Adjustability in Robust Linear Optimization

Abstract

We investigate the concept of adjustability – the difference in objective values between two types of dynamic robust optimization formulations: one where (static) decisions are made before uncertainty realization, and one where uncertainty is resolved before (adjustable) decisions. This difference reflects the value of information and decision timing in optimization under uncertainty, and is related to several other concepts such as interchangeability in games and optimality of decision rules in robust optimization. We develop a theoretical framework to quantify adjustability based on the input data of a robust optimization problem with linear objective, linear constraints, and fixed recourse. We make very few additional assumptions. In particular, we do not assume constraint-wise separability or parameter nonnegativity that are commonly imposed in the literature for the study of adjustability. This allows us to study important but previously under-investigated problems, such as formulations with equality constraints and problems with both upper and lower bound constraints. Based on the discovery of an interesting connection between the reformulations of the static and fully adjustable problems, our analysis gives a necessary and sufficient condition – in the form of a theorem-of-the-alternatives – for adjustability to be zero when the uncertainty set is polyhedral. Based on this sharp characterization, we provide a mixed-integer optimization formulation as a certificate of zero adjustability. Then, we develop a constructive approach to quantify adjustability when the uncertainty set is general, which results in an efficient and tight algorithm to bound adjustability. We demonstrate the efficiency and tightness via both theoretical and numerical analyses.

Keywords. adjustability; adjustable robust optimization; game with strategy coupling.

1 Introduction

Consider a decision-maker who wants to find optimal decisions y in an environment plagued by some uncertainty represented by parameters ξ. Assume that the decision-maker cares about worst-case performance, either because she is risk-averse or because uncertainty realization is picked by an adversary. If the sequence of events is fixed and known, *i.e.*, the order of decision commitments and uncertain parameter realizations can be specified, then one can formulate this problem into a dynamic robust optimization model. Otherwise, even setting up an optimization model becomes challenging. How should the decision-maker analyze this ill-posed problem?

It turns out that something can still be said by just looking at the two "extreme" cases of the problem, represented by the following two optimization models,

$$\text{(RO)}\ \ \mathrm{I} := \min_{y \in \bigcap\limits_{\xi\in\Xi} \mathcal{Y}_\xi} \max_{\xi\in\Xi} f(\xi, y), \qquad \text{(FARO)}\ \ \mathrm{II} := \min_{y(\cdot) \in \prod\limits_{\xi\in\Xi} \mathcal{Y}_\xi} \max_{\xi\in\Xi} f(\xi, y(\xi)),$$

where $\mathcal{X}, \mathcal{Y}$ are two Euclidean spaces, $\Xi \subseteq \mathcal{X}$ is the uncertainty set, $\mathcal{Y}_\xi \subseteq \mathcal{Y}$ is the feasible decision space of y depending on the fixed set of parameters ξ. Then, $\bigcap_{\xi\in\Xi} \mathcal{Y}_\xi$ consists of solutions that are feasible for every realization of $\xi \in \Xi$, while $\prod_{\xi\in\Xi} \mathcal{Y}_\xi$, called the *policy space*, contains every function $y : \mathcal{X} \to \mathcal{Y}$ that satisfies $y(\xi) \in \mathcal{Y}_\xi$ for all $\xi \in \Xi$ (the product sign is commonly used for the set of dependent functions). Problems I and II are recognized as (static) robust optimization (RO) and fully adjustable robust optimization (FARO) (Ben-Tal *et al.* 2009) models. In the former case, all decisions have to be made prior to any uncertainty realization, while in the latter, all decisions are made afterward.

The main research focus of this paper is to understand the difference between these two problems. More precisely, we ask two questions: (i) can we identify conditions under which the value of the two problems, denoted by $z(\mathrm{I})$ and $z(\mathrm{II})$, are equal? (ii) When $z(\mathrm{I}) \neq z(\mathrm{II})$, can we measure the corresponding gap?

