $\mathbb{R}^k$, $\Xi = \{\boldsymbol{\xi} \in \mathbb{R}^k \, : \, \|\boldsymbol{\xi}\|_\infty \leq 1\}$, on the other hand, then Ξ_S can contain up to

$|\mathrm{ext}\,\Xi| = 2^k$ scenarios, and the sets Ξ_A form the class of 0/1 polytopes which can have exponentially many facets [64]. The hope is that in practice, instances of $\overline{\mathcal{P}}(\Xi_S)$ with compact descriptions of Ξ_A and Ξ_S produce tight bounds on the optimal value of problem $\mathcal{P}$. Section 3 presents an iterative procedure that aims to determine such instances, and Section 5 investigates the performance of this procedure in an inventory management problem.

2.2 Hierarchy of Dual Bounds

The paper [37] proposes a progressive bound for the two-stage robust optimization problem $\mathcal{P}$ that replaces the uncertainty set Ξ with a subset of its extreme points $\mathrm{ext}\,\Xi$. This bound has found widespread use in both two-stage [6, 61] and multi-stage [13, 14, 49] robust optimization. In this section we derive a family of dual bounds that tighten the bound of [37]. Our bounds can be interpreted as an application of the primal approximation from the previous section to a dual extreme point formulation of $\mathcal{P}$. To this end, we consider the linear programming dual of the extreme point reformulation $\mathcal{P}'$:

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{\boldsymbol{\xi} \in \mathrm{ext}\,\Xi} \boldsymbol{h}(\boldsymbol{\xi})^\top \boldsymbol{\lambda}(\boldsymbol{\xi}) & \\ \text{subject to} & \displaystyle\sum_{\boldsymbol{\xi} \in \mathrm{ext}\,\Xi} \boldsymbol{T}(\boldsymbol{\xi})^\top \boldsymbol{\lambda}(\boldsymbol{\xi}) = \boldsymbol{q} & \\ & \boldsymbol{W}^\top \boldsymbol{\lambda}(\boldsymbol{\xi}) = \boldsymbol{0} & \forall \boldsymbol{\xi} \in \mathrm{ext}\,\Xi \\ & \boldsymbol{\lambda} : \mathrm{ext}\,\Xi \mapsto \mathbb{R}^m_+ & \end{array} \tag{1}$$

Contrary to the primal extreme point reformulation $\mathcal{P}'$, problem (1) is a maximization problem. Therefore, a conservative approximation to problem (1) will amount to a progressive approximation to problem $\mathcal{P}'$, and hence to problem $\mathcal{P}$. In order to conservatively approximate problem (1), we proceed along the lines of the previous section: We partition the extreme points $\mathrm{ext}\,\Xi$ into a scenario set Ξ_S, for which we select individual decisions $\boldsymbol{\lambda}_S : \Xi_S \mapsto \mathbb{R}^m_+$, and an implied affine set $\Xi_A = \mathrm{conv}\,([\mathrm{ext}\,\Xi] \setminus \Xi_S)$, for which the decisions $\boldsymbol{\lambda}_A : \Xi_A \overset{A}{\mapsto} \mathbb{R}^m_+$ are restricted to an affine function of $\boldsymbol{\xi}$. Contrary to the primal bounding problem $\overline{\mathcal{P}}(\Xi_S)$, however, the uncertain problem parameters $\boldsymbol{\xi}$ enter the dual extreme point reformulation (1) quadratically, and they thus require some additional care. One can then derive the following family of dual bounds to problem $\mathcal{P}$:

$$\begin{array}{ll} \text{maximize} & \left\langle \boldsymbol{H}^\top \boldsymbol{\Lambda}, \boldsymbol{\Sigma}_A \right\rangle + \sum_{\boldsymbol{\xi} \in \Xi_S} \boldsymbol{h}(\boldsymbol{\xi})^\top \boldsymbol{\lambda}_S(\boldsymbol{\xi}) \\ \text{subject to} & \left\langle \boldsymbol{T}_i^\top \boldsymbol{\Lambda}, \boldsymbol{\Sigma}_A \right\rangle + \sum_{\boldsymbol{\xi} \in \Xi_S} \boldsymbol{T}_i(\boldsymbol{\xi})^\top \boldsymbol{\lambda}_S(\boldsymbol{\xi}) = q_i \qquad \forall i = 1, \ldots, n_1 \\ & \boldsymbol{W}^\top \boldsymbol{\lambda}_S(\boldsymbol{\xi}) = \boldsymbol{0} \qquad \forall \boldsymbol{\xi} \in \Xi_S \\ & \boldsymbol{W}^\top \boldsymbol{\lambda}_A(\boldsymbol{\xi}) = \boldsymbol{0} \qquad \forall \boldsymbol{\xi} \in \Xi_A \\ & \boldsymbol{\lambda}_S : \Xi_S \mapsto \mathbb{R}^m_+, \ \ \boldsymbol{\lambda}_A : \Xi_A \overset{A}{\mapsto} \mathbb{R}^m_+ \end{array} \tag{$\underline{\mathcal{P}}(\Xi_S)$}$$

In this problem, we use the notational shorthands $\boldsymbol{H} = (\boldsymbol{h}_0, \boldsymbol{H}_J)$, $\boldsymbol{T}_i = (\boldsymbol{t}_{i,0}, \boldsymbol{T}_{i,J})$ and $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}_0, \boldsymbol{\Lambda}_J)$, where $\boldsymbol{h}_0, \boldsymbol{t}_{i,0}, \boldsymbol{\lambda}_0 \in \mathbb{R}^m$ denote the intercepts and $\boldsymbol{H}_J, \boldsymbol{T}_{i,J}, \boldsymbol{\Lambda}_J \in \mathbb{R}^{m \times k}$ the slopes (Jacobians) of the affine functions $\boldsymbol{h}(\boldsymbol{\xi})$, $\boldsymbol{T}_i(\boldsymbol{\xi}) = (T_{1,i}(\boldsymbol{\xi}), \ldots, T_{m,i}(\boldsymbol{\xi}))^\top$ and $\boldsymbol{\lambda}_A(\boldsymbol{\xi})$, and the *moment matrix*

$$\boldsymbol{\Sigma}_A = \sum_{\boldsymbol{\xi} \in \operatorname{ext} \Xi_A} \begin{pmatrix} 1 \\ \boldsymbol{\xi} \end{pmatrix} \begin{pmatrix} 1 \\ \boldsymbol{\xi} \end{pmatrix}^\top = \begin{pmatrix} |\operatorname{ext} \Xi_A| & \sum_{\boldsymbol{\xi} \in \operatorname{ext} \Xi_A} \boldsymbol{\xi}^\top \\ \sum_{\boldsymbol{\xi} \in \operatorname{ext} \Xi_A} \boldsymbol{\xi} & \sum_{\boldsymbol{\xi} \in \operatorname{ext} \Xi_A} \boldsymbol{\xi} \boldsymbol{\xi}^\top \end{pmatrix}$$

records the sum of the extreme point scenarios in Ξ_A, as well as their outer products. Standard robust optimization techniques allow us to reformulate problem $\underline{\mathcal{P}}(\Xi_S)$ as a linear program that scales polynomially in the parameters $(n_1, n_2, m, |\Xi_S|, k, \ell)$, where ℓ denotes the number of inequalities in the description of Ξ_A (see page 15 below). The choice $\Xi_S = \emptyset$ (and thus $\Xi_A = \Xi$) corresponds to a highly tractable but usually suboptimal dual affine decision rule formulation, whereas $\Xi_S = \operatorname{ext} \Xi$ (and thus $\Xi_A = \emptyset$) recovers the optimal but typically intractable dual extreme point reformulation (1). We use $\underline{\mathcal{P}}(\Xi_S)$ both to refer to the bounding problem and to its optimal objective value.

We now formalize our reasoning that $\underline{\mathcal{P}}(\Xi_S)$ bounds problem $\mathcal{P}$ from below.

Proposition 3 (Dual Bound)**.** *Problem $\underline{\mathcal{P}}(\Xi_S)$ satisfies the following two properties.*

(i) Any feasible solution to $\underline{\mathcal{P}}(\Xi_S)$ can be transformed into a feasible solution to $\underline{\mathcal{P}}(\Xi'_S)$, $\Xi_S \subseteq \Xi'_S$, that attains the same objective value.

(ii) $\underline{\mathcal{P}}(\operatorname{ext} \Xi)$ is infeasible if $\mathcal{P}$ is unbounded and vice versa. If both problems are feasible, then their optimal values coincide.

Proof of Proposition 3. By weak linear programming duality, the dual extreme point reformulation (1) bounds the extreme point reformulation $\mathcal{P}'$—and hence, by virtue of Proposition 1, the two-stage robust optimization problem $\mathcal{P}$—from below. In particular, (1) is infeasible if $\mathcal{P}$ is unbounded and vice versa. Moreover, strong linear programming duality implies that both problems attain the same optimal value if they are both feasible. The second statement of the proposition now follows from the fact that the dual extreme point reformulation (1) is equivalent to $\underline{\mathcal{P}}(\operatorname{ext}\Xi)$.

As for the first statement, fix a feasible solution $(\boldsymbol{\lambda}_S, \boldsymbol{\lambda}_A)$ for the problem $\underline{\mathcal{P}}(\Xi_S)$. We claim that the solution $(\boldsymbol{\lambda}'_S, \boldsymbol{\lambda}'_A)$ defined through $\boldsymbol{\lambda}'_S(\boldsymbol{\xi}) := \boldsymbol{\lambda}_S(\boldsymbol{\xi})$ for $\boldsymbol{\xi} \in \Xi_S$; $:= \boldsymbol{\lambda}_A(\boldsymbol{\xi})$ for $\boldsymbol{\xi} \in \Xi'_S \setminus \Xi_S$ and $\boldsymbol{\lambda}'_A(\boldsymbol{\xi}) := \boldsymbol{\lambda}_A(\boldsymbol{\xi})$, $\boldsymbol{\xi} \in \Xi'_A$, is feasible in $\underline{\mathcal{P}}(\Xi'_S)$ and attains the same objective value as $(\boldsymbol{\lambda}_S, \boldsymbol{\lambda}_A)$ does in $\underline{\mathcal{P}}(\Xi_S)$. Indeed, the objective function value of $(\boldsymbol{\lambda}'_S, \boldsymbol{\lambda}'_A)$ in $\underline{\mathcal{P}}(\Xi'_S)$ evaluates to

$$
\begin{aligned}
\left\langle \boldsymbol{H}^\top \boldsymbol{\Lambda}, \boldsymbol{\Sigma}'_A \right\rangle + \sum_{\boldsymbol{\xi} \in \Xi'_S} \boldsymbol{h}(\boldsymbol{\xi})^\top \boldsymbol{\lambda}'_S(\boldsymbol{\xi}) \;&=\; \left\langle \boldsymbol{H}^\top \boldsymbol{\Lambda}, \boldsymbol{\Sigma}'_A \right\rangle + \sum_{\boldsymbol{\xi} \in \Xi'_S \setminus \Xi_S} \boldsymbol{h}(\boldsymbol{\xi})^\top \boldsymbol{\lambda}'_S(\boldsymbol{\xi}) + \sum_{\boldsymbol{\xi} \in \Xi_S} \boldsymbol{h}(\boldsymbol{\xi})^\top \boldsymbol{\lambda}'_S(\boldsymbol{\xi}) \\
&=\; \left\langle \boldsymbol{H}^\top \boldsymbol{\Lambda}, \boldsymbol{\Sigma}'_A \right\rangle + \sum_{\substack{\boldsymbol{\xi} \in \operatorname{ext}\Xi_A : \\ \boldsymbol{\xi} \notin \operatorname{ext}\Xi'_A}} \boldsymbol{h}(\boldsymbol{\xi})^\top \boldsymbol{\lambda}_A(\boldsymbol{\xi}) + \sum_{\boldsymbol{\xi} \in \Xi_S} \boldsymbol{h}(\boldsymbol{\xi})^\top \boldsymbol{\lambda}_S(\boldsymbol{\xi}) \\
&=\; \left\langle \boldsymbol{H}^\top \boldsymbol{\Lambda}, \boldsymbol{\Sigma}'_A \right\rangle + \left\langle \boldsymbol{H}^\top \boldsymbol{\Lambda}, \boldsymbol{\Sigma}_A - \boldsymbol{\Sigma}'_A \right\rangle + \sum_{\boldsymbol{\xi} \in \Xi_S} \boldsymbol{h}(\boldsymbol{\xi})^\top \boldsymbol{\lambda}_S(\boldsymbol{\xi}) \\
&=\; \left\langle \boldsymbol{H}^\top \boldsymbol{\Lambda}, \boldsymbol{\Sigma}_A \right\rangle + \sum_{\boldsymbol{\xi} \in \Xi_S} \boldsymbol{h}(\boldsymbol{\xi})^\top \boldsymbol{\lambda}_S(\boldsymbol{\xi}),
\end{aligned}
$$

where $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}_0, \boldsymbol{\Lambda}_J)$ and $\boldsymbol{\Lambda}_J \in \mathbb{R}^{m \times k}$ and $\boldsymbol{\lambda}_0 \in \mathbb{R}^m$ denote the slopes and the intercept of $\boldsymbol{\lambda}_A$, respectively, while $\boldsymbol{\Sigma}_A$ and $\boldsymbol{\Sigma}'_A$ are the moment matrices corresponding to the affine sets Ξ_A and Ξ'_A. Here, the second identity follows from the definition of $\boldsymbol{\lambda}'_S$ and the fact that

$$
\operatorname{ext}\Xi_A \;=\; \operatorname{ext}\left(\operatorname{conv}\left([\operatorname{ext}\Xi] \setminus \Xi_S\right)\right) \;=\; (\operatorname{ext}\Xi) \setminus \Xi_S
$$

and hence,

$$
(\operatorname{ext}\Xi_A) \setminus (\operatorname{ext}\Xi'_A) \;=\; ([\operatorname{ext}\Xi] \setminus \Xi_S) \setminus ([\operatorname{ext}\Xi] \setminus \Xi'_S) \;=\; \Xi'_S \setminus \Xi_S.
$$

The third identity holds because $\boldsymbol{h}(\boldsymbol{\xi}) = \boldsymbol{H}_J \boldsymbol{\xi} + \boldsymbol{h}_0$ and $\boldsymbol{\lambda}_A(\boldsymbol{\xi}) = \boldsymbol{\Lambda}_J \boldsymbol{\xi} + \boldsymbol{\lambda}_0$, and the last identity follows from the linearity of the inner product operator. An analogous argument shows that $(\boldsymbol{\lambda}'_S, \boldsymbol{\lambda}'_A)$ satisfies the first constraint in $\underline{\mathcal{P}}(\Xi'_S)$, and one readily verifies that the remaining constraints are satisfied as well. Thus, the first statement follows. $\square$

The second property states that the (extended real-valued) optimal values of $\underline{\mathcal{P}}(\operatorname{ext}\Xi)$ and the two-stage robust optimization problem $\mathcal{P}$ coincide if at least one of the problems is feasible. It is possible, however, that both problems are infeasible: This is the case, for example, if $\boldsymbol{T}(\boldsymbol{\xi}) = \mathbf{0}$ and $\boldsymbol{W} = \mathbf{0}$ but $\boldsymbol{q} \neq \mathbf{0}$ and $\boldsymbol{h}(\boldsymbol{\xi}) \not\leq \mathbf{0}$ for some $\boldsymbol{\xi} \in \Xi$. Our incremental lifting scheme in Section 3 avoids this pathological case by solely operating on feasible instances of the bounding problems. For such instances, the second statement guarantees that the optimal values of $\underline{\mathcal{P}}(\operatorname{ext}\Xi)$ and $\mathcal{P}$ coincide. The first statement implies that $\underline{\mathcal{P}}(\Xi_S)$ provides a progressive approximation to $\underline{\mathcal{P}}(\Xi'_S)$ if $\Xi_S \subseteq \Xi'_S$, and in particular—by virtue of the second statement—to problem $\mathcal{P}$.

Remark 2 (Discrete Here-and-Now Decisions)**.** *Perhaps surprisingly, we can derive a convergent hierarchy of dual bounds to the mixed-integer two-stage robust optimization problem $\mathcal{P}_{\mathrm{d}}$ as well. To this end, consider the following extreme point reformulation $\mathcal{P}'_{\mathrm{d}}$ of problem $\mathcal{P}_{\mathrm{d}}$, where we separately optimize over the discrete and continuous here-and-now decisions:*

$$
\begin{array}{ll}
\text{minimize} & \boldsymbol{q}_{\mathrm{d}}^\top \boldsymbol{x}_{\mathrm{d}} + \left[\begin{array}{lll}
\text{minimize} & \boldsymbol{q}_{\mathrm{c}}^\top \boldsymbol{x}_{\mathrm{c}} & \\
\text{subject to} & \boldsymbol{T}_{\mathrm{c}}(\boldsymbol{\xi})\, \boldsymbol{x}_{\mathrm{c}} + \boldsymbol{W} \boldsymbol{y}(\boldsymbol{\xi}) \geq [\boldsymbol{h}(\boldsymbol{\xi}) - \boldsymbol{T}_{\mathrm{d}}(\boldsymbol{\xi})\, \boldsymbol{x}_{\mathrm{d}}] & \forall \boldsymbol{\xi} \in \operatorname{ext}\Xi \\
 & \boldsymbol{x}_{\mathrm{c}} \in \mathbb{R}^{nc_1}, \;\; \boldsymbol{y} : \operatorname{ext}\Xi \mapsto \mathbb{R}^{n_2} &
\end{array} \right] \\
\text{subject to} & \boldsymbol{x}_{\mathrm{d}} \in \mathbb{Z}^{nd_1}
\end{array}
$$

For any fixed discrete here-and-now decision $\boldsymbol{x}_{\mathrm{d}} \in \mathbb{Z}^{nd_1}$, we can replace the embedded minimization problem over the continuous here-and-now decision $\boldsymbol{x}_{\mathrm{c}} \in \mathbb{R}^{nc_1}$ with its linear programming dual:

$$
\begin{array}{ll}
\text{minimize} & \boldsymbol{q}_{\mathrm{d}}^\top \boldsymbol{x}_{\mathrm{d}} + \left[\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{\boldsymbol{\xi} \in \operatorname{ext}\Xi} [\boldsymbol{h}(\boldsymbol{\xi}) - \boldsymbol{T}_{\mathrm{d}}(\boldsymbol{\xi})\, \boldsymbol{x}_{\mathrm{d}}]^\top \boldsymbol{\lambda}(\boldsymbol{\xi}) & \\
\text{subject to} & \displaystyle\sum_{\boldsymbol{\xi} \in \operatorname{ext}\Xi} \boldsymbol{T}_{\mathrm{c}}(\boldsymbol{\xi})^\top \boldsymbol{\lambda}(\boldsymbol{\xi}) = \boldsymbol{q}_{\mathrm{c}} & \\
 & \boldsymbol{W}^\top \boldsymbol{\lambda}(\boldsymbol{\xi}) = \mathbf{0} & \forall \boldsymbol{\xi} \in \operatorname{ext}\Xi \\
 & \boldsymbol{\lambda} : \operatorname{ext}\Xi \mapsto \mathbb{R}^m_+ &
\end{array} \right] \\
\text{subject to} & \boldsymbol{x}_{\mathrm{d}} \in \mathbb{Z}^{nd_1}
\end{array}
$$

Similar arguments as before show that the inner maximization problem can be bounded from below:

$$
\begin{array}{ll}
\text{minimize} & \boldsymbol{q}_{\mathrm{d}}^{\top}\boldsymbol{x}_{\mathrm{d}} + \left[\begin{array}{ll}
\text{maximize} & \left\langle \left[\boldsymbol{H} - \sum_{i=1}^{nd_1} \boldsymbol{T}_{\mathrm{d},i}\, x_{\mathrm{d},i}\right]^{\top} \boldsymbol{\Lambda}, \boldsymbol{\Sigma}_A \right\rangle + \sum_{\boldsymbol{\xi}\in\Xi_S} \left[\boldsymbol{h}(\boldsymbol{\xi}) - \boldsymbol{T}_{\mathrm{d}}(\boldsymbol{\xi})\,\boldsymbol{x}_{\mathrm{d}}\right]^{\top} \boldsymbol{\lambda}_S(\boldsymbol{\xi}) \\
\text{subject to} & \left\langle \boldsymbol{T}_{\mathrm{c},i}{}^{\top}\boldsymbol{\Lambda}, \boldsymbol{\Sigma}_A \right\rangle + \sum_{\boldsymbol{\xi}\in\Xi_S} \boldsymbol{T}_{\mathrm{c},i}(\boldsymbol{\xi})^{\top}\boldsymbol{\lambda}_S(\boldsymbol{\xi}) = q_{\mathrm{c},i} \qquad \forall i = 1,\ldots,nc_1 \\
& \boldsymbol{W}^{\top}\boldsymbol{\lambda}_S(\boldsymbol{\xi}) = \mathbf{0} \qquad\qquad \forall \boldsymbol{\xi} \in \Xi_S \\
& \boldsymbol{W}^{\top}\boldsymbol{\lambda}_A(\boldsymbol{\xi}) = \mathbf{0} \qquad\qquad \forall \boldsymbol{\xi} \in \Xi_A \\
& \boldsymbol{\lambda}_S : \Xi_S \mapsto \mathbb{R}^m_+,\;\; \boldsymbol{\lambda}_A : \Xi_A \overset{A}{\mapsto} \mathbb{R}^m_+,
\end{array}\right] \\
\text{subject to} & \boldsymbol{x}_{\mathrm{d}} \in \mathbb{Z}^{nd_1}
\end{array}
$$

Here, $\boldsymbol{T}_{\mathrm{c},i} = (\boldsymbol{t}_{\mathrm{c},i,0}, \boldsymbol{T}_{\mathrm{c},i,J})$, $\boldsymbol{T}_{\mathrm{d},i} = (\boldsymbol{t}_{\mathrm{d},i,0}, \boldsymbol{T}_{\mathrm{d},i,J})$ *with* $\boldsymbol{t}_{\mathrm{c},i,0}, \boldsymbol{t}_{\mathrm{d},i,0} \in \mathbb{R}^m$ *denote the intercepts and* $\boldsymbol{T}_{\mathrm{c},i,J}, \boldsymbol{T}_{\mathrm{d},i,J} \in \mathbb{R}^{m\times k}$ *the slopes of the affine functions* $\boldsymbol{T}_{\mathrm{c},i}(\boldsymbol{\xi}) = (T_{\mathrm{c},1,i}(\boldsymbol{\xi}),\ldots,T_{\mathrm{c},m,i}(\boldsymbol{\xi}))^{\top}$ *and* $\boldsymbol{T}_{\mathrm{d},i}(\boldsymbol{\xi}) = (T_{\mathrm{d},1,i}(\boldsymbol{\xi}),\ldots,T_{\mathrm{d},m,i}(\boldsymbol{\xi}))^{\top}$. *For any fixed discrete here-and-now decision* $\boldsymbol{x}_{\mathrm{d}} \in \mathbb{Z}^{nd_1}$, *the embedded maximization problem is a robust optimization problem that is linear in the decision variables* $\boldsymbol{\lambda}_S$ *and* $\boldsymbol{\lambda}_A$. *Using robust optimization techniques to reformulate this semi-infinite problem as a linear program and subsequently invoking strong linear programming duality results in an embedded* minimization *problem that attains the same optimal value. The resulting min-min problem collapses to the following single-stage mixed-integer linear program:*

$$
\begin{array}{ll}
\text{minimize} & \boldsymbol{q}_{\mathrm{d}}{}^{\top}\boldsymbol{x}_{\mathrm{d}} + \boldsymbol{q}_{\mathrm{c}}{}^{\top}\boldsymbol{\alpha} \\
\text{subject to} & \boldsymbol{h}(\boldsymbol{\xi}) - \boldsymbol{T}_{\mathrm{d}}(\boldsymbol{\xi})\,\boldsymbol{x}_{\mathrm{d}} - \sum_{i=1}^{nc_1} \alpha_i \cdot \boldsymbol{T}_{\mathrm{c},i}(\boldsymbol{\xi}) + \boldsymbol{W}\boldsymbol{\beta}(\boldsymbol{\xi}) \leq \mathbf{0} \qquad \forall \boldsymbol{\xi} \in \Xi_S \\
& \left[\boldsymbol{H} - \sum_{i=1}^{nd_1} \boldsymbol{T}_{\mathrm{d},i}\, x_{\mathrm{d},i} - \sum_{i=1}^{nc_1} \alpha_i \boldsymbol{T}_{\mathrm{c},i}\right]\boldsymbol{\Sigma}_A = \big(\boldsymbol{W}(\boldsymbol{\gamma}^1 - \boldsymbol{\gamma}^2) - \boldsymbol{\eta}\;\; \boldsymbol{W}(\boldsymbol{\Delta}^1 - \boldsymbol{\Delta}^2) - \boldsymbol{Z}\big) \\
& \boldsymbol{F}\boldsymbol{\delta}^1_{\nu} \leq \gamma^1_{\nu} \cdot \boldsymbol{g},\;\; \boldsymbol{F}\boldsymbol{\delta}^2_{\nu} \leq \gamma^2_{\nu} \cdot \boldsymbol{g} \qquad \forall \nu = 1,\ldots,n_2 \\
& \boldsymbol{F}\boldsymbol{\zeta}_{\mu} \leq \eta_{\mu} \cdot \boldsymbol{g} \qquad \forall \mu = 1,\ldots,m \\
& \boldsymbol{\alpha} \in \mathbb{R}^{nc_1},\;\; \boldsymbol{\beta}(\boldsymbol{\xi}) \in \mathbb{R}^{n_2},\; \boldsymbol{\xi} \in \Xi_S, \\
& \boldsymbol{\gamma}^1 \in \mathbb{R}^{n_2}_+,\;\; \boldsymbol{\gamma}^2 \in \mathbb{R}^{n_2}_-,\;\; \boldsymbol{\Delta}^1, \boldsymbol{\Delta}^2 \in \mathbb{R}^{n_2\times k},\;\; \boldsymbol{\eta} \in \mathbb{R}^m_+,\;\; \boldsymbol{Z} \in \mathbb{R}^{m\times k} \\
& \boldsymbol{x}_{\mathrm{d}} \in \mathbb{Z}^{nd_1}.
\end{array}
$$

In this formulation, we assume that $\Xi_A = \{\boldsymbol{\xi} \in \mathbb{R}^k : \boldsymbol{F}\boldsymbol{\xi} \leq \boldsymbol{g}\}$ *with* $\boldsymbol{F} \in \mathbb{R}^{l\times k}$ *and* $\boldsymbol{g} \in \mathbb{R}^l$, *the matrices* $\boldsymbol{\Delta}^i \in \mathbb{R}^{n_2\times k}, i = 1,2$, *contain the rows* $(\boldsymbol{\delta}^i_{\nu})^{\top} \in \mathbb{R}^k$, $\nu = 1,\ldots,n_2$, *and the matrix*

$\boldsymbol{Z} \in \mathbb{R}^{m\times k}$ *contains the rows* $(\boldsymbol{\zeta}_\mu)^\top \in \mathbb{R}^k$, $\mu = 1, \ldots, m$.

We note that a similar reformulation can be derived for the linear two-stage robust optimization problem $\mathcal{P}$*. Since this reformulation lacks the intuitive structure that our dual bounds* $\underline{\mathcal{P}}(\Xi_S)$ *for the problem* $\mathcal{P}$ *afford, however, we skip its derivation for the sake of brevity.*

As in the previous section, the size of problem $\underline{\mathcal{P}}(\Xi_S)$ depends on the number of scenarios $|\Xi_S|$ and the number of inequalities ℓ required to describe Ξ_A, both of which depend on the choice of the uncertainty set Ξ. In addition, the dual approximation $\underline{\mathcal{P}}(\Xi_S)$ contains the moment matrix $\boldsymbol{\Sigma}_A$ which appears difficult to compute as it involves sums of the (outer products of the) extreme points in the affine set Ξ_A. For several class of commonly used uncertainty sets, we show in the appendix how $\boldsymbol{\Sigma}_A$ can be derived analytically when $\Xi_A = \Xi$. In the next section we present a solution scheme for the two-stage robust optimization problem $\mathcal{P}$ which computes the dual affine decision rule bound $\underline{\mathcal{P}}(\emptyset)$ and subsequently removes individual vertices of $\operatorname{ext}\Xi_A$. Thus, if the initial moment matrix $\boldsymbol{\Sigma}_A$ for $\Xi_A = \Xi$ can be computed efficiently, then the subsequent updates of Ξ_A merely require the subtraction of individual extreme points. On the other hand, it remains unclear whether the moment matrices $\boldsymbol{\Sigma}_A$ corresponding to generic polyhedral uncertainty sets can be calculated efficiently, that is, without explicitly enumerating them.

We note that our dual bound $\underline{\mathcal{P}}(\Xi_S)$ is closely related to the lower bound proposed in [37].

Remark 3 (**Relation to the Sampling Bound of [37]**)**.** *The progressive bound of [37] can be interpreted as a relaxation of the extreme point reformulation* $\mathcal{P}'$ *that only considers the scenarios* $\boldsymbol{\xi} \in \Xi_S$ *in a single, fixed scenario set* $\Xi_S \subseteq \operatorname{ext}\Xi$*:*

$$\begin{array}{lll} \text{minimize} & \boldsymbol{q}^\top \boldsymbol{x} & \\ \text{subject to} & \boldsymbol{T}(\boldsymbol{\xi})\,\boldsymbol{x} + \boldsymbol{W}\boldsymbol{y}(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi}) & \forall \boldsymbol{\xi} \in \Xi_S \\ & \boldsymbol{x} \in \mathbb{R}^{n_1}, \;\; \boldsymbol{y} : \Xi_S \mapsto \mathbb{R}^{n_2} & \end{array} \tag{SB}$$

The dual associated with this problem corresponds to an instance of our dual bound $\underline{\mathcal{P}}(\Xi_S)$ *where we fix* $\boldsymbol{\lambda}_A(\boldsymbol{\xi}) = \boldsymbol{0}$*,* $\boldsymbol{\xi} \in \Xi_A$*, and only optimize over* $\boldsymbol{\lambda}_S$*. Thus, our bound* $\underline{\mathcal{P}}(\Xi_S)$ *is at least as tight as the bound of [37].* $\square$

We close this section with a summary of the findings of Propositions 2 and 3:

Theorem 1 (Duality)**.** *The primal and dual bounds* $\overline{\mathcal{P}}(\Xi_S)$ *and* $\underline{\mathcal{P}}(\Xi_S)$ *satisfy:*

(i) ***Weak Duality.*** $\underline{\mathcal{P}}(\Xi_S) \leq \mathcal{P} \leq \overline{\mathcal{P}}(\Xi_S)$ *for all* $\Xi_S \subseteq \operatorname{ext} \Xi$*.*

(ii) ***Strong Duality.*** $\underline{\mathcal{P}}(\Xi_S) = \mathcal{P} = \overline{\mathcal{P}}(\Xi_S)$ *for* $\Xi_S = \operatorname{ext} \Xi$ *if* $\mathcal{P}$ *is feasible.*

3 Primal-Dual Lifting Scheme

Our solution scheme for problem $\mathcal{P}$ starts with a *feasibility phase*, which determines a feasible solution (or recognizes that no such solution exists), and then proceeds with an *optimality phase*, which computes an optimal solution (or identifies that the problem is unbounded). The algorithm starts with the efficiently computable affine decision rule bounds $\overline{\mathcal{P}}(\emptyset)$ and $\underline{\mathcal{P}}(\emptyset)$ and iteratively transfers extreme points $\boldsymbol{\xi}^\star \in \operatorname{ext} \Xi_A$ of the affine set to the scenario set Ξ_S.

Our algorithm can be summarized as follows.

1. **Initialization.** Set $\Xi_A = \Xi$ and $\Xi_S = \emptyset$.

2. **Feasibility Phase.** Consider the *feasibility problem*

$$\begin{array}{ll} \text{minimize} & v \\ \text{subject to} & \boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x} + \boldsymbol{W} \boldsymbol{y}(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi}) - v\mathbf{e} \qquad \forall \boldsymbol{\xi} \in \Xi \\ & \boldsymbol{x} \in \mathbb{R}^{n_1}, \;\; \boldsymbol{y} : \Xi \mapsto \mathbb{R}^{n_2}, \;\; v \in \mathbb{R}_+. \end{array}$$

   (i) Solve $\overline{\mathcal{P}}(\Xi_S)$ and $\underline{\mathcal{P}}(\Xi_S)$ associated with this problem. If the optimal value of $\overline{\mathcal{P}}(\Xi_S)$ is zero, go to Step 3: the current solution to $\overline{\mathcal{P}}(\Xi_S)$ is feasible in $\mathcal{P}$. If the optimal value of $\underline{\mathcal{P}}(\Xi_S)$ is strictly positive, terminate: problem $\mathcal{P}$ is infeasible.

   (ii) Select any $\boldsymbol{\xi}^\star \in \Xi_A^\star$ (defined below), update $\Xi_S \leftarrow \Xi_S \cup \{\boldsymbol{\xi}^\star\}$ as well as $\Xi_A \leftarrow \operatorname{conv}([\operatorname{ext} \Xi_A] \setminus \{\boldsymbol{\xi}^\star\})$, and go to back to Step 2(i).

3. **Optimality Phase.** Consider the *optimality problem*

$$\begin{array}{ll} \text{minimize} & \boldsymbol{q}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x} + \boldsymbol{W} \boldsymbol{y}(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi}) \qquad \forall \boldsymbol{\xi} \in \Xi \\ & \boldsymbol{x} \in \mathbb{R}^{n_1}, \;\; \boldsymbol{y} : \Xi \mapsto \mathbb{R}^{n_2}. \end{array}$$

   (i) Solve $\overline{\mathcal{P}}(\Xi_S)$ and $\underline{\mathcal{P}}(\Xi_S)$ associated with this problem. If $\overline{\mathcal{P}}(\Xi_S)$ is unbounded, terminate:

problem $\mathcal{P}$ is unbounded. Otherwise, if the optimal values of $\overline{\mathcal{P}}(\Xi_S)$ and $\underline{\mathcal{P}}(\Xi_S)$ coincide, terminate: the current solution to $\overline{\mathcal{P}}(\Xi_S)$ is optimal in $\mathcal{P}$.

(ii) Select any $\boldsymbol{\xi}^\star \in \Xi_A^\star$ (defined below), update $\Xi_S \leftarrow \Xi_S \cup \{\boldsymbol{\xi}^\star\}$ as well as $\Xi_A \leftarrow \operatorname{conv}([\operatorname{ext} \Xi_A] \setminus \{\boldsymbol{\xi}^\star\})$, and go to back to Step 3(i).

The *feasibility phase* operates on a feasibility variant of problem $\mathcal{P}$ that minimizes the maximum constraint violation. Both bounding problems $\overline{\mathcal{P}}(\Xi_S)$ and $\underline{\mathcal{P}}(\Xi_S)$ associated with this feasibility problem are always feasible since Ξ is bounded. Thus, Propositions 2 and 3 guarantee the validity of the conservative and progressive bounds, and the update step 2(ii) ensures that both bounds converge monotonically. After finitely many iterations, either the conservative bound evaluates to zero (indicating that a feasible solution has been found), or the progressive bound becomes strictly positive (indicating that the problem is infeasible).