The answers to these two questions will provide insights for multiple associated problems and concepts, such as the interchangeablity of zero-sum games with strategy coupling, the conservativeness of static robust solutions, the performance of various policy families in adjustable robust optimization (ARO), among others. Moreover, a quantified gap between I and II can very well inform the decision-maker's actions. Suppose this gap is large, the decision-maker is inclined to postpone their decisions and invest more in revealing the uncertainty and reducing the decision lead time. Otherwise, making decision early could be a viable option.

In this paper, we answer both questions positively, providing theoretical characterizations and algorithmic tools to verify the conditions for the two values to be equal and quantify the corresponding gap termed *adjustability*. As aforementioned, this concept is closely connected to several interesting problems.

1.1 Related Problems and Concepts

The study of adjustability can provide theoretical insights to several classes of problems. In this subsection, we provide a precise account of these relationships.

1.1.1 Equivalence of Equilibrium Concepts in Games with Strategy Coupling

Focus on the following problem,

$$\text{III} := \max_{\xi \in \Xi} \min_{y \in \mathcal{Y}_\xi} f(\xi, y).$$

It is intuitively true and easy to show that $z(\text{II}) = z(\text{III})$. Thus, the adjustability also measures the gap between Problem I and III. Notice that Problem I can be considered as the minimax counterpart of III when there exists strategy dependence between the two players, *i.e.*, one player's action ξ affects the other player's feasible strategy space $\mathcal{Y}_\xi$. Therefore, adjustability quantifies the interchangeability of "min" and "max" of such zero-sum sequential games. In particular, the condition for the gap to be zero entails the corresponding minimax theorem. This type of coupling constraints commonly appears in interdiction and defender-attacker-defender games, which have broad applications in transportation (Israeli and Wood 2002), communication (Wei *et al.* 2021), human trafficking prevention (Konrad *et al.* 2017), cyber security (Etesami and Başar 2019) , power networks (An *et al.* 2020), evacuation planning (Matisziw and Murray 2009), among others. In these games, interchangeability (*i.e.*, $z(\text{I}) = z(\text{III})$) implies that the defender can obtain an optimal solution without detecting the attacker's strategy, *i.e.*, Stackelberg and pure Nash equilibria both exist and are equivalent (Korzhyk *et al.* 2011, Tsaknakis *et al.* 2021).

1.1.2 Policy Family Performance in ARO

In general, Formulation II suffers from tractability issues caused by the infinite dimensionality of the policy space $\prod_{\xi\in\Xi}\mathcal{Y}_\xi$ (Ben-Tal *et al.* 2004). To address this, a common method is to restrict the decision space to some subset $\mathcal{Y}_\Xi \subseteq \prod_{\xi\in\Xi}\mathcal{Y}_\xi$ called a *policy family* (also called *decision rules*). Then, the performance of the selected policy family $\mathcal{Y}_\Xi$ is measured by the gap between Problem II and the following,

$$\text{IV} := \min_{y(\cdot)\in\mathcal{Y}_\Xi} \max_{\xi\in\Xi} f(\xi, y(\xi)).$$

In particular, Problem I corresponds to the case where $\mathcal{Y}_\Xi$ consists of only the constant policies (Bertsimas *et al.* 2015). Interestingly, most commonly considered policy families in the literature, such as affine (Bertsimas *et al.* 2010), piecewise constant (also called K-adaptability) (Hanasusanto *et al.* 2015), piecewise affine (Chen *et al.* 2008), and polynomial policy families (Bampou and Kuhn 2011), all contain the constant policy family as a subset. Thus, adjustability bounds the performances of all these policy families. Moreover, every dynamic robust optimization of the following form,

$$\left(\min_{y_1\in\mathcal{Y}_1}\max_{\xi_1\in\Xi_1}\min_{y_2\in\mathcal{Y}_2}\max_{\xi_2\in\Xi_2}\cdots\min_{y_l\in\mathcal{Y}_l}\max_{\xi_l\in\Xi_l}\cdots\right) f(\xi, y),$$

can be rewritten as

$$\text{V} := \min_{y(\cdot)\in\mathcal{Y}'_\Xi} \max_{\xi\in\Xi} f(\xi, y(\xi)),$$

where $\mathcal{Y}'_\Xi$ is the set of feasible history-dependent policies. Hence, same as before, $z(\text{V})$ is sandwiched between $z(\text{I})$ and $z(\text{II})$. These problems are commonly encountered in minimax control and online algorithm design problems (Bertsimas *et al.* 2010, Bertsekas 2012).

1.1.3 Reformulation-Linearization Technique (RLT)

Sherali and Alameddine (1992) introduced RLT as a reformulation technique that transforms a given bilinear formulation into a linear program to improve tractability. Recently, Ardestani-Jaafari and Delage (2021) discovered that the linear reformulation obtained from RLT is equivalent to an ARO restricted by the affine policy family, *i.e.*, the so-called affinely adjustable robust counterpart (AARC) of the ARO. Consider the following bilinear program,

$$\text{VI} := \max_{(x,y)\in\mathcal{Z}} c^\intercal x + d^\intercal y + x^\intercal G y,$$

where $\mathcal{Z}$ is an arbitrary joint space for x and y. If for every fixed x (or y), the problem has a dual representation that satisfies strong duality, then Formulation VI can be reformulated equivalently into a maximin problem as Formulation III. Then, the linear program obtained from RLT is equivalent to the AARC of Formulation II. Therefore, the adjustability gap also bounds the optimality gap induced by RLT in a given bilinear program.

1.1.4 Regret Optimization

A two-stage worst-case regret optimization (Poursoltani and Delage 2021) is modeled as

$$\text{VII} := \min_{y\in\mathcal{Y}_\Xi} \max_{\xi\in\Xi} \left(f(\xi, y) - \min_{y'\in\mathcal{Y}_\xi} f(\xi, y') \right),$$

which searches for a solution y that minimizes the point-wise regret between a realization $f(\xi, y)$ and an oracle. Interestingly, adjustability is a valid lower bound of $z(\text{VII})$. To see this, we rewrite Formulation VII as the following equivalent formulation,

$$\min_{y\in\mathcal{Y}_\Xi} \max_{\xi\in\Xi} \min_{\xi'\in\Xi:\xi'=\xi} \left(f(\xi, y) - \min_{y'\in\mathcal{Y}_{\xi'}} f(\xi', y') \right).$$

Without the constraint $\xi' = \xi$, this formulation is equivalent to Formulation I minus III. Thus, for every fixed pair $y \in \mathcal{Y}_\Xi$ and $\xi \in \Xi$, $z(\text{VII})$ is greater than or equal to $z(\text{I}) - z(\text{III})$, which is also true for the optimal y and ξ. Thus, $z(\text{VII})$ is lower bounded by the adjustability gap.

1.1.5 *A Priori* Optimization & Stochastic Gap

When the uncertainty set Ξ is coupled with a joint probability distribution $\mathcal{D}$, one may be interested in comparing the average performances between making decisions proactively and the ideal case where reoptimization is conducted for every $\xi \in \Xi$.

$$\text{VIII} := \min_{y\in\bigcap_{\xi\in\Xi}\mathcal{Y}_\xi} \mathbb{E}_{\xi\sim\mathcal{D}}\left[f(\xi, y)\right], \qquad \text{IX} := \mathbb{E}_{\xi\sim\mathcal{D}}\left[\min_{y\in\mathcal{Y}_\xi} f(\xi, y)\right].$$

These two problems are introduced as *a priori optimization* and *reoptimization* methods in Bertsimas *et al.* (1990). Then, the gap or ratio between z(VIII) and z(IX) is a measure of the quality of *a priori* decisions. From this perspective, adjustability can be viewed as the counterpart of the difference between z(VIII) and z(IX) under the worst-case lens. We note that the relationship between $z(\text{VIII}) - z(\text{IX})$ and the adjustability gap is inconclusive in general since the distribution over Ξ can be arbitrary. However, as pointed out by Bertsimas and Goyal (2010), we have $z(\text{IX}) \leq z(\text{III})$, which implies the relationship $z(\text{IX}) \leq z(\text{II}) \leq z(\text{I})$. Therefore, given $z(\text{IX}) > 0$, adjustability ratio $z(\text{II})/z(\text{IX})$ is a valid lower bound of the *stochastic gap* $z(\text{I})/z(\text{IX})$ studied in Bertsimas and Goyal (2010) and Bertsimas *et al.* (2011).