If a feasible solution has been found, the algorithm proceeds with the *optimality phase* and computes conservative and progressive bounds on the problem $\mathcal{P}$ itself. Since a feasible solution has already been determined, these bounds are valid and converge monotonically. After finitely many iterations, either the bounds coincide (indicating that an optimal solution has been found), or the conservative bound evaluates to $-\infty$ (indicating that the problem is unbounded).

The update steps 2(ii) and 3(ii) move one extreme point of the polyhedron Ξ_A to the scenario set Ξ_S. Ideally, we would transfer the extreme point that leads to the greatest improvement of the conservative and progressive bounds. To this end, we define the *binding scenario set* as $\Xi_A^\star = \{\boldsymbol{\xi} \in \operatorname{ext} \Xi_A \, : \, \boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x} + \boldsymbol{W} \boldsymbol{y}(\boldsymbol{\xi}) \not> \boldsymbol{h}(\boldsymbol{\xi}) - v\mathbf{e}\}$ in the feasibility phase and as $\Xi_A^\star = \{\boldsymbol{\xi} \in \operatorname{ext} \Xi_A \, : \, \boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x} + \boldsymbol{W} \boldsymbol{y}(\boldsymbol{\xi}) \not> \boldsymbol{h}(\boldsymbol{\xi})\}$ in the optimality phase. We now show that we can restrict ourselves to these binding scenarios in the update steps.

Observation 1 (Binding Scenarios)**.** *For any scenario set $\Xi_S \subseteq \operatorname{ext} \Xi$, the binding scenario set $\Xi_A^\star$ satisfies the following two properties.*

(i) If $\boldsymbol{\xi}^\star \notin \Xi_A^\star$, then the update $\Xi_S \leftarrow \Xi_S \cup \{\boldsymbol{\xi}^\star\}$ and $\Xi_A \leftarrow \operatorname{conv}([\operatorname{ext} \Xi_A] \setminus \{\boldsymbol{\xi}^\star\})$ does not improve the objective value of the conservative approximation $\overline{\mathcal{P}}(\Xi_S)$.

(ii) If $\Xi_A^\star = \emptyset$, then $\overline{\mathcal{P}}(\Xi_S) = \underline{\mathcal{P}}(\Xi_S)$.

Proof of Observation 1. We prove both statements for the optimality problem in Step 3 of our primal-dual lifting scheme; similar arguments apply to the feasibility problem in Step 2.

As for the first statement, let $(\boldsymbol{x}, \boldsymbol{y}_A, \boldsymbol{y}_S)$ be an optimal solution to the conservative approximation $\overline{\mathcal{P}}(\Xi_S)$ in Step 3(i), and assume that $\boldsymbol{y}_A(\boldsymbol{\xi}) = \boldsymbol{Y}_A \boldsymbol{\xi} + \boldsymbol{y}_A$ for all $\boldsymbol{\xi} \in \Xi_A$. Fix a non-binding scenario $\boldsymbol{\xi}^\star \in \operatorname{ext} \Xi_A \setminus \Xi_A^\star$, and assume to the contrary that $\overline{\mathcal{P}}(\Xi_S') < \overline{\mathcal{P}}(\Xi_S)$ for the bounding problem $\overline{\mathcal{P}}(\Xi_S')$, $\Xi_S' = \Xi_S \cup \{\boldsymbol{\xi}^\star\}$, that results from lifting the scenario $\boldsymbol{\xi}^\star$. Let $(\boldsymbol{x}', \boldsymbol{y}_A', \boldsymbol{y}_S')$ be an optimal solution to the lifted problem $\overline{\mathcal{P}}(\Xi_S')$, and assume that $\boldsymbol{y}_A'(\boldsymbol{\xi}) = \boldsymbol{Y}_A' \boldsymbol{\xi} + \boldsymbol{y}_A'$ for all $\boldsymbol{\xi} \in \Xi_A'$. Consider the convex combinations $\hat{\boldsymbol{x}} = \lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{x}'$, $\hat{\boldsymbol{y}}_A(\boldsymbol{\xi}) = [\lambda \boldsymbol{Y}_A + (1-\lambda)\boldsymbol{Y}_A'] \boldsymbol{\xi} + [\lambda \boldsymbol{y}_A + (1-\lambda)\boldsymbol{y}_A']$, $\boldsymbol{\xi} \in \Xi_A$, and $\hat{\boldsymbol{y}}_S(\boldsymbol{\xi}) = \lambda \boldsymbol{y}_S(\boldsymbol{\xi}) + (1-\lambda)\boldsymbol{y}_S'(\boldsymbol{\xi})$, $\boldsymbol{\xi} \in \Xi_S$. We show that for $\lambda \uparrow 1$, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}_A, \hat{\boldsymbol{y}}_S)$ is feasible in the conservative approximation $\overline{\mathcal{P}}(\Xi_S)$. Since $\boldsymbol{q}^\top \hat{\boldsymbol{x}} < \boldsymbol{q}^\top \boldsymbol{x}$, this would contradict the optimality of $(\boldsymbol{x}, \boldsymbol{y}_A, \boldsymbol{y}_S)$ in $\overline{\mathcal{P}}(\Xi_S)$.

We first note that for all $\boldsymbol{\xi} \in \Xi_S$, we have

$$\begin{aligned} \boldsymbol{T}(\boldsymbol{\xi})\, \hat{\boldsymbol{x}} + \boldsymbol{W} \hat{\boldsymbol{y}}_S(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi}) \quad &\Longleftrightarrow \quad \boldsymbol{T}(\boldsymbol{\xi}) \left[\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{x}'\right] + \boldsymbol{W} \left[\lambda \boldsymbol{y}_S(\boldsymbol{\xi}) + (1-\lambda)\boldsymbol{y}_S'(\boldsymbol{\xi})\right] \geq \boldsymbol{h}(\boldsymbol{\xi}) \\ &\Longleftrightarrow \quad \lambda \left[\boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x} + \boldsymbol{W} \boldsymbol{y}_S(\boldsymbol{\xi})\right] + (1-\lambda) \left[\boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x}' + \boldsymbol{W} \boldsymbol{y}_S'(\boldsymbol{\xi})\right] \geq \boldsymbol{h}(\boldsymbol{\xi}), \end{aligned}$$

and the last inequality holds since $\boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x} + \boldsymbol{W} \boldsymbol{y}_S(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi})$ and $\boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x}' + \boldsymbol{W} \boldsymbol{y}_S'(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi})$ for all $\boldsymbol{\xi} \in \Xi_S$ by construction. Similarly, we observe that for all $\boldsymbol{\xi} \in \Xi_A$, we have

$$\begin{aligned} &\boldsymbol{T}(\boldsymbol{\xi})\, \hat{\boldsymbol{x}} + \boldsymbol{W} \hat{\boldsymbol{y}}_A(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi}) \\ \Longleftrightarrow\ &\boldsymbol{T}(\boldsymbol{\xi}) \left[\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{x}'\right] + \boldsymbol{W} \left[\lambda(\boldsymbol{Y}_A \boldsymbol{\xi} + \boldsymbol{y}_A) + (1-\lambda)(\boldsymbol{Y}_A' \boldsymbol{\xi} + \boldsymbol{y}_A')\right] \geq \boldsymbol{h}(\boldsymbol{\xi}) \\ \Longleftrightarrow\ &\lambda \left[\boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x} + \boldsymbol{W}(\boldsymbol{Y}_A \boldsymbol{\xi} + \boldsymbol{y}_A)\right] + (1-\lambda) \left[\boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x}' + \boldsymbol{W}(\boldsymbol{Y}_A' \boldsymbol{\xi} + \boldsymbol{y}_A')\right] \geq \boldsymbol{h}(\boldsymbol{\xi}). \end{aligned}$$

The last inequality holds for all $\boldsymbol{\xi} \in \operatorname{ext} \Xi_A'$ since $\boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x} + \boldsymbol{W} \boldsymbol{y}_A(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi})$ and $\boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x}' + \boldsymbol{W} \boldsymbol{y}_A'(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi})$ for all $\boldsymbol{\xi} \in \operatorname{ext} \Xi_A'$ by construction. We furthermore observe that $\boldsymbol{T}(\boldsymbol{\xi}^\star)\, \hat{\boldsymbol{x}} + \boldsymbol{W} \hat{\boldsymbol{y}}_A(\boldsymbol{\xi}^\star) \geq \boldsymbol{h}(\boldsymbol{\xi}^\star)$ for all λ sufficiently close to 1 since $\boldsymbol{T}(\boldsymbol{\xi}^\star)\, \boldsymbol{x} + \boldsymbol{W}(\boldsymbol{Y}_A \boldsymbol{\xi}^\star + \boldsymbol{y}_A) > \boldsymbol{h}(\boldsymbol{\xi}^\star)$ by assumption. We thus conclude that $\boldsymbol{T}(\boldsymbol{\xi})\, \hat{\boldsymbol{x}} + \boldsymbol{W} \hat{\boldsymbol{y}}_A(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi})$ for all $\boldsymbol{\xi} \in \operatorname{ext} \Xi_A$ as long as λ sufficiently close to 1. The linearity of $\boldsymbol{T}$, $\hat{\boldsymbol{y}}_A$ and $\boldsymbol{h}$ then implies that $\boldsymbol{T}(\boldsymbol{\xi})\, \hat{\boldsymbol{x}} + \boldsymbol{W} \hat{\boldsymbol{y}}_A(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi})$ for all $\boldsymbol{\xi} \in \Xi_A$ as desired.

In view of the second statement, we note that if $\Xi_A^\star = \emptyset$, then we can remove from the conservative approximation $\overline{\mathcal{P}}(\Xi_S)$ all constraints involving the realizations $\boldsymbol{\xi} \in \Xi_A$ without changing the optimal value of the problem. The dual of this reduced problem can be interpreted as an instance of the progressive approximation $\underline{\mathcal{P}}(\Xi_S)$ where $\boldsymbol{\lambda}_A(\boldsymbol{\xi}) = \mathbf{0}$ for all $\boldsymbol{\xi} \in \Xi_A$. Strong linear programming duality holds since $\overline{\mathcal{P}}(\Xi_S)$ is feasible by construction. The statement then follows since the dual of the reduced problem is a restriction of the progressive approximation $\underline{\mathcal{P}}(\Xi_S)$ that attains the same optimal value as the conservative approximation $\overline{\mathcal{P}}(\Xi_S)$. □

The first property of Observation 1 implies that the update steps 2(ii) and 3(ii) only need to consider the scenarios $\boldsymbol{\xi}^\star$ of the binding scenario set $\Xi_A^\star$. The second property guarantees that the binding scenario set is never empty: If $\Xi_A^\star$ was empty in Step 2(ii), then $\overline{\mathcal{P}}(\Xi_S) = \underline{\mathcal{P}}(\Xi_S)$ in Step 2(i), which implies that the algorithm would have either proceeded with Step 3 (if both bounds are zero) or terminated (if both bounds are positive). A similar reasoning applies if $\Xi_A^\star$ was empty in Step 3(ii). We note that the crucial role of binding scenarios has previously been recognized in the context of uncertainty set partitioning approaches, see [13], [49] and [52].

Observation 1 does not specify which scenario $\boldsymbol{\xi}^\star \in \Xi_A^\star$ to lift. A natural approach to determine candidate scenarios $\boldsymbol{\xi}^\star \in \Xi_A^\star$ to lift is to fix the optimal solution to the current bounding problem $\overline{\mathcal{P}}(\Xi_S)$ and determine one binding scenario for each constraint by minimizing the constraint's slack over all $\boldsymbol{\xi} \in \Xi_A$. This requires the solution of a linear program for each constraint. Any of the binding scenarios thus identified is a candidate for the lifting in Steps 2(ii) and 3(ii) of the algorithm. We describe a more elaborate selection heuristic in our numerical example in Section 5.2.

The updates of the affine set Ξ_A in Steps 2(ii) and 3(ii) guarantee that $\Xi_A = \operatorname{conv}([\operatorname{ext}\Xi] \setminus \Xi_S)$ throughout the algorithm. The update steps are intimately related to the 'Forbidden Vertices Problem' [3], which optimizes a linear function over all but a few designated vertices of a polyhedron. The update steps can be implemented without enumerating all vertices of Ξ_A. To this end, we determine the neighbouring extreme points of $\boldsymbol{\xi}^\star$ in Ξ_A, $N(\boldsymbol{\xi}^\star)$, for example through simplex pivoting steps [28]. We then determine the halfspaces defining $\operatorname{conv} N(\boldsymbol{\xi}^\star)$ via facet enumeration [51], and we add those halfspaces of $\operatorname{conv} N(\boldsymbol{\xi}^\star)$ to the description of Ξ_A that do not contain the extreme point $\boldsymbol{\xi}^\star$ to be removed. Figure 2 illustrates our update procedure.

We are now ready to prove the correctness of our iterative solution scheme.

Theorem 2 (Finite Convergence)**.** *The algorithm terminates in finite time, and it either determines an optimal solution to $\mathcal{P}$ or it correctly identifies infeasibility or unboundedness of the problem.*

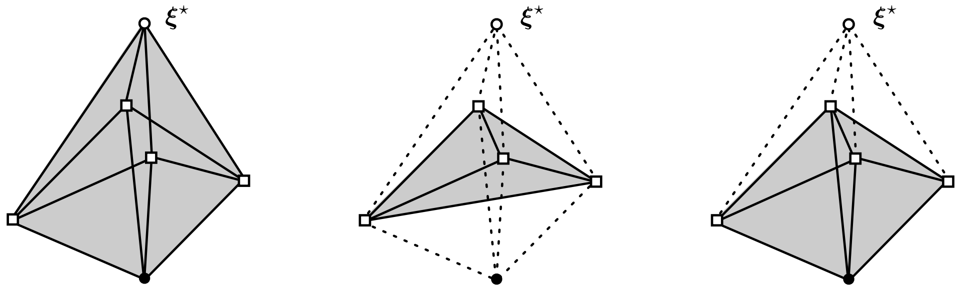

Figure 2. Illustration of the update step in $\mathbb{R}^3$. On the left, the circular vertex $\boldsymbol{\xi}^\star$ to be removed from the distorted diamond has the four square vertices as neighbours. The middle polyhedron illustrates $\operatorname{conv} N(\boldsymbol{\xi}^\star)$, the convex hull of these neighbours. Out of the four halfspaces defining this convex hull, the two upper do not contain $\boldsymbol{\xi}^\star$ and are thus added to the updated description of Ξ_A on the right.

Proof of Theorem 2. We first show that the algorithm terminates after at most $|\operatorname{ext} \Xi|$ executions of the solution steps 2(i) and 3(i). Indeed, assume to the contrary that the algorithm would execute Steps 2(i) and 3(i) more than $|\operatorname{ext} \Xi|$ times. During each execution of the update steps 2(ii) and 3(ii), one extreme point of the affine set Ξ_A is transferred to the scenario set Ξ_S. Thus, after $|\operatorname{ext} \Xi|$ executions of the update steps, the bounding problems in the solution steps 2(i) and 3(i) become $\underline{\mathcal{P}}(\operatorname{ext} \Xi)$ and $\overline{\mathcal{P}}(\operatorname{ext} \Xi)$. Since the feasibility and optimality bounding problems are feasible by construction, we must have $\underline{\mathcal{P}}(\operatorname{ext} \Xi) = \overline{\mathcal{P}}(\operatorname{ext} \Xi)$. In that case, however, either $\underline{\mathcal{P}}(\operatorname{ext} \Xi) > 0$ in Step 2(i) or one of the two termination criteria in Step 3(i) is met.

If the algorithm terminates because $\underline{\mathcal{P}}(\Xi_S) > 0$ in Step 2(i), then there is no solution $(\boldsymbol{x}, \boldsymbol{y}, v)$ to the progressive bound $\underline{\mathcal{P}}(\Xi_S)$ such that $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfy $\boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x} + \boldsymbol{W} \boldsymbol{y}(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi})$ for all $\boldsymbol{\xi} \in \Xi$. Proposition 3 then implies that there is no solution $(\boldsymbol{x}, \boldsymbol{y})$ to problem $\mathcal{P}$ that satisfies $\boldsymbol{T}(\boldsymbol{\xi})\, \boldsymbol{x} + \boldsymbol{W} \boldsymbol{y}(\boldsymbol{\xi}) \geq \boldsymbol{h}(\boldsymbol{\xi})$ for all $\boldsymbol{\xi} \in \Xi$, that is, problem $\mathcal{P}$ is indeed infeasible.

Assume now that the algorithm terminates because $\overline{\mathcal{P}}(\Xi_S)$ unbounded in Step 3(i). By Proposition 2, any feasible solution to $\overline{\mathcal{P}}(\Xi_S)$ can be transformed into a feasible solution to $\mathcal{P}$ that achieves the same objective value. We thus conclude that problem $\mathcal{P}$ is unbounded as well.