1.2 Literature Review

Robust optimization (RO) is a modeling approach to address parameter uncertainty in various decision problems. This method has been extensively studied (Ben-Tal *et al.* 2009, Ben-Tal and Nemirovski 2002, Bertsimas and Sim 2003, 2004) and has gained traction in many application areas such as inventory theory (Bertsimas and Thiele 2006, Ardestani-Jaafari and Delage 2016), supply chain management (Simchi-Levi *et al.* 2019, Bandi *et al.* 2019a), queuing theory (Bandi *et al.* 2019b), scheduling and transportation (Lappas and Gounaris 2016, Ardestani-Jaafari and Delage 2021, Shi *et al.* 2020), portfolio optimization (Ghaoui *et al.* 2003), healthcare (Iancu *et al.* 2021), Markov decision process (Nilim and El Ghaoui 2005), among others (Delage and Ye 2010, Wiesemann *et al.* 2014). The core idea of RO is to produce a worst-case optimal solution that is feasible to all possible uncertainty realizations.

Sometimes robust optimization is considered to be too conservative. Consequently, a handful of results have been developed to measure this conservativeness, which are closely related to the concept of adjustability. When first introducing linear ARO with nonnegative fixed recourse, Ben-Tal *et al.* (2004) showed that constant policy is optimal if the uncertainty set Ξ is a box set (*i.e.*, hyperrectangle). Marandi and Den Hertog (2018) generalized this result to ARO with constraints that are convex-concave on the product space of uncertainty set and decision space. These results heavily depend on the constraint-wise separability condition provided by the box uncertainty set. For non-box uncertainty sets, Bertsimas and Goyal (2010) derived several constant policy approximation ratios for uncertainty sets with special properties. Later, Bertsimas *et al.* (2011) generalized this method to provide tighter bounds for various uncertainty sets using an upper bound called the stochastic gap. For a particular class of variable recourse ARO problems, Bertsimas *et al.* (2015) proved that constant policy is optimal if the constraint set satisfies certain convexity conditions, and a non-convexity measurement can bound the approximation ratio. Awasthi *et al.* (2019) studied the constant optimality gap in a two-stage adjustable robust packing linear optimization problem where the uncertainty set is column-wise and constraint-wise. For these non-box set results, the uncertainty set under consideration is assumed to locate entirely within the nonnegative orthant. Recently, Iancu *et al.* (2021) showed that in a multiperiod problem, the constant policy is optimal if the objective has certain monotonicity properties and the uncertainty set has certain ordering (*e.g.*, lattice) properties.

To overcome the conservativeness nature of RO, Ben-Tal *et al.* (2004) proposed the ARO problem, where some of the decision variables can be determined after uncertainty realization, *i.e.*, are policies over the uncertainty set. The downside is, solving ARO exactly is intractable in general (Ben-Tal *et al.* 2004, Ben-Tal *et al.* 2009). Thus, one either has to use heuristics (*e.g.*, various policy families) or adopt a column/row generation method (Zhen *et al.* 2018). Common policy families include constant (Bertsimas *et al.* 2015), affine (Bertsimas *et al.* 2010), piecewise constant (also called K-adaptability (Hanasusanto *et al.* 2015)), piecewise affine (Chen *et al.* 2008), or polynomial policies (Bampou and Kuhn 2011). A central question about ARO is the optimality criteria and gap of various policy families. In particular, the affine policy family has attracted considerable attention (Chen and Zhang 2009, Bertsimas *et al.* 2010, Bertsimas and Goyal 2012, Iancu *et al.* 2013, Ardestani-Jaafari and Delage 2016, Simchi-Levi *et al.* 2019, El Housni and Goyal 2021, Haddad-Sisakht and Ryan 2018) in recent years due to its balanced trade-off between tractability in computation and quality in approximation. We refer interested readers to İhsan Yanıkoğlu *et al.* (2019) for a comprehensive survey on this topic.