Finally, assume that the algorithm terminates because $\underline{\mathcal{P}}(\Xi_S) = \overline{\mathcal{P}}(\Xi_S)$ in Step 3(i). Since the bounding problems are feasible by construction, Propositions 2 and 3 imply that

$$\overline{\mathcal{P}}(\Xi_S) \;\; \geq \;\; \overline{\mathcal{P}}(\operatorname{ext} \Xi) \;\; = \;\; \mathcal{P} \;\; = \;\; \underline{\mathcal{P}}(\operatorname{ext} \Xi) \;\; \geq \;\; \underline{\mathcal{P}}(\Xi_S),$$

where we denote by $\mathcal{P}$ the optimal value of problem $\mathcal{P}$. Since $\underline{\mathcal{P}}(\Xi_S) = \overline{\mathcal{P}}(\Xi_S)$, however, we know that the optimal values of all these problems coincide. Proposition 2 then allows us to transform any optimal solution to $\overline{\mathcal{P}}(\Xi_S)$ into an optimal solution to problem $\mathcal{P}$. $\square$

The runtime of the algorithm is determined by *(i)* the number of iterations (*i.e.*, the number of times that Steps 2 and 3 are executed), *(ii)* the size of the bounding problems $\overline{\mathcal{P}}(\Xi_S)$ and $\underline{\mathcal{P}}(\Xi_S)$ in each iteration, *(iii)* the complexity of selecting the scenario $\boldsymbol{\xi}^\star \in \Xi_A^\star$ to be removed, *(iv)* the complexity of the update $\Xi_A \leftarrow \operatorname{conv}([\operatorname{ext}\Xi_A] \setminus \{\boldsymbol{\xi}^\star\})$ and *(v)* the computation of the moment matrices $\boldsymbol{\Sigma}_A$. The algorithm performs up to $|\operatorname{ext}\Xi|$ iterations in the worst case, and an upper bound for this number is provided in Section 2. The complexity of the bounding problems, as well as the selection of $\boldsymbol{\xi}^\star \in \Xi_A^\star$, primarily depends on the description of the intermediate sets Ξ_A, which has been discussed in Section 2.1. For our lifting method, the time required for the update $\Xi_A \leftarrow \operatorname{conv}([\operatorname{ext}\Xi_A] \setminus \{\boldsymbol{\xi}^\star\})$ is determined by:

(i) *The size of* $N(\boldsymbol{\xi}^\star)$*:* This quantity depends on the degeneracy of $\boldsymbol{\xi}^\star$ in Ξ_A. If $k+\sigma$ constraints of Ξ_A are binding at $\boldsymbol{\xi}^\star$, then $\boldsymbol{\xi}^\star$ has at most $\binom{k+\sigma}{k-1}(\ell-k-\sigma)$ neighbours, where ℓ denotes the number of constraints that describe Ξ_A. In the worst case, each vertex of Ξ_A is a neighbour of $\boldsymbol{\xi}^\star$, which is the case if Ξ_A is 2-neighbourly. The primal upper bound theorem implies that $\operatorname{conv} N(\boldsymbol{\xi}^\star)$ can consist of up to $\binom{|N(\boldsymbol{\xi}^\star)| - \lceil k/2 \rceil}{\lfloor k/2 \rfloor} + \binom{|N(\boldsymbol{\xi}^\star)| - 1 - \lceil (k-1)/2 \rceil}{\lfloor (k-1)/2 \rfloor}$ halfspaces, and this bound is attained by primal cyclic polytopes [51].

(ii) *The complexity of computing the convex hull* $\operatorname{conv} N(\boldsymbol{\xi}^\star)$*:* This convex hull can be computed in time $\mathcal{O}(|N(\boldsymbol{\xi}^\star)|^{\lfloor k/2 \rfloor})$.

It can be shown that even when a single scenario $\boldsymbol{\xi}^\star$ is lifted, our description of $\Xi_A = \operatorname{conv}([\operatorname{ext}\Xi_A] \setminus \{\boldsymbol{\xi}^\star\}$ can grow exponentially [3, Proposition 5]. While there are *lifted* formulations of Ξ_A that only grow *quadratically* if a single scenario is lifted [3, Proposition 6], it is known that optimizing a linear function over Ξ_A after a flexible (*i.e.*, not a priori fixed) number of scenarios has been lifted is NP-hard [3, Theorem 11] for generic polyhedral uncertainty sets. On the other hand, for special classes of uncertainty sets, such as uncertainty sets with 0-1 vertices, the same problem is solvable in polynomial time if a suitable lifted formulation of Ξ_A is used [3, §3]. The initial moment matrix $\boldsymbol{\Sigma}_A$, finally, can often be computed efficiently, see the appendix, and updating $\boldsymbol{\Sigma}_A$ only requires to subtract the (outer product of the) scenario $\boldsymbol{\xi}^\star \in \Xi_A^\star$ that is removed in each iteration.

We close this section by outlining possible extensions of our iterative solution scheme.

Remark 4 (Discrete Here-and-Now Decisions)**.** *Theorem 2 does not exploit any specific properties of the here-and-now decisions* $\boldsymbol{x}$*, and our primal-dual lifting scheme therefore immediately carries over to the mixed-integer two-stage robust optimization problem* $\mathcal{P}_{\mathrm{d}}$*. Observation 1, on the other hand, crucially relies on the convexity in* $\boldsymbol{x}$*, and one can construct counterexamples where lifting a non-binding scenario results in an improved bound. Thus, in the presence of discrete here-and-now decisions we can no longer restrict ourselves to lifting scenarios* $\boldsymbol{\xi}^\star \in \Xi_A^\star$*.*

Remark 5 (Other Extensions)**.** *The algorithm can be altered in several ways. For example, the progressive and conservative bounding problems in Steps 2(i) and 3(i) could involve multiple affine sets* $\Xi_{A,1}, \ldots, \Xi_{A,s}$*, or they could assign piecewise affine decision rules to the realizations* $\boldsymbol{\xi} \in \Xi_A$*. If convergence to an optimal solution is not required, then the affine set* Ξ_A *could be replaced with outer approximations of* $\mathrm{conv}\left([\mathrm{ext}\,\Xi] \setminus \Xi_S\right)$ *without affecting the validity of the bounds. This is advantageous if these outer approximations have compact descriptions, as is the case for Löwner-John ellipsoids [40], for example. One could also transfer multiple extreme points to* Ξ_S *in Steps 2(ii) and 3(ii), and it might be advantageous to transfer extreme points back to* Ξ_A*. Finally, one could envision lifting different extreme points for the primal and the dual bounds.* □

4 Relation to Piecewise Affine Decision Rules

While a feasible solution $(\boldsymbol{x}, \boldsymbol{y}_A, \boldsymbol{y}_S)$ to the conservative approximation $\overline{\mathcal{P}}(\Xi_S)$ provides an implementable first-stage decision, it only provides implementable recourse decisions for the parameter realizations $\boldsymbol{\xi} \in \Xi_A \cup \Xi_S$. This is of no concern for most applications, where only the first-stage decision will be implemented. In some situations, however, an implementable recourse decision $\boldsymbol{y}(\boldsymbol{\xi})$ for the two-stage robust optimization problem $\mathcal{P}$ might be required here-and-now for every $\boldsymbol{\xi} \in \Xi$. This is frequently the case in real-time control applications, where there is not enough time to solve optimization problems to determine the recourse actions, as well as in embedded systems that lack the processing power or energy supply to solve optimization problems. In this section, we therefore elaborate how the second-stage decision $(\boldsymbol{y}_A, \boldsymbol{y}_S)$ of a feasible solution $(\boldsymbol{x}, \boldsymbol{y}_A, \boldsymbol{y}_S)$ to $\overline{\mathcal{P}}(\Xi_S)$ can be transformed into a decision rule $\boldsymbol{y} : \Xi \mapsto \mathbb{R}^{n_2}$ that prescribes implementable recourse decisions to $\mathcal{P}$ for all $\boldsymbol{\xi} \in \Xi$. Along the way, we will discover some insightful connections between our lifting scheme and the piecewise affine decision rules studied in [24, 25, 31, 32] and others.

We define the *graph* of a decision rule $\boldsymbol{y} : \Xi \mapsto \mathbb{R}^{n_2}$ as

$$\operatorname{gr} \boldsymbol{y} \;=\; \left\{ \begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{y}(\boldsymbol{\xi}) \end{pmatrix} \;:\; \boldsymbol{\xi} \in \Xi \right\}.$$

By construction, $\operatorname{gr} \boldsymbol{y}$ is uniquely specified through $\boldsymbol{y}$ and vice versa. We also define the *recourse set* of a second-stage decision $(\boldsymbol{y}_A, \boldsymbol{y}_S)$ in problem $\overline{\mathcal{P}}(\Xi_S)$ as

$$\operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S) \;=\; \operatorname{conv}\left(\left\{ \begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{y}_A(\boldsymbol{\xi}) \end{pmatrix} \;:\; \boldsymbol{\xi} \in \Xi_A \right\} \;\cup\; \left\{ \begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{y}_S(\boldsymbol{\xi}) \end{pmatrix} \;:\; \boldsymbol{\xi} \in \Xi_S \right\} \right).$$

To economize on notation, we omit the dependence of $\operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S)$ on Ξ_A and Ξ_S. Intuitively speaking, $\operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S)$ constitutes the convex hull of the graphs $\operatorname{gr} \boldsymbol{y}_A$ and $\operatorname{gr} \boldsymbol{y}_S$, restricted to their respective domains Ξ_A and Ξ_S. We now show that $\operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S)$ contains those recourse decisions $\boldsymbol{y}$ to the two-stage problem $\mathcal{P}$ that correspond to the second-stage decision $(\boldsymbol{y}_A, \boldsymbol{y}_S)$ in $\overline{\mathcal{P}}(\Xi_S)$.

Proposition 4. *If $(\boldsymbol{x}, \boldsymbol{y}_A, \boldsymbol{y}_S)$ is feasible in $\overline{\mathcal{P}}(\Xi_S)$, then for any decision rule $\boldsymbol{y} : \Xi \mapsto \mathbb{R}^{n_2}$ satisfying $\operatorname{gr} \boldsymbol{y} \subseteq \operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S)$, $(\boldsymbol{x}, \boldsymbol{y})$ is feasible in $\mathcal{P}$ and attains the same objective value as $(\boldsymbol{x}, \boldsymbol{y}_A, \boldsymbol{y}_S)$ in $\overline{\mathcal{P}}(\Xi_S)$.*

Proof of Proposition 4. It is clear that $(\boldsymbol{x}, \boldsymbol{y})$ attains the same objective value in $\mathcal{P}$ as $(\boldsymbol{x}, \boldsymbol{y}_A, \boldsymbol{y}_S)$ does in $\overline{\mathcal{P}}(\Xi_S)$. We now show that $(\boldsymbol{x}, \boldsymbol{y})$ is feasible in $\mathcal{P}$. One readily verifies that

$$\operatorname{ext}\operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S) \;=\; \left\{ \begin{pmatrix} \boldsymbol{\xi}' \\ \boldsymbol{y}_A(\boldsymbol{\xi}') \end{pmatrix} \;:\; \boldsymbol{\xi}' \in \operatorname{ext} \Xi_A \right\} \;\cup\; \left\{ \begin{pmatrix} \boldsymbol{\xi}' \\ \boldsymbol{y}_S(\boldsymbol{\xi}') \end{pmatrix} \;:\; \boldsymbol{\xi}' \in \Xi_S \right\},$$

and since $\operatorname{gr} \boldsymbol{y} \subseteq \operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S)$, by construction, for every $\boldsymbol{\xi} \in \Xi$ there is a function $\lambda : \operatorname{ext} \Xi \mapsto \mathbb{R}_+$ that satisfies $\sum_{\boldsymbol{\xi}' \in \operatorname{ext} \Xi} \lambda(\boldsymbol{\xi}') = 1$ and

$$\begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{y}(\boldsymbol{\xi}) \end{pmatrix} = \sum_{\boldsymbol{\xi}' \in \operatorname{ext} \Xi_A} \lambda(\boldsymbol{\xi}') \begin{pmatrix} \boldsymbol{\xi}' \\ \boldsymbol{y}_A(\boldsymbol{\xi}') \end{pmatrix} \;+\; \sum_{\boldsymbol{\xi}' \in \Xi_S} \lambda(\boldsymbol{\xi}') \begin{pmatrix} \boldsymbol{\xi}' \\ \boldsymbol{y}_S(\boldsymbol{\xi}') \end{pmatrix}. \tag{2}$$

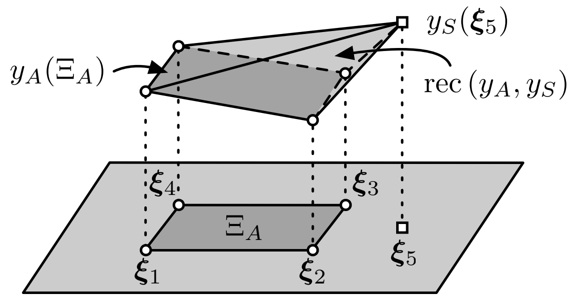

Figure 3. For $\Xi_A = \mathrm{conv}\,\{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_4\}$ and $\Xi_S = \{\boldsymbol{\xi}_5\}$ in the shaded plane, the recourse set $\mathrm{rec}\,(y_A, y_S)$ is shown as the light-shaded pyramid. Its darker shaded bottom represents $y_A(\Xi_A) = \{y_A(\boldsymbol{\xi}) : \boldsymbol{\xi} \in \Xi_A\}$.

We thus obtain that

$$
\begin{aligned}
&\boldsymbol{T}(\boldsymbol{\xi})\,\boldsymbol{x} + \boldsymbol{W}\boldsymbol{y}(\boldsymbol{\xi}) \\
&= \boldsymbol{T}\left(\sum_{\boldsymbol{\xi}' \in \mathrm{ext}\,\Xi_A} \lambda(\boldsymbol{\xi}')\,\boldsymbol{\xi}' + \sum_{\boldsymbol{\xi}' \in \Xi_S} \lambda(\boldsymbol{\xi}')\,\boldsymbol{\xi}'\right) \boldsymbol{x} + \boldsymbol{W}\boldsymbol{y}\left(\sum_{\boldsymbol{\xi}' \in \mathrm{ext}\,\Xi_A} \lambda(\boldsymbol{\xi}')\,\boldsymbol{\xi}' + \sum_{\boldsymbol{\xi}' \in \Xi_S} \lambda(\boldsymbol{\xi}')\,\boldsymbol{\xi}'\right) \\
&= \sum_{\boldsymbol{\xi}' \in \mathrm{ext}\,\Xi_A} \lambda(\boldsymbol{\xi}')\,\boldsymbol{T}(\boldsymbol{\xi}')\,\boldsymbol{x} + \sum_{\boldsymbol{\xi}' \in \Xi_S} \lambda(\boldsymbol{\xi}')\,\boldsymbol{T}(\boldsymbol{\xi}')\,\boldsymbol{x} + \boldsymbol{W}\left(\sum_{\boldsymbol{\xi}' \in \mathrm{ext}\,\Xi_A} \lambda(\boldsymbol{\xi}')\,\boldsymbol{y}_A(\boldsymbol{\xi}') + \sum_{\boldsymbol{\xi}' \in \Xi_S} \lambda(\boldsymbol{\xi}')\,\boldsymbol{y}_S(\boldsymbol{\xi}')\right) \\
&= \sum_{\boldsymbol{\xi}' \in \mathrm{ext}\,\Xi_A} \lambda(\boldsymbol{\xi}')\left[\boldsymbol{T}(\boldsymbol{\xi}')\,\boldsymbol{x} + \boldsymbol{W}\boldsymbol{y}_A(\boldsymbol{\xi}')\right] \;+\; \sum_{\boldsymbol{\xi}' \in \Xi_S} \lambda(\boldsymbol{\xi}')\left[\boldsymbol{T}(\boldsymbol{\xi}')\,\boldsymbol{x} + \boldsymbol{W}\boldsymbol{y}_S(\boldsymbol{\xi}')\right] \\
&\geq \sum_{\boldsymbol{\xi}' \in \mathrm{ext}\,\Xi_A} \lambda(\boldsymbol{\xi}')\,\boldsymbol{h}(\boldsymbol{\xi}') \;+\; \sum_{\boldsymbol{\xi}' \in \Xi_S} \lambda(\boldsymbol{\xi}')\,\boldsymbol{h}(\boldsymbol{\xi}') \;=\; \boldsymbol{h}\left(\sum_{\boldsymbol{\xi}' \in \mathrm{ext}\,\Xi_A} \lambda(\boldsymbol{\xi}')\,\boldsymbol{\xi}' + \sum_{\boldsymbol{\xi}' \in \Xi_S} \lambda(\boldsymbol{\xi}')\,\boldsymbol{\xi}'\right) \;=\; \boldsymbol{h}(\boldsymbol{\xi}),
\end{aligned}
$$

where the first, the second and the last identity are due to (2) and the fact that $\boldsymbol{T}$ and $\boldsymbol{h}$ are affine, and the inequality holds because of the feasibility of $(\boldsymbol{x}, \boldsymbol{y}_A, \boldsymbol{y}_S)$ in $\overline{\mathcal{P}}(\Xi_S)$. $\square$

Figure 3 illustrates the recourse set of a one-dimensional recourse decision. Interestingly, the reverse implication of Proposition 4 does not hold in general. In fact, the next example shows that there can be feasible solutions $(\boldsymbol{x}, \boldsymbol{y})$ to problem $\mathcal{P}$ for which $\mathrm{gr}\,\boldsymbol{y} \not\subseteq \mathrm{rec}\,(\boldsymbol{y}_A, \boldsymbol{y}_S)$ for every feasible solution $(\boldsymbol{x}, \boldsymbol{y}_A, \boldsymbol{y}_S)$ to every conservative approximation $\overline{\mathcal{P}}(\Xi_S)$ of $\mathcal{P}$.

Example 2. *Consider the two-stage robust optimization problem*

$$\begin{array}{lll} \text{minimize} & x & \\ \text{subject to} & x \geq y(\xi) & \multirow{2}{*}{} \\ & y(\xi) \geq \xi, \;\; y(\xi) \geq -\xi & \end{array} \left.\begin{array}{l} \\ \\ \end{array}\right\} \; \forall \xi \in [-1, 1]. \tag{3}$$

For every scenario set $\Xi_S \subseteq \{-1, 1\}$*, the unique optimal solution* (x, y_A, y_S) *to* $\overline{\mathcal{P}}(\Xi_S)$ *satisfies* $x = 1$ *and* $\operatorname{rec}(y_A, y_S) = [-1, 1] \times \{1\}$*. Problem* (3) *is optimized, however, by* $x = 1$ *and every decision rule* $y : [-1, 1] \mapsto \mathbb{R}_+$ *satisfying* $y(\xi) \in [|\xi|, 1]$*.* □

The question naturally arises which decision rule from within $\operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S)$ we should select. A subclass of decision rules that are of special interest are *piecewise affine decision rules* $\boldsymbol{y} : \Xi \mapsto \mathbb{R}^{n_2}$ for which there is a partition of Ξ into finitely many polyhedra $\Xi_1, \ldots, \Xi_L$ such that $\boldsymbol{y}$ is affine on each Ξ_ℓ, $\ell = 1, \ldots, L$. It turns out that piecewise affine decision rules are intimately related to *simplicial decompositions* of Ξ, which are subdivisions of Ξ into finitely many k-dimensional simplices that only intersect at their boundaries and whose union recovers Ξ.

Proposition 5. *Consider a scenario set* $\Xi_S \subseteq \operatorname{ext} \Xi$ *as well as the second-stage decisions* $\boldsymbol{y}_A : \Xi_A \mapsto \mathbb{R}^{n_2}$ *and* $\boldsymbol{y}_S : \Xi_S \mapsto \mathbb{R}^{n_2}$*. Then:*

(i) For every piecewise affine decision rule $\boldsymbol{y} : \Xi \mapsto \mathbb{R}^{n_2}$ *with* $\operatorname{gr} \boldsymbol{y} \in \operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S)$*, there is a simplicial decomposition* $\{\Xi_\ell\}_\ell$ *of* Ξ *such that* $\boldsymbol{y}$ *is affine on every simplex* Ξ_ℓ*.*

(ii) For every simplicial decomposition $\{\Xi_\ell\}_\ell$ *of* Ξ*, there is a decision rule* $\boldsymbol{y} : \Xi \mapsto \mathbb{R}^{n_2}$ *with* $\operatorname{gr} \boldsymbol{y} \in \operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S)$ *such that* $\boldsymbol{y}$ *affine on every simplex* Ξ_ℓ*.*

Proof of Proposition 5. As for the first statement, there is a partition $\{\hat{\Xi}_\ell\}_\ell$ of Ξ into polyhedra such that $\boldsymbol{y}$ is affine over each $\hat{\Xi}_\ell$. Fix a simplicial decomposition $\{\Xi_{\ell,\ell'}\}_{\ell'}$ for each polyhedron $\hat{\Xi}_\ell$. Then $\boldsymbol{y}$ is also affine over the simplicial decomposition $\{\Xi_{\ell,\ell'}\}_{\ell,\ell'}$ of Ξ.

In view of the second statement, fix any simplicial decomposition $\{\Xi_\ell\}_\ell$ of Ξ, as well as

$$\boldsymbol{y}(\boldsymbol{\xi}') \in \left\{ \boldsymbol{y} \in \mathbb{R}^{n_2} \; : \; \begin{pmatrix} \boldsymbol{\xi}' \\ \boldsymbol{y} \end{pmatrix} \in \operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S) \right\} \qquad \text{for } \boldsymbol{\xi}' \in \bigcup_\ell \operatorname{ext} \Xi_\ell, \tag{4}$$

where $\bigcup_\ell \operatorname{ext} \Xi_\ell$ constitutes the set of all corner points of the simplices Ξ_ℓ in the simplicial decomposition. For each simplex Ξ_ℓ, we set $\boldsymbol{y}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}' \in \operatorname{ext} \Xi_\ell} \lambda(\boldsymbol{\xi}; \boldsymbol{\xi}') \cdot \boldsymbol{y}(\boldsymbol{\xi}')$ for all remaining points

$\boldsymbol{\xi} \in \Xi_\ell \setminus (\operatorname{ext} \Xi_\ell)$, where $\lambda : \Xi_\ell \times \operatorname{ext} \Xi_\ell \mapsto \mathbb{R}_+$ is the unique weighting function that satisfies

$$\sum_{\boldsymbol{\xi}' \in \operatorname{ext} \Xi_\ell} \lambda(\boldsymbol{\xi}; \boldsymbol{\xi}') = 1 \qquad \text{and} \qquad \sum_{\boldsymbol{\xi}' \in \operatorname{ext} \Xi_\ell} \lambda(\boldsymbol{\xi}; \boldsymbol{\xi}') \cdot \boldsymbol{\xi}' = \boldsymbol{\xi} \qquad \forall \boldsymbol{\xi} \in \Xi_\ell,$$

that is, $\lambda(\boldsymbol{\xi}; \cdot)$ are the barycentric coordinates of $\boldsymbol{\xi}$ in Ξ_ℓ. The statement now follows if

$$\begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{y}(\boldsymbol{\xi}) \end{pmatrix} \in \operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S) \qquad \forall \boldsymbol{\xi} \in \Xi_\ell.$$

Due to (4), this holds for $\boldsymbol{\xi} \in \operatorname{ext} \Xi_\ell$. Moreover, for $\boldsymbol{\xi} \in \Xi_\ell \setminus (\operatorname{ext} \Xi_\ell)$ we have

$$\begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{y}(\boldsymbol{\xi}) \end{pmatrix} = \sum_{\boldsymbol{\xi}' \in \operatorname{ext} \Xi_\ell} \lambda(\boldsymbol{\xi}; \boldsymbol{\xi}') \cdot \begin{pmatrix} \boldsymbol{\xi}' \\ \boldsymbol{y}(\boldsymbol{\xi}') \end{pmatrix} \in \operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S),$$

where the membership follows from (4) and the convexity of $\operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S)$. Since the simplex Ξ_ℓ was chosen arbitrarily, we thus conclude that $\operatorname{gr} \boldsymbol{y} \subseteq \operatorname{rec}(\boldsymbol{y}_A, \boldsymbol{y}_S)$. □

Propositions 4 and 5 allow us to complete any feasible solution $(\boldsymbol{x}, \boldsymbol{y}_A, \boldsymbol{y}_S)$ in problem $\overline{\mathcal{P}}(\Xi_S)$ to a feasible solution $(\boldsymbol{x}, \boldsymbol{y})$ in problem $\mathcal{P}$ that attains the same objective value and that is piecewise affine over any fixed simplicial decomposition $\{\Xi_\ell\}_\ell$ of Ξ. To construct a piecewise affine decision rule with a compact description, we propose to combine the affine decision rule $\boldsymbol{y}_A$ over Ξ_A with a piecewise affine decision rule over $\Xi \setminus \Xi_A$ that is affine on every simplex of a simplicial decomposition of $\operatorname{cl}(\Xi \setminus \Xi_A)$. Note that $\operatorname{cl}(\Xi \setminus \Xi_A) = \operatorname{conv}\big(\Xi_S \cup \bigcup \{N(\boldsymbol{\xi}) \,:\, \boldsymbol{\xi} \in \Xi_S\}\big)$, where $N(\boldsymbol{\xi})$ denotes the neighbouring extreme points of $\boldsymbol{\xi}$. A simplicial decomposition of $\operatorname{cl}(\Xi \setminus \Xi_A)$ can be found with standard triangulation schemes [23].

5 Numerical Experiments

We now analyze the computational performance of our primal-dual lifting scheme from Section 3 in the context of two illustrative examples (Section 5.1) as well as an inventory management problem (Section 5.2). In our experiments, we will assess the scalability of our algorithm in terms of the problem size (measured by the number of decision variables and constraints) and the number of uncertain problem parameters. We will also investigate to which degree the different components

	Technique	Convergence	Recourse	Uncertainty Sets	Subproblems
Primal-Dual	lifting of uncertainty set	finite	generic	specific classes	convex hulls & LPs
FME & SB	elimination of second-stage decisions	finite	generic†	generic polyhedra	elimination & LPs
Zeng & Zhao	column-and-constraint generation	finite	rel. compl.	generic polyhedra	MILPs
ADR & SB	(piecewise) affine decision rules	none	generic†	generic polyhedra	LPs

† with the exception of [12] which requires relatively complete recourse.

Table 1. Comparison of our method ('Primal-Dual') with the Fourier-Motzkin elimination approach proposed by Zhen et al. [63] ('FME & SB'), the column-and-constraint generation scheme due to Zeng and Zhao [61] ('Zeng & Zhao') as well as the piecewise linear decision rules [9, 31, 43] combined with the progressive bounds provided by Hadjiyiannis et al. [37] or Bertsimas and de Ruiter [12] ('ADR & SB').

of our algorithm add to the algorithm's runtime.

A secondary objective of this section is to compare our algorithm with some of the state-of-the-art solution approaches for two-stage robust optimization problems. To this end, we will compare our solution scheme with the conservative bounds offered by linear and piecewise linear decision rules [9, 31, 43], the progressive bounds provided by Hadjiyiannis et al. [37] and Bertsimas and de Ruiter [12], the column-and-constraint generation scheme due to Zeng and Zhao [61] as well as the Fourier-Motzkin elimination approach proposed by Zhen et al. [63]. Table 1 compares the guarantees offered by these approaches, as well as their underlying assumptions. While the pitfalls of drawing conclusions from a small test set are well-known, we hope to generate some insights into the intricate relationship between the characteristics of problem $\mathcal{P}$ and the suitability of the different solution schemes.

All optimization problems in this section were solved in single-threaded mode with the Gurobi 7.5 optimization package (see `http://www.gurobi.com`) on a 2.9GHz computer with 8GB RAM. Our C++ implementation of the primal-dual lifting scheme uses the LRS package for vertex and facet enumeration (see `http://cgm.cs.mcgill.ca/~avis/C/lrs.html`).

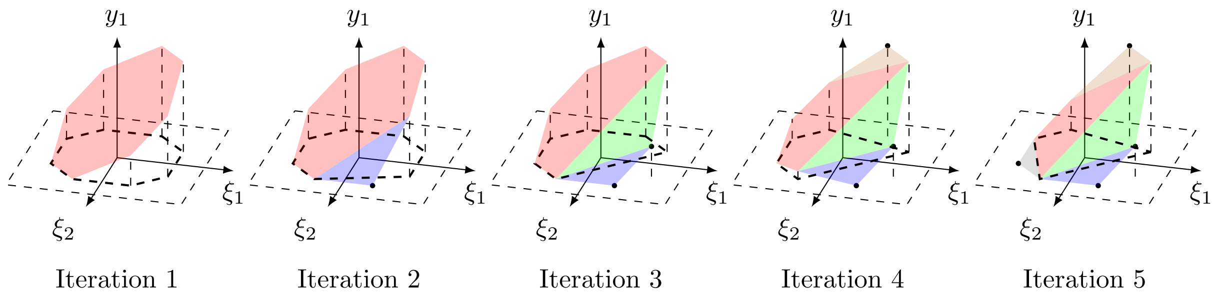

Figure 4. Illustration of the piecewise affine decision rules $y_1(\boldsymbol{\xi})$ corresponding to the optimal primal solutions $(\boldsymbol{x}, \boldsymbol{y}_A, \boldsymbol{y}_S)$ in every iteration of our lifting scheme. From left to right, the optimal objective values of $\overline{\mathcal{P}}(\emptyset)$ are 6, 6, 5, 4.86 and 4.

5.1 Illustrative Examples

We adapt two examples from the literature to illustrate the structure of the piecewise affine decision rules of our lifting approach (Section 5.1.1), and we show how the affine decision rules $\boldsymbol{\lambda}_A$ in the dual problem $\underline{\mathcal{P}}(\Xi_S)$ can contribute to the tightness of the lower bounds (Section 5.1.2).

5.1.1 Worst-Case Value of a Sum-of-Max Function

We consider the following adaptation of the example (TOY2) from [33]:

$$\begin{array}{ll} \text{minimize} & \tau \\ \text{subject to} & \left.\begin{array}{l} \tau \geq \mathbf{e}^\top \boldsymbol{y}(\boldsymbol{\xi}) \\ y_1(\boldsymbol{\xi}) \geq \max\{x, x + \xi_1 + \xi_2\} \\ y_2(\boldsymbol{\xi}) \geq \max\{x, x + \xi_1 - \xi_2\} \\ y_3(\boldsymbol{\xi}) \geq \max\{x, x - \xi_1 + \xi_2\} \\ y_4(\boldsymbol{\xi}) \geq \max\{x, x - \xi_1 - \xi_2\} \end{array}\right\} \ \forall \boldsymbol{\xi} \in \Xi \\ & \tau, x \in \mathbb{R}_+, \quad \boldsymbol{y} : \Xi \mapsto \mathbb{R}^4 \end{array}$$

In this problem, we set the uncertainty set to $\Xi = \{\boldsymbol{\xi} \in [-2, 2]^2 \, : \, \|\boldsymbol{\xi}\|_1 \leq 3\}$. The problem is optimized by the here-and-now decisions $(\tau^\star, x^\star) = (4, 0)$.

The uncertainty set Ξ satisfies $|\text{ext}\, \Xi| = 8$. If we lift one of the binding scenarios $\boldsymbol{\xi}^\star \in \Xi_A^\star$ randomly in each iteration, then our lifting scheme converges after 5 iterations. Figure 4 visualizes the piecewise affine decision rules $y_1(\boldsymbol{\xi})$ corresponding to the optimal primal solutions $(\boldsymbol{x}, \boldsymbol{y}_A, \boldsymbol{y}_S)$ in every iteration of our lifting scheme. The initial dual bound $\underline{\mathcal{P}}(\emptyset)$ has an optimal objective value of

3.33, and the dual bound attains the optimal objective value 4 of the two-stage robust optimization problem after lifting any single extreme point $\boldsymbol{\xi} \in \operatorname{ext} \Xi_A$.

5.1.2 Worst-Case Makespan of a Temporal Network

We now employ the two-stage robust optimization problem $\mathcal{P}$ to estimate the worst-case makespan of a temporal network (*e.g.*, a project, a digital circuit or a production process). To this end, we define a temporal network as a directed, acyclic graph $G = (V, E)$ whose nodes $V = \{1, \ldots, n\}$ represent the tasks and whose arcs $E \subset V \times V$ denote the temporal precedences between the tasks. We assume that the duration $d_i(\boldsymbol{\xi})$ of each task $i \in V$ depends on the uncertain problem parameters $\boldsymbol{\xi} \in \Xi$. Moreover, we assume that the start times of the tasks $i \in V_0 \subseteq V$ have to be chosen here-and-now, that is, before the realizations of the uncertain parameters $\boldsymbol{\xi}$ are observed. This could be required to synchronize the network with other projects, circuits or production processes.

The problem is a variant of model (1) in [58], and it can be formulated as

$$
\begin{array}{ll}
\text{minimize} & \tau \\
\text{subject to} & \left.\begin{array}{ll} \tau \geq y_n(\boldsymbol{\xi}) + d_n(\boldsymbol{\xi}) & \\ y_j(\boldsymbol{\xi}) \geq y_i(\boldsymbol{\xi}) + d_i(\boldsymbol{\xi}) & \forall (i,j) \in E \\ y_i(\boldsymbol{\xi}) = x_i & \forall i \in V_0 \end{array}\right\} \ \forall \boldsymbol{\xi} \in \Xi \\
& \tau \in \mathbb{R}_+, \ \ x_i \in \mathbb{R}_+, \, i \in V_0, \ \ \boldsymbol{y} : \Xi \mapsto \mathbb{R}^n_+.
\end{array}
$$

In this formulation, the wait-and-see decisions $y_i(\boldsymbol{\xi})$ capture the start times of the tasks $i \in V$, the epigraphical here-and-now decision τ records the worst-case makespan of the network, and the here-and-now decisions x_i correspond to the static start times of the restricted tasks $i \in V_0$. The presence of the restricted tasks $i \in V_0$ implies that the problem has no relatively complete recourse.

We apply our primal-dual lifting scheme to the temporal network in Figure 5. We also present a variant of our dual bound $\underline{\mathcal{P}}(\Xi_S)$ where we fix all affine decisions to $\boldsymbol{\lambda}_A(\boldsymbol{\xi}) = \mathbf{0}$ for all $\boldsymbol{\xi} \in \Xi_A$. This variant, which we refer to as 'sampling bound', can be interpreted as an extension of the sampling bound in [37] that iteratively grows the scenario set Ξ_S (see Remark 3). The results of both methods are shown in Figure 6. Our approach converges to the optimal objective value of the two-stage robust optimization problem $\mathcal{P}$ after 6 out of $|\operatorname{ext} \Xi| = 18$ possible iterations. The less flexible sampling bound requires 9 iterations to converge in this example.