1.3 Contributions

Most results in the RO literature regarding adjustability have restrictive assumptions on the uncertainty set and/or optimization model, such as the aforementioned non-negativity condition (Bertsimas and Goyal 2010, Bertsimas *et al.* 2011, 2015, Awasthi *et al.* 2019), and the constraint-wise separability requirement (Ben-Tal *et al.* 2004, Marandi and Den Hertog 2018). This poses certain restrictions on the application scope of the corresponding results. In particular, it precludes the opportunities of modeling problems with certain types of natural constraints, such as capacity constraints in network optimization, budget constraints in a resource-limited setting, or problems with equality constraints where the right hand side vector is nonzero. To fill these gaps, we develop a theoretical framework that relaxes all such assumptions. Moreover, this framework provides new perspectives, understandings, and algorithms for analyzing adjustability. The summary of the main contributions follows,

Theory: We characterize a necessary and sufficient condition for adjustability to be zero using the relationship between certain direction vectors in the problem and some facets of the polyhedral uncertainty set, which provides a geometrically intuitive interpretation in the form of a theorem-of-the-alternatives. Based on this, we further derive a constructive approach to approximate adjustability for problems that can have general-shaped uncertainty sets. It turns out that the optimality criteria and gap derived using our framework are more general and tighter than existing results in the literature.

Algorithms: We design two algorithmic procedures to analyze adjustability. The first one is based on a mixed-integer linear program that can verify whether the adjustability gap is zero. The second algorithm computes a tight bound for adjustability using a type of geometric object called the anchor cone. We conduct extensive numerical experiments to analyze the efficiency of the proposed algorithms and the accuracy of the resulting bounds. We demonstrate that the algorithm based on the anchor cone can produce a tight bound within a short execution time.

The presentation of our results goes as follows. Section 2 provides the notation set and an exact account of the problem definition. In Section 3, we discover several equivalent reformulations of the RO and FARO problems and an algebraic property that bridges them together. In Section 4, we derive a necessary and sufficient condition for the adjustability gap to be zero. This result leads to an exact zero-adjustability verifier based on a mixed-integer program formulation. In Section 5, we introduce a constructive approach to characterize and efficiently approximate the adjustability ratio. Finally, in Section 6, we conduct numerical experiments to analyze the computation efficiency and approximation accuracy of the proposed algorithms. To better streamline the exposition of the paper, we only include proofs for the key results the main sections, and defer the others to Appendix A.

2 Preliminary

2.1 Notation

For an optimization problem Π, we use $z(\Pi)$ to denote the associated optimal objective value. Given $n \in \mathbb{N}$, $[n]$ is defined as the set $\{1, 2, \cdots, n\}$. For any subset $S \subseteq \mathbb{R}^n$, $\mathrm{int}(S)$ is the interior points of S, and we use $\mathrm{conv}(S)$ and $\mathrm{cone}(S)$ to represent the respective convex hull and conic hull formed by the elements in S. For a scalar $r \in \mathbb{R}$, we use rS to denote the scaled set $\{r\xi \mid \xi \in S\}$. Given any two sets of vectors S_1 and S_2, the Minkowski sum is defined as $S_1 + S_2 := \{v_1 + v_2 \mid v_1 \in S_1, v_2 \in S_2\}$. For a given polyhedron Ξ, $\mathrm{ext}(\Xi)$, $\mathrm{eray}(\Xi)$ are the sets of extreme points and extreme rays. Slightly abusing the notation, we use $\mathrm{conv}(\Xi) := \mathrm{conv}(\mathrm{ext}(\Xi))$, $\mathrm{cone}(\Xi) := \mathrm{cone}(\mathrm{eray}(\Xi))$ to represent the polytope part and cone part of Ξ. It is well known that every polyhedron Ξ can be decomposed into $\Xi = \mathrm{conv}(\Xi) + \mathrm{cone}(\Xi)$.