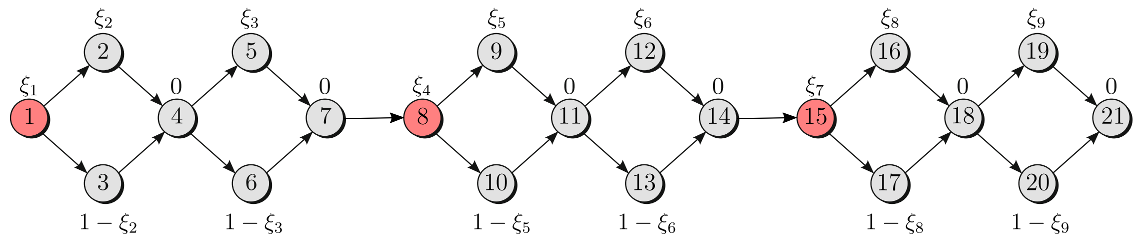

Figure 5. Temporal network with 21 tasks and 26 precedences. The durations $d_i(\boldsymbol{\xi})$ are displayed above the tasks $i \in V$, and the uncertainty set is $\Xi = \{\boldsymbol{\xi} \in \mathbb{R}^9 : \|\boldsymbol{\xi} - \mathbf{e}/2\|_1 \leq 1/2\}$. The start times of the tasks 1, 8 and 15 have to be selected here-and-now.

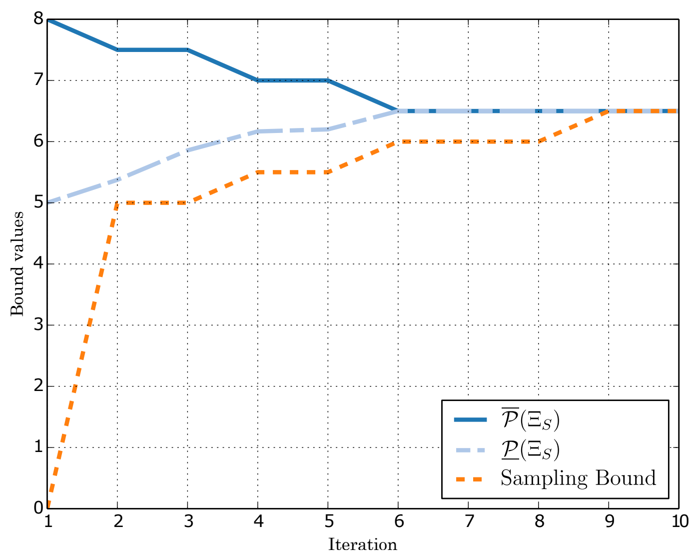

Figure 6. Results of our primal-dual lifting scheme (top two blue lines) and the sampling bound (bottom orange line) applied to the temporal network in Figure 5.

5.2 Case Study: Inventory Management

We consider an inventory management problem with n products whose demands $D_i(\boldsymbol{\xi})$, $i = 1, \ldots, n$, are governed by the uncertain risk factors $\boldsymbol{\xi} \in \mathbb{R}^k$. The demand for product i can be served through a standard order x_i (with unit cost c_x), which has to be placed before the demand is known, or through an express order y_i (with unit cost $c_y > c_x$), which can be submitted after the demand has been observed. Any excess inventory $h_i(\boldsymbol{\xi})$ and any backlogged demand $b_i(\boldsymbol{\xi})$ in the second period incurs unit costs of c_h and c_b, respectively, and the express orders across all products must not exceed B units. The objective is to determine an ordering policy that minimizes the worst-case sum of the ordering, inventory holding and backlogging costs over all anticipated demand realizations.

The problem can be formulated as the following instance of problem $\mathcal{P}$:

$$\begin{array}{ll} \text{minimize} & \tau \\ \text{subject to} & \left.\begin{array}{l} \tau \geq \mathbf{e}^\top \left[c_x \cdot \boldsymbol{x} + c_y \cdot \boldsymbol{y}(\boldsymbol{\xi}) + c_h \cdot \boldsymbol{h}(\boldsymbol{\xi}) + c_b \cdot \boldsymbol{b}(\boldsymbol{\xi}) \right] \\ \boldsymbol{h}(\boldsymbol{\xi}) \geq \boldsymbol{x} + \boldsymbol{y}(\boldsymbol{\xi}) - \boldsymbol{D}(\boldsymbol{\xi}) \\ \boldsymbol{b}(\boldsymbol{\xi}) \geq \boldsymbol{D}(\boldsymbol{\xi}) - \boldsymbol{x} - \boldsymbol{y}(\boldsymbol{\xi}) \\ \mathbf{e}^\top \boldsymbol{y}(\boldsymbol{\xi}) \leq B \end{array} \right\} \forall \boldsymbol{\xi} \in \Xi \\ & \boldsymbol{x} \in \mathbb{R}^n_+, \quad \boldsymbol{y}, \boldsymbol{h}, \boldsymbol{b} : \Xi \mapsto \mathbb{R}^n_+ \end{array} \tag{5}$$

Here, the epigraphical variable τ records the worst-case costs over all demand realizations. Although the existence of this first-stage epigraphical variable implies that problem (5) lacks a relatively complete recourse, we emphasize that an equivalent min-max formulation of the problem (as employed by the column-and-constraint generation scheme) enjoys a relatively complete recourse. Note also that we model the second-stage decisions $\boldsymbol{y}$, $\boldsymbol{h}$ and $\boldsymbol{b}$ as functions of the (typically unobservable) risk factors, as opposed to functions of the (eventually observable) product demands $\boldsymbol{D}(\boldsymbol{\xi})$. This simplification is justified since by construction, no optimal solution to problem (5) will take different second-stage decisions for different realizations of the risk factors $\boldsymbol{\xi}$ that give rise to the same product demands $\boldsymbol{D}(\boldsymbol{\xi})$. Thus, we can always convert the optimal second-stage decisions in problem (5) to equivalent implementable decisions that only depend on the product demands $\boldsymbol{D}(\boldsymbol{\xi})$.

We assume that the n products are grouped into $\lfloor \sqrt{n} \rfloor$ different product categories such that each category contains between $\lfloor \sqrt{n} \rfloor$ and $\lceil \sqrt{n} \rceil + 1$ products. The product demands are governed by a factor model of the form

$$D_i(\boldsymbol{\xi}) = \boldsymbol{\phi}_i^\top \boldsymbol{\xi} + \varphi_i, \; i = 1, \ldots, n, \quad \text{with} \quad \boldsymbol{\xi} \in \Xi = [-1, 1]^k.$$

For each product $i = 1, \ldots, n$, we choose the factor loading vector $\boldsymbol{\phi}_i \in \mathbb{R}^k$ uniformly at random from $[-1, 1]^k$ and subsequently scale it so that $\|\boldsymbol{\phi}_i\|_1 = 1$. We then set $\varphi_i = \|\boldsymbol{\phi}_i\|_1$. This parameter choice implies that $\{\boldsymbol{\phi}_i^\top \boldsymbol{\xi} + \varphi_i \, : \, \boldsymbol{\xi} \in \Xi\} = [0, 2]$. Moreover, we ensure that the factor loading vectors $\boldsymbol{\phi}_i$, $\boldsymbol{\phi}_j$ associated with two products i, j of the same category have the same signs (but typically not the same values) for each component, that is, $\operatorname{sgn}(\phi_{il}) = \operatorname{sgn}(\phi_{jl})$ for all $l = 1, \ldots, k$. This expresses the assumption that the demands for products of the same category are positively correlated. We fix $c_x = 0$ and select c_y uniformly at random from $[0, 2]$. Thus, we interpret the standard orders

	Primal-Dual			FME & SB			Zeng & Zhao			ADR & SB
Instance	10 min	1 hour	6 hours	10 min	1 hour	6 hours	10 min	1 hour	6 hours	
40-20	10.5%	9.8%	5.6%	14.0%	9.5%	9.5%	63.9%	1.3%	0.9%	21.0%
40-30	18.5%	17.0%	16.3%	21.4%	20.7%	20.7%	76.0%	2.3%	1.8%	26.5%
40-40	16.8%	14.8%	9.9%	24.7%	20.7%	20.7%	64.4%	2.5%	2.3%	27.4%
60-20	12.0%	11.3%	10.3%	19.1%	16.1%	16.1%	93.6%	2.3%	2.2%	20.0%
60-30	16.2%	16.0%	14.6%	21.6%	18.7%	18.7%	82.7%	24.3%	2.4%	23.5%
60-40	17.4%	15.4%	15.0%	21.2%	18.0%	18.0%	92.7%	42.3%	2.6%	21.2%

Table 2. Inventory Management with Relatively Complete Recourse: Optimality gaps of our method ('Primal-Dual'), the Fourier-Motzkin elimination proposed by Zhen et al. [63] ('FME & SB'), the column-and-constraint generation of Zeng and Zhao [61] ('Zeng & Zhao') as well as the affine decision rule approximation with the sampling bound of Hadjiyiannis et al. [37] ('ADR & SB') for various time limits and instance classes. We do not show the runtimes for the affine decision rule approximation as the results were obtained within 1 minute for all instances.

as sunk costs, and we aim to minimize the amount of express deliveries, which carry a per-unit premium of c_y over the standard orders. We select the inventory holding costs uniformly at random from $[3, 5]$, whereas the backlogging costs are chosen uniformly at random from $[0, 1/2]$. The upper bound on the express orders $\boldsymbol{y}(\boldsymbol{\xi})$ is set to $B = n$, which implies that up to 50% of the maximum demand can be covered via express deliveries.

To select the binding scenarios $\boldsymbol{\xi}^\star \in \Xi_A^\star$ to lift in each iteration of our primal-dual lifting scheme, we fix the optimal solution to the bounding problem $\overline{\mathcal{P}}(\Xi_S)$ and determine one binding scenario for each constraint by minimizing the constraint's slack over all $\boldsymbol{\xi} \in \Xi_A$. We then tentatively lift each such scenario $\boldsymbol{\xi}^\star$ and evaluate its impact on the upper bounding problem. Afterwards, instead of lifting a single scenario in each iteration (as described in our algorithm outline in Section 3), we lift every scenario that improves the upper bound (see also our discussion in Remark 5). Our modification leads to an algorithm that performs fewer iterations than the one outlined in Section 3, but it is able to better close the gap for larger instances.

We first compare the scalability of our primal-dual lifting scheme with the other approaches from Table 1 as the number of products and the number of uncertain parameters vary. The results are presented in Table 2. Each instance class is identified by the label 'n-k', where n denotes the number of products and k refers to the number of uncertain parameters, respectively. Here and in the following, the optimality gaps are computed relative to the mean value of *(i)* the smallest upper bound and *(ii)* the largest lower bound generated by any of the solution approaches listed in the table after six hours runtime. All values are averages over 25 randomly generated instances.

	Upper bounds	Lower bounds	Neighbours	Convex Hulls	Remainder
40-20	28.8%	0.3%	9.4%	58.7%	2.7%
40-30	31.9%	0.3%	15.7%	48.5%	3.5%
40-40	43.2%	0.9%	17.9%	34.8%	3.2%
60-20	45.1%	0.5%	14.1%	37.0%	3.3%
60-30	61.0%	0.8%	16.5%	19.4%	2.3%
60-40	69.6%	0.9%	19.6%	7.7%	2.2%

Table 3. Inventory Management with Relatively Complete Recourse: Percentages of the overall runtime of our algorithm that are spent in each iteration on solving the upper bound problems $\overline{\mathcal{P}}(\Xi_S)$ ('Upper bounds'), solving the lower bound problems $\underline{\mathcal{P}}(\Xi_S)$ ('Lower bounds'), finding the neighbours of the extreme points $\boldsymbol{\xi}^\star \in \Xi_A^\star$ to be lifted ('Neighbours'), calculating the convex hulls $\mathrm{conv}\left([\mathrm{ext}\, \Xi_A] \setminus \{\boldsymbol{\xi}^\star\}\right)$ ('Convex Hulls') and the remaining overhead ('Remainder').

Table 2 shows that the column-and-constraint generation scheme of Zeng and Zhao [61] clearly outperforms the other three methods.[1] Our primal-dual lifting scheme is able to close between 29.2% (instance 60-40) and 73.3% (instance 40-20) of the optimality gap of the affine decision rules. The Fourier-Motzkin elimination, on the other hand, only closes between 15.1% (instance 60-40) and 54.8% (instance 40-20) of the optimality gap of the affine decision rules. A closer inspection of the algorithms revealed that the majority of the runtime of the column-and-constraint generation scheme is spent on the mixed-integer linear programming subproblems, which do not grow in size and therefore guarantee a steady progress of the overall procedure. In contrast, both the scenario selection and the lifting steps of our primal-dual approach require considerable time, and the bounding problems tend to grow larger over time due to the increased complexity of the convex hulls of the remaining extreme points Ξ_A. The Fourier-Motzkin elimination scheme, finally, suffers from a rapid growth of the number of constraints, and the constraint elimination procedure outlined in Section 4 of [63] was only able to identify a small fraction of the constraints as redundant.

We have also solved the inventory management problem with the piecewise affine decision rules proposed in [31]. We have not been able to obtain any noticeable improvements relative to the results of the affine decision rules. Moreover, we have attempted to replace the sampling bound of [37] with the refined bound proposed in [12]. This has not led to significant improvements either. Finally, we have tried to solve the instances monolithically with the extreme point reformulation $\mathcal{P}'$. For the given time limit of six hours, this reformulation could not be solved for any of the instances.

[1] Our implementation of the column-and-constraint generation scheme uses a time limit of 10 minutes for the solution of each subproblem. While this ensures that the algorithm does not get stuck in the attempt to solve one of the subproblems, it implies that the overall scheme will typically not obtain obtain optimal solutions.

Table 3 shows the fractions of the runtime that are spent on the different steps of our primal-dual lifting scheme. We observe that the majority of the runtime is spent on *(i)* solving the upper bound problems $\overline{\mathcal{P}}(\Xi_S)$ and *(ii)* calculating the convex hulls $\operatorname{conv}([\operatorname{ext}\Xi_A] \setminus \{\boldsymbol{\xi}^\star\})$. This is not surprising since in each iteration, our algorithm tentatively lifts every binding scenario $\boldsymbol{\xi}^\star \in \Xi_A^\star$ and evaluates its impact on the upper bounding problem, thus requiring the solution of many upper bounding problems as well as the calculation of many convex hulls. This approach trades off the conflicting goals of lifting sufficiently many extreme points in each iteration (and thus not wasting useful computation) and lifting only those extreme points that improve the bounds (and thus not unduly growing the representations of the bounding problems $\overline{\mathcal{P}}(\Xi_S)$ and $\underline{\mathcal{P}}(\Xi_S)$).

We now consider a variant of the inventory management problem (5) that additionally imposes an upper bound I on the number of products $\mathbf{e}^\top \boldsymbol{h}(\boldsymbol{\xi})$ held in inventory:

$$
\begin{array}{ll}
\text{minimize} & \tau \\
\text{subject to} & \left.\begin{array}{l}
\tau \geq \mathbf{e}^\top \left[c_x \cdot \boldsymbol{x} + c_y \cdot \boldsymbol{y}(\boldsymbol{\xi}) + c_h \cdot \boldsymbol{h}(\boldsymbol{\xi}) + c_b \cdot \boldsymbol{b}(\boldsymbol{\xi})\right] \\
\boldsymbol{h}(\boldsymbol{\xi}) \geq \boldsymbol{x} + \boldsymbol{y}(\boldsymbol{\xi}) - \boldsymbol{D}(\boldsymbol{\xi}) \\
\boldsymbol{b}(\boldsymbol{\xi}) \geq \boldsymbol{D}(\boldsymbol{\xi}) - \boldsymbol{x} - \boldsymbol{y}(\boldsymbol{\xi}) \\
\mathbf{e}^\top \boldsymbol{y}(\boldsymbol{\xi}) \leq B \\
\mathbf{e}^\top \boldsymbol{h}(\boldsymbol{\xi}) \leq I
\end{array}\right\} \quad \forall \boldsymbol{\xi} \in \Xi \\
& \boldsymbol{x} \in \mathbb{R}^n_+, \quad \boldsymbol{y}, \boldsymbol{h}, \boldsymbol{b} : \Xi \mapsto \mathbb{R}^n_+
\end{array}
\tag{6}
$$

In our experiments we choose the inventory bound $I = n/2$, which implies that up to 25% of the maximum demand can be held in inventory. We emphasize that even the equivalent min-max formulation of this problem (as employed by the column-and-constraint generation scheme) lacks a relatively complete recourse: Excessive first-stage orders $\boldsymbol{x}$ that outstrip the demand by more than I units, that is, first-stage orders satisfying $\mathbf{e}^\top [\boldsymbol{x} - \boldsymbol{D}(\boldsymbol{\xi})]_+ > I$, lead to an infeasible second stage.

Table 4 compares the scalability of our primal-dual lifting scheme with the other approaches from Table 1 for the problem (6). Interestingly, the column-and-constraint generation scheme is now outperformed by all of the other approaches. A closer inspection of the results revealed that the column-and-constraint generation scheme, which now requires the solution of both feasibility and optimality subproblems, typically gets stuck at an early iteration. Indeed, contrary to the optimality subproblems, which can be terminated early as long as a feasible (but possibly suboptimal) solution

	Primal-Dual			FME & SB			Zeng & Zhao			ADR & SB
Instance	10 min	1 hour	6 hours	10 min	1 hour	6 hours	10 min	1 hour	6 hours	
40-20	11.6%	9.4%	7.9%	12.5%	12.5%	12.5%	118.4%	118.4%	118.4%	21.7%
40-30	20.3%	18.1%	17.6%	23.5%	22.8%	22.8%	125.1%	125.1%	125.1%	29.4%
40-40	14.9%	14.8%	13.0%	23.8%	21.9%	21.9%	118.7%	118.7%	118.7%	26.9%
60-20	13.7%	13.1%	11.7%	19.1%	16.2%	16.2%	120.5%	120.5%	120.5%	19.7%
60-30	19.6%	18.0%	17.8%	21.8%	18.9%	18.9%	119.2%	119.2%	119.2%	22.7%
60-40	15.4%	14.9%	11.9%	18.9%	18.3%	18.3%	118.6%	118.6%	118.6%	20.7%

Table 4. Inventory Management without Relatively Complete Recourse: The columns and table entries have the same interpretation as in Table 2.

has been determined, the feasibility subproblems need to be solved until either the lower bound strictly exceeds zero (indicating that the current first-stage decision is infeasible, and producing a violated realization of the uncertain parameter vector $\boldsymbol{\xi}$) or until the upper bound equals zero (in which case the feasibility subproblem has been solved to optimality, indicating that the current first-stage decision is feasible). It turns out that the solution of these feasibility subproblems becomes very challenging, and the algorithm typically gets stuck at an early iteration where it neither produces a violated realization of $\boldsymbol{\xi}$ nor confirms the feasibility of the candidate first-stage decision $\boldsymbol{x}$. In contrast, neither our primal-dual lifting scheme nor the Fourier-Motzkin elimination are significantly affected by the lack of a relatively complete recourse: The primal-dual lifting scheme closes between 21.6% (instance 60-30) and 63.6% (instance 40-20) of the optimality gap of the affine decision rules, and the Fourier-Motzkin elimination closes between 11.6% (instance 60-40) and 42.4% (instance 40-20) of the optimality gap of the affine decision rules.