We use upper and lower case letters to denote matrices and vectors, respectively. For a matrix A, we take a_i and a_{ij} as the ith row and (i, j)th entry of A. We adopt the convention that all vectors without the transpose sign are column vectors. For instance, the ith row a_i of matrix A or any explicitly constructed vector (v_1, v_2) are all considered as column vectors, of which the row vector counterparts are denoted as $a_i^\intercal$ and $(v_1, v_2)^\intercal = (v_1^\intercal, v_2^\intercal)$. Given v_1, v_2 with the same size, we use the notation $[v_1, v_2]$ (separated by comma) to horizontally concatenate them into a two-column matrix. Similarly, vertical stacking is done by $[v_1^\intercal; v_2^\intercal]$ (separated by semi-colon). These two stacking operations naturally extend to multiple matrices and/or vectors with compatible shapes. We also view a matrix as the set of its rows, *i.e.*, $a \in A$ is some row in A (viewed as a column vector).

We define the inner product $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ in the common sense, *i.e.*, $\langle x_1, x_2 \rangle$ (or $\langle X_1, X_2 \rangle$ for two matrices) is the sum of the product of all the entries. We have the identities $\langle v_1, Av_2 \rangle = \langle A^\intercal v_1, v_2 \rangle$ and $\langle A, BC \rangle = \langle B^\intercal A, C \rangle = \langle AC^\intercal, B \rangle$.

2.2 Problem Setting

In this paper, we focus on games with a linear payoff function and linear coupling constraints. Then, with the input parameters (Ξ, a, A, c, C), the two general formulations I and II can be specified in the following forms,

$$\Pi := \max_{\xi \in \Xi} \min_{y \in \mathbb{R}^m} \langle c, \xi \rangle + \langle a, y \rangle \quad (1a) \qquad \bar{\Pi} := \min_{y \in \mathbb{R}^m} \max_{\xi \in \Xi} \langle c, \xi \rangle + \langle a, y \rangle \quad (2a)$$

$$\text{s.t. } C\xi - Ay \leq 0, \quad (1b) \qquad \text{s.t. } C\xi - Ay \leq 0, \quad \forall \xi \in \Xi. \quad (2b)$$

where Ξ is an uncertainty set embedded in $\mathbb{R}^n$. When Ξ is a polyhedron, we represent it as $\Xi := \{\xi \in \mathcal{X} \mid B\xi \leq b\}$. Also, we use k to indicate the number of constraints and define the augmented matrix $\bar{C} := [c^\intercal; C]$. Thus, c_0 is sometimes used to refer the input vector c. Throughout the paper, we assume $C \neq 0$, Ξ is a closed set, and both $\bar{\Pi}$ and Π are feasible and bounded. This directly leads to the following proposition.

Proposition 1. *The feasibility assumption implies (i) for every $\xi \in \Xi$ and $u \in \mathbb{R}^k_+$, if $A^\intercal u = 0$ then $\langle C\xi, u \rangle \leq 0$; (ii) for every $\xi \in cone(\Xi)$, $\langle c_i, \xi \rangle \leq 0$. The boundedness assumption entails (i) there exists some $u \in \mathbb{R}^k_+$ such that $A^\intercal u = a$; (ii) for every $\xi \in cone(\Xi)$, $\langle c, \xi \rangle \leq 0$.*

Proposition 1 shows that there always exists an optimal solution ξ in $\text{conv}(\Xi)$, even though we do not require Ξ to be bounded. In this case, we say Ξ is *effectively compact*. It also implies that the set $\{u \in \mathbb{R}^k_+ \mid A^\intercal u = a\}$ is nonempty. We define this polyhedron formally as,

Definition 1 (Dual Polyhedron)**.** *Given Π, the dual polyhedron is defined as $\mathcal{U} := \{u \in \mathbb{R}^k_+ \mid A^\intercal u = a\}$.*

Thus, the input parameters (Ξ, a, A, c, C) can also be represented as $(\Xi, \mathcal{U}, \bar{C})$. We will use the notation $\bar{\Pi} = (\Xi, \mathcal{U}, \bar{C})$ or $\Pi = (\Xi, \mathcal{U}, \bar{C})$ to indicate the two formulations with the specific input. In this setting, we define adjustability as follows.

Definition 2 (Adjustability)**.** *Given $\Pi = (\Xi, \mathcal{U}, \bar{C})$ and the corresponding $\bar{\Pi}$, we define* adjustability gap $\delta_{abs}(\Pi) := z(\bar{\Pi}) - z(\Pi)$ *and* adjustability ratio $\delta_{rel}(\Pi) := |z(\bar{\Pi})|/|z(\Pi)|$ *given $z(\Pi) > 0$ or $z(\bar{\Pi}) < 0$. When $\delta_{abs}(\Pi) = 0$, we say Π is* zero-adjustable*.*

The focus of this paper is to study the conditions for zero-adjustability and estimate a tight bound for adjustability ratio when these conditions are violated. In the next section, we will derive a general criterion.

3 Symmetry Gap and Symmetric Optimality

Our main goal is to quantify the gaps δ_{abs} and δ_{rel}. In this section, we focus on polyhedral uncertainty set Ξ and will convert δ_{abs} and δ_{rel} to the equivalent metrics defined on the following problem.

Definition 3 (Bidual Problem)**.** *Given* $\bar{\Pi}$ *with a polyhedral uncertanty set* Ξ*, the corresponding* bidual *and* symmetric bidual *problems are defined as*

$$\bar{\Delta} := \max_{\xi\in\Xi, u\in\mathcal{U}, V} \quad \langle c, \xi\rangle + \langle C, V\rangle \qquad (3a)$$
$$\text{s.t.} \quad BV^{\intercal} \leq bu^{\intercal}, \qquad (3b)$$

$$\bar{\Delta}^* := \max_{\xi\in\Xi, u\in\mathcal{U}, V} \langle c, \xi\rangle + \langle C, V\rangle \qquad (4a)$$
$$\text{s.t.} \; BV^{\intercal} \leq bu^{\intercal}, \qquad (4b)$$
$$V = u\xi^{\intercal}, \qquad (4c)$$

where the constraint set $V = u\xi^{\intercal}$ *is called the symmetry contraint.*

Definition 4 (Symmetric Optimality)**.** *A feasible solution* (ξ, u, V) *of the bidual* $\bar{\Delta}$ *is said to be symmetric if* $V = u\xi^{\intercal}$*. The* symmetry gap *and* symmetry ratio *are defined as* $\delta^*_{abs}(\bar{\Delta}) := z(\bar{\Delta}) - z(\bar{\Delta}^*)$ *and* $\delta^*_{rel}(\bar{\Delta}) := |z(\bar{\Delta})|/|z(\bar{\Delta}^*)|$*, respectively. We say* $\bar{\Delta}$ *is* symmetrically optimal *if* $\delta^*_{abs}(\bar{\Delta}) = 0$*.*

The requirement $V = u\xi^{\intercal}$ implies it is a rank-1 condition, yet the converse is false. The bidual problem $\bar{\Delta}$ is obtained from $\bar{\Pi}$ using the standard *bidualization reformulation* technique. First, we dualize each of the constraint. Then, we swap the minimization and maximization in $\bar{\Pi}$ citing the classical minimax theorem. Finally, we dualize the inner problem for a fixed ξ. Thus, we have the following lemma.