Acknowledgments

The authors gratefully acknowledge funding from the Imperial Business Analytics Centre and the EPSRC grants EP/M027856/1, EP/M028240/1 and EP/N020030/1. We are also indebted to David Avis and Komei Fukuda for valuable advice on various aspects of polyhedral combinatorics, as well as the constructive suggestions of the three anonymous referees.

References

[1] Y. An, B. Zeng, Y. Zhang, and L. Zhao. Reliable p-median facility location problem: Two-stage robust models and algorithms. *Transportation Research Part B: Methodological*, 64:54–72, 2014.

[2] B. D. O. Anderson and J. B. Moore. *Optimal Control: Linear Quadratic Methods.* Dover, 1990.

[3] G. Angulo, S. Ahmed, S. S. Dey, and V. Kaibel. Forbidden vertices. *Mathematics of Operations Research*, 40(2):350–360, 2015.

[4] A. Ardestani-Jaafari and E. Delage. Robust optimization of sums of piecewise linear functions with application to inventory problems. *Operations Research*, 64(2):474–494, 2016.

[5] A. Atamtürk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, 2007.

[6] J. Ayoub and M. Poss. Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Computational Management Science*, 13(2):219–239, 2016.

[7] C. Bandi and D. Bertsimas. Tractable stochastic analysis in high dimensions via robust optimization. *Mathematical Programming*, 134(1):23–70, 2012.

[8] A. Ben-Tal, B. Golany, A. Nemirovski, and J.-P. Vial. Retailer-supplier flexible commitments contracts: A robust optimization approach. *Manufacturing & Service Operations Management*, 7(3):248–271, 2005.

[9] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.

[10] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[11] D. Bertsimas and C. Caramanis. Finite adaptibility in multistage linear optimization. *IEEE Transactions on Automatic Control*, 55(12):2751–2766, 2010.

[12] D. Bertsimas and F. J. C. T. de Ruiter. Duality in two-stage adaptive linear optimization: Faster computation and stronger bounds. *INFORMS Journal on Computing*, 28(3):500–511, 2016.

[13] D. Bertsimas and I. Dunning. Multistage robust mixed-integer optimization with adaptive partitions. *Operations Research*, 64(4):980–998, 2016.

[14] D. Bertsimas and A. Georghiou. Design of near optimal decision rules in multistage adaptive mixed-integer optimization. *Operations Research*, 63(3):610–627, 2015.

[15] D. Bertsimas and V. Goyal. On the power of robust solutions in two-stage stochastic and adaptive optimization problems. *Mathematics of Operations Research*, 35(2):284–305, 2010.

[16] D. Bertsimas and V. Goyal. On the power and limitations of affine policies in two-stage adaptive optimization. *Mathematical Programming*, 134(2):491–531, 2012.

[17] D. Bertsimas, V. Goyal, and X. A. Sun. A geometric characterization of the power of finite adaptability in multistage stochastic and adaptive optimization. *Mathematics of Operations Research*, 36(1):24–54, 2011.

[18] D. Bertsimas, D. A. Iancu, and P. A. Parrilo. Optimality of affine policies in multi-stage robust optimization. *Mathematics of Operations Research*, 35(2):363–394, 2010.

[19] D. Bertsimas, D. A. Iancu, and P. A. Parrilo. A hierarchy of near-optimal policies for multistage adaptive optimization. *IEEE Transactions on Automatic Control*, 56(12):2809–2824, 2011.

[20] D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng. Adaptive robust optimization for the security constrained unit commitment problem. *IEEE Transactions on Power Systems*, 28(1):52–63, 2013.

[21] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[22] C. Buchta, J. Müller, and R. F. Tichy. Stochastical approximation of convex bodies. *Mathematische Annalen*, 271(2):225–235, 1985.

[23] B. Büeler, A. Enge, and K. Fukuda. Exact volume computation for polytopes: A practical study. In G. Kalai and G. M. Ziegler, editors, *Polytopes – Combinatorics and Computation*, pages 131–154. Birkhäuser, 2000.

[24] X. Chen, M. Sim, P. Sun, and J. Zhang. A linear decision-based approximation approach to stochastic programming. *Operations Research*, 56(2):344–357, 2008.

[25] X. Chen and Y. Zhang. Uncertain linear programs: Extended affinely adjustable robust counterparts. *Operations Research*, 57(6):1469–1482, 2009.

[26] E. Delage and D. A. Iancu. Robust multistage decision making. *INFORMS Tutorials in Operations Research*, pages 20–46, 2015.

[27] V. Gabrel, C. Murat, and A. Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014.

[28] T. Gal. Degeneracy graphs: Theory and application an updated survey. *Annals of Operations Research*, 46(1):81–105, 1993.

[29] C. Gauvin, E. Delage, and M. Gendreau. A stochastic program with tractable time series and affine decision rules for the reservoir management problem. Available on Optimization Online, 2018.

[30] A. Georghiou, A. Tsoukalas, and W. Wiesemann. Robust dual dynamic programming. Available on Optimization Online, 2016.

[31] A. Georghiou, W. Wiesemann, and D. Kuhn. Generalized decision rule approximations for stochastic programming via liftings. *Mathematical Programming*, 152(1):301–338, 2015.

[32] J. Goh and M. Sim. Distributionally robust optimization and its tractable approximations. *Operations Research*, 58(4):902–917, 2010.

[33] B. L. Gorissen and D. den Hertog. Robust counterparts of inequalities containing sums of maxima of linear functions. *European Journal of Operational Research*, 227(1):30–43, 2013.

[34] B. L. Gorissen, İ. Yanıkoğlu, and D. den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015.

[35] C. E. Gounaris, W. Wiesemann, and C. A. Floudas. The robust capacitated vehicle routing problem under demand uncertainty. *Operations Research*, 61(3):677–603, 2013.

[36] E. Guslitser. Uncertainty-immunized solutions in linear programming. Master's thesis, Technion, 2002.

[37] M. J. Hadjiyiannis, P. J. Goulart, and D. Kuhn. A scenario approach for estimating the suboptimality of linear decision rules in two-stage robust optimization. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7386–7391, 2011.

[38] G. A. Hanasusanto and D. Kuhn. Conic programming reformulations of two-stage distributionally robust linear programs over Wasserstein balls. Available on Optimization Online, 2017.

[39] G. A. Hanasusanto, D. Kuhn, and W. Wiesemann. K-adaptability in two-stage robust binary programming. *Operations Research*, 63(4):877–891, 2015.

[40] M. Henk. Löwner–John ellipsoids. In *Documenta Mathematica – Extra volume: Optimization Stories*, pages 95–106. DMV, 2012.

[41] R. Jiang, M. Zhang, G. Li, and Y. Guan. Two-stage robust power grid optimization problem. Available on Optimization Online, 2010.

[42] Q. Kong, S. Li, N. Liu, C.-P. Teo, and Z. Yan. Robust multi-period vehicle routing under customer order uncertainty. Available on `http://www.columbia.edu/~nl2320/doc/Noshow-MS-1030c.pdf`, 2017.

[43] D. Kuhn, W. Wiesemann, and A. Georghiou. Primal and dual linear decision rules in stochastic and robust optimization. *Mathematical Programming*, 130(1):177–209, 2011.

[44] N. H. Lappas and C. E. Gounaris. Multi-stage adjustable robust optimization for process scheduling under uncertainty. *AIChE Journal*, 62(5):1646–1667, 2016.

[45] C. Liebchen, M. Lübbecke, R. H. Möhring, and S. Stiller. The concept of recoverable robustness, linear programming recovery, and railway applications. In R. K. Ahuja, R. H. Möhring, and C. D. Zaroliagis, editors, *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*, pages 1–27. Springer, 2009.

[46] A. Lorca and X. A. Sun. Adaptive robust optimization with dynamic uncertainty sets for multi-period economic dispatch under significant wind. *IEEE Transactions on Power Systems*, 30(4):1702–1713, 2015.

[47] A. Lorca and X. A. Sun. Multistage robust unit commitment with dynamic uncertainty sets and energy storage. *IEEE Transactions on Power Systems*, 32(3):1678–1688, 2017.

[48] L. Narici and E. Beckenstein. *Topological Vector Spaces.* Chapman and Hall, 2nd edition.

[49] K. Postek and D. den Hertog. Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing*, 28(3):553–574, 2016.

[50] C. Ruiz and A. J. Conejo. Robust transmission expansion planning. *European Journal of Operational Research*, 242(2):390–401, 2015.

[51] R. Seidel. Convex hull computations. In C. D. Toth, J. O'Rourke, and J. E. Goodman, editors, *Handbook of Discrete and Computational Geometry*, pages 494–512. Chapman and Hall, 2004.

[52] Y. Song and J. Luedtke. An adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse. *SIAM Journal on Optimization*, 25(3):1344–1367, 2015.

[53] A. Subramanyam, C. Gounaris, and W. Wiesemann. K-adaptability in two-stage mixed-integer robust optimization. Available on Optimization Online, 2017.

[54] A. Subramanyam, F. Mufalli, J. M. Pinto, and C. E. Gounaris. Robust multi-period vehicle routing under customer order uncertainty. Available on Optimization Online, 2017.

[55] A. Thiele, T. Terry, and M. Epelman. Robust linear optimization with recourse. Technical report, Lehigh University and University of Michigan, 2010.

[56] P. Vayanos, D. Kuhn, and B. Rustem. Decision rules for information discovery in multi-stage stochastic programming. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7368–7373, 2011.

[57] J. Warrington, P. Goulart, S. Mariéthoz, and M. Morari. Policy-based reserves for power systems. *IEEE Transactions on Power Systems*, 28(4):4427–4437, 2013.

[58] W. Wiesemann, D. Kuhn, and B. Rustem. Robust resource allocations in temporal networks. *Mathematical Programming*, 135(1–2):437–471, 2012.

[59] G. Xu and S. Burer. A copositive approach for two-stage adjustable robust optimization with uncertain right-hand sides. Available on Optimization Online, 2016.

[60] İ. Yanıkoğlu, B. L. Gorissen, and D. den Hertog. Adjustable robust optimization – a survey and tutorial. Available on ResearchGate, 2017.

[61] B. Zeng and L. Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–561, 2013.

[62] C. Zhao, J. Wang, J.-P. Watson, and Y. Guan. Multi-stage robust unit commitment considering wind and demand response uncertainties. *IEEE Transactions on Power Systems*, 28(3):2708–2717, 2013.

[63] J. Zhen, D. den Hertog, and M. Sim. Adjustable robust optimization via Fourier-Motzkin elimination. Available on Optimization Online, 2016.

[64] G. M. Ziegler. Lectures on 0/1-polytopes. In G. Kalai and G. M. Ziegler, editors, *Polytopes – Combinatorics and Computation*, pages 1–41. Birkhäuser, 2000.

A Appendix: Moment Matrices for Common Uncertainty Sets

This appendix elaborates on the calculation of the moment matrix $\boldsymbol{\Sigma}_A$ in the dual problem $\underline{\mathcal{P}}(\Xi_S)$. We first present the moment matrices $\boldsymbol{\Sigma}_A$ of several commonly used primitive uncertainty sets in Section A.1. Afterwards, Section A.2 shows how a moment matrix changes if particular classes of transformations are applied to an uncertainty set.

A.1 Primitive Uncertainty Sets

In the following, we list the moment matrices $\boldsymbol{\Sigma}_A$ of several commonly used primitive uncertainty sets. Since the derivations are basic but tedious, we omit them for the sake of brevity. In the following, we denote by $\lfloor x \rfloor$ and $\lceil x \rceil$ the largest (smallest) integer less than or equal to (larger than or equal to) $x \in \mathbb{R}$.

A.1.1 1-Norm Ball Uncertainty Sets

For a 1-norm ball uncertainty set of the form

$$\Xi = \left\{ \boldsymbol{\xi} \in \mathbb{R}^k \,:\, \|\boldsymbol{\xi}\|_1 \leq 1 \right\},$$

the set of extreme points satisfies

$$\operatorname{ext} \left\{ \boldsymbol{\xi} \in \mathbb{R}^k \,:\, \|\boldsymbol{\xi}\|_1 \leq 1 \right\} \;=\; \operatorname{ext} \left\{ \boldsymbol{\xi} \in \mathbb{R}^k \,:\, \sum_{i=1}^k |\xi_i| \leq 1 \right\} \;=\; \left\{ \pm \mathbf{e}_i \in \mathbb{R}^k \,:\, i = 1, \ldots, k \right\},$$

and we therefore obtain the moment matrix

$$\boldsymbol{\Sigma}_A \;=\; \begin{pmatrix} 2k & \mathbf{0}^\top \\ \mathbf{0} & 2 \cdot \boldsymbol{I} \end{pmatrix}.$$

A.1.2 ∞-Norm Ball Uncertainty Sets

We now consider an ∞-norm ball uncertainty set of the form

$$\Xi = \left\{ \boldsymbol{\xi} \in \mathbb{R}^k \,:\, \|\boldsymbol{\xi}\|_\infty \leq 1 \right\}.$$

Uncertainty sets of this type are commonly employed to describe the demand in operations management applications [8, 34] or when the uncertainty underlying the parameters $\boldsymbol{\xi}$ is described by a factor model [35]. The set of extreme points satisfies

$$\operatorname{ext}\left\{\boldsymbol{\xi} \in \mathbb{R}^k \,:\, \|\boldsymbol{\xi}\|_\infty \leq 1\right\} \;=\; \operatorname{ext}\left\{\boldsymbol{\xi} \in \mathbb{R}^k \,:\, \max_{i=1,\ldots,k} |\xi_i| \leq 1\right\} \;=\; \{-1,1\}^k,$$

and we therefore obtain the moment matrix

$$\boldsymbol{\Sigma}_A \;=\; \begin{pmatrix} 2^k & \mathbf{0}^\top \\ \mathbf{0} & 2^k \cdot \boldsymbol{I} \end{pmatrix}.$$

A.1.3 $(1 \cap \infty)$-Norm Ball Uncertainty Sets

We now consider an uncertainty set that emerges from the intersection of a scaled 1-norm ball and an ∞-norm ball:

$$\Xi = \left\{\boldsymbol{\xi} \in \mathbb{R}^k \,:\, \|\boldsymbol{\xi}\|_1 \leq \kappa, \;\; \|\boldsymbol{\xi}\|_\infty \leq 1\right\}$$

For $\kappa \leq 1$ and $\kappa \geq k$, the uncertainty set reduces to a scaled 1-norm ball and an ∞-norm ball, respectively. We therefore assume that $\kappa \in (1, k)$. Uncertainty sets of this type are commonly used as polyhedral outer approximations of ellipsoidal uncertainty sets [29, 35].

If $\kappa \in \mathbb{N}$, we obtain the moment matrix

$$\boldsymbol{\Sigma}_A \;=\; \begin{pmatrix} 2^\kappa \binom{k}{\kappa} & \mathbf{0}^\top \\ \mathbf{0} & 2^\kappa \dbinom{k-1}{\kappa-1} \cdot \boldsymbol{I} \end{pmatrix}.$$

For fractional κ, on the other hand, the moment matrix is

$$\boldsymbol{\Sigma}_A \;=\; \begin{pmatrix} 2^{\lceil\kappa\rceil} \lceil\kappa\rceil \binom{k}{\lceil\kappa\rceil} & \mathbf{0}^\top \\ \mathbf{0} & 2^{\lceil\kappa\rceil} \dbinom{k-1}{\lfloor\kappa\rfloor} \left(\lfloor\kappa\rfloor + (\kappa - \lfloor\kappa\rfloor)^2\right) \cdot \boldsymbol{I} \end{pmatrix}.$$

A.1.4 Budget Uncertainty Sets

We next consider a budget uncertainty set of the form

$$\Xi = \left\{\boldsymbol{\xi} \in [0,1]^k \,:\, \mathbf{e}^\top \boldsymbol{\xi} \leq B\right\},$$

where $B \in \mathbb{N}_0$. Note that $B = 0$ and $B \geq k$ correspond to the cases $\Xi = \{\mathbf{0}\}$ and $\Xi = [0,1]^k$ (the latter being a translation of the ∞-norm ball, which can be calculated using the results of Section A.1.2 and the transformations from Section A.2) and can therefore be omitted. Budget uncertainty sets have been popularized by [21] and have since been applied widely across domains.

For $B \in \{1, \ldots, k-1\}$, the moment matrix is

$$\boldsymbol{\Sigma}_A = \begin{pmatrix} \displaystyle\sum_{i=0}^{B} \binom{k}{i} & \left[\displaystyle\sum_{i=0}^{B-1} \binom{k-1}{i}\right] \cdot \mathbf{e}^\top \\ \left[\displaystyle\sum_{i=0}^{B-1} \binom{k-1}{i}\right] \cdot \mathbf{e} & \left[\displaystyle\sum_{i=0}^{B-1} \binom{k-1}{i} - \sum_{i=0}^{B-2} \binom{k-2}{i}\right] \cdot \boldsymbol{I} + \left[\displaystyle\sum_{i=0}^{B-2} \binom{k-2}{i}\right] \cdot \mathbf{e}\mathbf{e}^\top \end{pmatrix}.$$

A.1.5 Central Limit Theorem-Type Uncertainty Sets

We finally consider a central limit theorem-type uncertainty set of the form

$$\Xi = \left\{ \boldsymbol{\xi} \in [-1,1]^k \,:\, -\Gamma \leq \left| \mathbf{e}^\top \boldsymbol{\xi} \right| \leq +\Gamma \right\},$$

where $\Gamma \in (0, k)$ [7]. In this case, we have $|\mathrm{ext}\, \Xi| = \eta_1 + \eta_2 + \eta_3$, where

$$\eta_1 = \begin{cases} \displaystyle\binom{k}{\frac{k}{2}} + 2 \sum_{i=1}^{\lfloor \frac{\Gamma}{2} \rfloor} \binom{k}{\frac{k}{2} + i} & \text{if } k \text{ is even,} \\ \displaystyle 2 \sum_{i=0}^{\lfloor \frac{\Gamma-1}{2} \rfloor} \binom{k}{\frac{k+1}{2} + i} & \text{if } k \text{ is odd;} \end{cases}$$

$$\eta_2 = \begin{cases} 2k \displaystyle\binom{k-1}{\frac{k+\Gamma-1}{2}} & \text{if } k \text{ is even and } \Gamma \text{ is odd, or vice versa,} \\ 0 & \text{otherwise;} \end{cases}$$

$$\eta_3 = \begin{cases} 2 \displaystyle\binom{k}{\frac{k}{2} + \lfloor \frac{\Gamma}{2} \rfloor} \cdot \left(\frac{k}{2} - \left\lfloor \frac{\Gamma}{2} \right\rfloor \right) & \text{if } k \text{ is even and } \Gamma \text{ is fractional,} \\ 2 \displaystyle\binom{k}{\frac{k+1}{2} + \lfloor \frac{\Gamma-1}{2} \rfloor} \cdot \left(\frac{k-1}{2} - \left\lfloor \frac{\Gamma-1}{2} \right\rfloor \right) & \text{if } k \text{ is odd and } \Gamma \text{ is fractional,} \\ 0 & \text{otherwise.} \end{cases}$$

Due to the symmetry of Ξ, the sum of first moments satisfies $\sum_{\boldsymbol{\xi} \in \operatorname{ext} \Xi} \boldsymbol{\xi} = \mathbf{0}$. The sum of second moments, finally, satisfies $\sum_{\boldsymbol{\xi} \in \operatorname{ext} \Xi} \boldsymbol{\xi}\boldsymbol{\xi}^\top = \omega \cdot (\mathbf{e}\mathbf{e}^\top) + (\delta - \omega) \cdot \boldsymbol{I}$, where $\delta = \delta_1 + \delta_2 + \delta_3$ with

$$\delta_1 = \begin{cases} \binom{k}{\frac{k}{2}} + 2 \sum_{i=1}^{\lfloor \frac{\Gamma}{2} \rfloor} \binom{k}{\frac{k}{2} + i} & \text{if } k \text{ is even,} \\ 2 \sum_{i=0}^{\lfloor \frac{\Gamma-1}{2} \rfloor} \binom{k}{\frac{k+1}{2} + i} & \text{if } k \text{ is odd;} \end{cases}$$

$$\delta_2 = \begin{cases} 2(k-1)\binom{k-1}{\frac{k+\Gamma-1}{2}} & \text{if } k \text{ is even and } \Gamma \text{ is odd, or vice versa,} \\ 0 & \text{otherwise;} \end{cases}$$

$$\delta_3 = \begin{cases} 2\binom{k}{\frac{k}{2} + \lfloor \frac{\Gamma}{2} \rfloor} \left(\frac{k}{2} - \left\lfloor \frac{\Gamma}{2} \right\rfloor\right) \left[\frac{k-1}{k} + \frac{1}{k} \cdot \left(1 - \left[\Gamma - 2\left\lfloor \frac{\Gamma}{2} \right\rfloor\right]\right)^2\right] & \text{if } k \text{ is even and } \Gamma \text{ is fractional,} \\ 2\binom{k}{\frac{k+1}{2} + \lfloor \frac{\Gamma-1}{2} \rfloor} \left(\frac{k-1}{2} - \left\lfloor \frac{\Gamma-1}{2} \right\rfloor\right) \left[\frac{k-1}{k} + \frac{1}{k} \cdot \left(2 - \Gamma + 2\left\lfloor \frac{\Gamma-1}{2} \right\rfloor\right)^2\right] & \text{if } k \text{ is odd and } \Gamma \text{ is fractional,} \\ 0 & \text{otherwise,} \end{cases}$$

as well as $\omega = \omega_1 + \omega_2 + \omega_3$ with

$$\omega_1 = \begin{cases} 2 \sum_{i=\max\{2-\frac{k}{2}, -\lfloor \frac{\Gamma}{2} \rfloor\}}^{\min\{\frac{k}{2}, \lfloor \frac{\Gamma}{2} \rfloor\}} \binom{k-2}{\frac{k}{2} + i - 2} - 2 \sum_{i=\max\{1-\frac{k}{2}, -\lfloor \frac{\Gamma}{2} \rfloor\}}^{\min\{\frac{k}{2}-1, \lfloor \frac{\Gamma}{2} \rfloor\}} \binom{k-2}{\frac{k}{2} + i - 1} & \text{if } k \text{ is even,} \\ 2 \sum_{i=\max\{\frac{1}{2}-\frac{k}{2}, -\lfloor \frac{\Gamma-1}{2} \rfloor - 2\}}^{\min\{\frac{k}{2}-\frac{3}{2}, \lfloor \frac{\Gamma-1}{2} \rfloor - 1\}} \binom{k-2}{\frac{k-1}{2} + i} - 2 \sum_{i=\max\{\frac{1}{2}-\frac{k}{2}, -\lfloor \frac{\Gamma-1}{2} \rfloor - 1\}}^{\min\{\frac{k}{2}-\frac{3}{2}, \lfloor \frac{\Gamma-1}{2} \rfloor\}} \binom{k-2}{\frac{k-1}{2} + i} & \text{if } k \text{ is odd;} \end{cases}$$

$\omega_2 = \omega_{21} + \omega_{22} + \omega_{23}$ if k is even and Γ is odd, or vice versa, and $\omega_2 = 0$ otherwise, where

$$\omega_{21} = \begin{cases} 2(k-2)\binom{k-3}{\frac{k+\Gamma-5}{2}} & \text{if } k + \Gamma \geq 5,\ k \geq \Gamma + 1, \\ 0 & \text{otherwise;} \end{cases}$$

$$\omega_{22} = \begin{cases} 2(k-2)\binom{k-3}{\frac{k-\Gamma-5}{2}} & \text{if } k \geq \Gamma + 5, \\ 0 & \text{otherwise;} \end{cases} \qquad \omega_{23} = \begin{cases} -4 \cdot (k-2)\binom{k-3}{\frac{k+\Gamma-3}{2}} & \text{if } k \geq \Gamma + 3, \\ 0 & \text{otherwise;} \end{cases}$$

and finally $\omega_3 = \omega_{31} - \omega_{32} + \omega_{33} - \omega_{34}$ if k is even and Γ is fractional, where

$$\omega_{31} = \begin{cases} 2\binom{k-2}{\frac{k}{2} + \lfloor \frac{\Gamma}{2} \rfloor - 2} \cdot \left(\frac{k}{2} - \left\lfloor \frac{\Gamma}{2} \right\rfloor\right) + 2\binom{k-2}{\frac{k}{2} - \lfloor \frac{\Gamma}{2} \rfloor - 2} \cdot \left(\frac{k}{2} - \left\lfloor \frac{\Gamma}{2} \right\rfloor - 2\right) & \text{if } \frac{k}{2} \geq \left\lfloor \frac{\Gamma}{2} \right\rfloor + 2, \\ 0 & \text{otherwise;} \end{cases}$$

$$\omega_{32} = 4\binom{k-2}{\frac{k}{2} + \lfloor \frac{\Gamma}{2} \rfloor - 1} \cdot \left(\frac{k}{2} - \left\lfloor \frac{\Gamma}{2} \right\rfloor - 1\right);$$

$$\omega_{33} = \begin{cases} 4\binom{k-2}{\frac{k}{2} - \lfloor \frac{\Gamma}{2} \rfloor - 2} & \text{if } \frac{k}{2} \geq \left\lfloor \frac{\Gamma}{2} \right\rfloor + 2, \\ 0 & \text{otherwise;} \end{cases} \qquad \omega_{34} = \begin{cases} 4\binom{k-2}{\frac{k}{2} - \lfloor \frac{\Gamma}{2} \rfloor - 1} & \text{if } \frac{k}{2} \geq \left\lfloor \frac{\Gamma}{2} \right\rfloor + 1, \\ 0 & \text{otherwise;} \end{cases}$$

as well as $\omega_3 = \omega_{35} + \omega_{36} - \omega_{37} + \omega_{38} - \omega_{39}$ if k is odd and Γ is fractional, where

$$\omega_{35} = \begin{cases} 2\binom{k-2}{\frac{k-3}{2} + \lfloor \frac{\Gamma-1}{2} \rfloor} \cdot \left(\frac{k-1}{2} - \left\lfloor \frac{\Gamma-1}{2} \right\rfloor\right) & \text{if } \frac{k-1}{2} \geq \left\lfloor \frac{\Gamma-1}{2} \right\rfloor, \\ 0 & \text{otherwise;} \end{cases}$$

$$\omega_{36} = \begin{cases} 2\binom{k-2}{\frac{k-5}{2} - \lfloor \frac{\Gamma-1}{2} \rfloor} \cdot \left(\frac{k-5}{2} - \left\lfloor \frac{\Gamma-1}{2} \right\rfloor\right) & \text{if } \frac{k-5}{2} \geq \left\lfloor \frac{\Gamma-1}{2} \right\rfloor, \\ 0 & \text{otherwise;} \end{cases}$$

$$\omega_{37} = \begin{cases} 4\binom{k-2}{\frac{k-1}{2} + \lfloor \frac{\Gamma-1}{2} \rfloor} \cdot \left(\frac{k-3}{2} - \left\lfloor \frac{\Gamma-1}{2} \right\rfloor\right) & \text{if } \frac{k-3}{2} \geq \left\lfloor \frac{\Gamma-1}{2} \right\rfloor, \\ 0 & \text{otherwise;} \end{cases}$$

$$\omega_{38} = \begin{cases} 4\binom{k-2}{\frac{k-5}{2} - \lfloor \frac{\Gamma-1}{2} \rfloor} & \text{if } \frac{k-5}{2} \geq \left\lfloor \frac{\Gamma-1}{2} \right\rfloor, \\ 0 & \text{otherwise;} \end{cases} \qquad \omega_{39} = \begin{cases} 4\binom{k-2}{\frac{k-3}{2} - \lfloor \frac{\Gamma-1}{2} \rfloor} & \text{if } \frac{k-3}{2} \geq \left\lfloor \frac{\Gamma-1}{2} \right\rfloor, \\ 0 & \text{otherwise.} \end{cases}$$

A.2 Transformations of Primitive Uncertainty Sets

In the following, we show how a moment matrix $\mathbf{\Sigma}_A$ for an uncertainty set Ξ changes if Ξ is transformed by an injective affine map, or if Ξ is composed of the cross product of primitive uncertainty sets Ξ_i with known moment matrices $\mathbf{\Sigma}_{A,i}$.

A.2.1 Injective Affine Maps

It follows from Theorem 9.2.3 in [48] that extreme points are preserved under injective affine maps. Let us assume that $\Xi' = f(\Xi)$ for an injective affine map $f : \mathbb{R}^k \overset{A}{\mapsto} \mathbb{R}^{k'}$. We then obtain that

$$\boldsymbol{\Sigma}'_A \;=\; \begin{pmatrix} |\mathrm{ext}\,\Xi| & \sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} f(\boldsymbol{\xi})^\top \\ \sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} f(\boldsymbol{\xi}) & \sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} f(\boldsymbol{\xi})f(\boldsymbol{\xi})^\top \end{pmatrix}.$$

Let us now additionally assume that $f(\boldsymbol{\xi}) = \boldsymbol{F}\boldsymbol{\xi} + \boldsymbol{f}$. From the previous assumption that f is injective, we conclude that the matrix $\boldsymbol{F}$ has full column rank. We then have

$$\sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} f(\boldsymbol{\xi}) \;=\; \sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} (\boldsymbol{F}\boldsymbol{\xi} + \boldsymbol{f}) \;=\; \boldsymbol{F}\left[\sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} \boldsymbol{\xi}\right] + \boldsymbol{f} + |\mathrm{ext}\,\Xi|\boldsymbol{f}.$$

In other words, we can calculate $\sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} f(\boldsymbol{\xi})$ efficiently from the quantities $|\mathrm{ext}\,\Xi|$ and $\sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} \boldsymbol{\xi}$. In a similar way, we obtain that

$$\begin{aligned} \sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} f(\boldsymbol{\xi})f(\boldsymbol{\xi})^\top \;=\; \sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} (\boldsymbol{F}\boldsymbol{\xi} + \boldsymbol{f})(\boldsymbol{F}\boldsymbol{\xi} + \boldsymbol{f})^\top \;=\; & \left(\boldsymbol{F}\left[\sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} \boldsymbol{\xi}\right]\boldsymbol{f}^\top + \left[\boldsymbol{F}\left[\sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} \boldsymbol{\xi}\right]\boldsymbol{f}^\top\right]^\top\right) \\ & + \boldsymbol{F}\left[\sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} \boldsymbol{\xi}\boldsymbol{\xi}^\top\right]\boldsymbol{F}^\top + |\mathrm{ext}\,\Xi|\boldsymbol{f}\boldsymbol{f}^\top. \end{aligned}$$

In other words, we can calculate $\sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} f(\boldsymbol{\xi})f(\boldsymbol{\xi})^\top$ efficiently from the quantities $|\mathrm{ext}\,\Xi|$, $\sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} \boldsymbol{\xi}$ and $\sum_{\boldsymbol{\xi}\in\mathrm{ext}\,\Xi} \boldsymbol{\xi}\boldsymbol{\xi}^\top$.

A.2.2 Cross Products

Assume that $\Xi = \Xi_1 \times \Xi_2$ with $\Xi_1 \subseteq \mathbb{R}^{k_1}$ and $\Xi_2 \subseteq \mathbb{R}^{k_2}$ such that $k_1 + k_2 = k$. We then have

$$|\operatorname{ext} \Xi| = |\operatorname{ext} \Xi_1| \cdot |\operatorname{ext} \Xi_2|,$$

$$\sum_{\boldsymbol{\xi} \in \operatorname{ext} \Xi} \boldsymbol{\xi} = \begin{pmatrix} |\operatorname{ext} \Xi_2| \cdot \displaystyle\sum_{\boldsymbol{\xi}_1 \in \operatorname{ext} \Xi_1} \boldsymbol{\xi}_1 \\ |\operatorname{ext} \Xi_1| \cdot \displaystyle\sum_{\boldsymbol{\xi}_2 \in \operatorname{ext} \Xi_2} \boldsymbol{\xi}_2 \end{pmatrix},$$

$$\sum_{\boldsymbol{\xi} \in \operatorname{ext} \Xi} \boldsymbol{\xi}\boldsymbol{\xi}^\top = \begin{pmatrix} |\operatorname{ext} \Xi_2| \cdot \displaystyle\sum_{\boldsymbol{\xi}_1 \in \operatorname{ext} \Xi_1} \boldsymbol{\xi}_1\boldsymbol{\xi}_1^\top & \left(\displaystyle\sum_{\boldsymbol{\xi}_1 \in \operatorname{ext} \Xi_1} \boldsymbol{\xi}_1\right)\left(\displaystyle\sum_{\boldsymbol{\xi}_2 \in \operatorname{ext} \Xi_2} \boldsymbol{\xi}_2\right)^\top \\ \left(\displaystyle\sum_{\boldsymbol{\xi}_2 \in \operatorname{ext} \Xi_2} \boldsymbol{\xi}_2\right)\left(\displaystyle\sum_{\boldsymbol{\xi}_1 \in \operatorname{ext} \Xi_1} \boldsymbol{\xi}_1\right)^\top & |\operatorname{ext} \Xi_1| \cdot \displaystyle\sum_{\boldsymbol{\xi}_2 \in \operatorname{ext} \Xi_2} \boldsymbol{\xi}_2\boldsymbol{\xi}_2^\top \end{pmatrix}.$$

In other words, we can calculate $|\operatorname{ext} \Xi|$, $\sum_{\boldsymbol{\xi} \in \operatorname{ext} \Xi} \boldsymbol{\xi}$ and $\sum_{\boldsymbol{\xi} \in \operatorname{ext} \Xi} \boldsymbol{\xi}\boldsymbol{\xi}^\top$ efficiently from the quantities $|\operatorname{ext} \Xi_i|$, $\sum_{\boldsymbol{\xi} \in \operatorname{ext} \Xi_i} \boldsymbol{\xi}$ and $\sum_{\boldsymbol{\xi} \in \operatorname{ext} \Xi_i} \boldsymbol{\xi}\boldsymbol{\xi}^\top$, $i = 1, 2$.