Lemma 1. *Given* $\bar{\Pi}$ *and its bidual* $\bar{\Delta}$*, we have* $z(\bar{\Pi}) = z(\bar{\Delta})$*.*

Using the properties of $\bar{\Delta}$ and $\bar{\Delta}^*$, we derive the following main result of this section.

Theorem 1. *Given* Π *with polyhedral uncertainty set, let* $\bar{\Delta}$ *be its bidual, we have* $\delta_{abs}(\Pi) = \delta^*_{abs}(\bar{\Delta})$ *and* $\delta_{rel}(\Pi) = \delta^*_{rel}(\bar{\Delta})$*.*

Proof. *First, we show that the dual of* Π*, denoted by* Δ*, is equivalent to the symmetric bidual,* i.e.*,* $\Delta = \bar{\Delta}^*$*. The formulation for* Π *is* (1)*. We fix the uncertainty variables at* ξ *and let* u *be the vector of dual variables for all the constraints. Then, dualizing the inner problem gives,*

$$\Delta := \max_{\xi\in\Xi, u\in\mathcal{U}} \langle c, \xi\rangle + \langle C\xi, u\rangle.$$

On the other hand, Formulation (4c) for $\bar{\Delta}^*$ can be written as

$$\begin{aligned} \max_{\xi \in \Xi, u \in \mathcal{U}} \ & \langle c, \xi \rangle + \langle C, u\xi^\intercal \rangle \\ s.t. \ & (B\xi - b)u^\intercal \leq 0. \end{aligned}$$

Notice that the constraint set $(B\xi - b)u^\intercal \leq 0$ is redundant since ξ and u are chosen from Ξ and $\mathcal{U}$, respectively. Hence, $\bar{\Delta}^* = \Delta$. Then, we have the following chain of relations,

$$z(\Pi) = z(\Delta) = z(\bar{\Delta}^*) \leq z(\bar{\Delta}) = z(\bar{\Pi}).$$

The first equality is due to strong duality, the second is by $\Delta = \bar{\Delta}^*$ we have just shown, the inequality is because $\bar{\Delta}^*$ has a more restricted feasible region than $\bar{\Delta}$, and the last equality is by Lemma 1. Therefore, the gap and ratio between $z(\Pi)$ and $z(\bar{\Pi})$ are entirely captured by the symmetry gap and ratio of $\bar{\Delta}$. □

Remark 1. *The dualization of Π is done by first fixing some $\xi \in \Xi$. Thus, we still have $z(\Pi) = z(\Delta)$ even when Ξ is not a polyhedron or not a convex set.*

The following corollary is a direct consequence of Theorem 1, so we omit its proof.

Corollary 1. Π *is zero-adjustable if and only if there exists an optimal solution of the bidual $\bar{\Delta}$ that is also symmetric.*

We will call these *symmetric-optimal solutions* hereafter. The next corollary allows us to restrict our attention to the extreme points of $\mathcal{U}$ and Ξ.

Corollary 2. Π *is zero-adjustable if and only if there exists a symmetric-optimal solution $(\xi^\star, u^\star, u^\star\xi^{\star\intercal})$ of the bidual $\bar{\Delta}$ such that $\xi^\star \in ext(\Xi)$ and $u^\star \in ext(\mathcal{U})$.*

Theorem 1 translates the adjustability gap δ_{abs} into the symmetry gap δ^*_{abs} that is defined upon the bidual problem $\bar{\Delta}$. Compared to the former, the latter gap is more advantageous for analytic purposes since it reveals an interesting structure, the symmetry constraint $V = u\xi^\intercal$, that dictates the adjustability gap. In later sections, we use this characterization to derive specific zero-adjustability criteria and adjustability ratio bounds.

4 When is Adjustability Gap Zero?

In this section, given polyhedral uncertainty set Ξ, we study the conditions under which $\delta_{\text{abs}} = 0$, *i.e.*, the zero-adjustability criteria of Π. In particular, we provide an exact characterization in the form of theorem-of-the-alternatives in 4.1, then develop an exact algorithmic verifier based on a mixed-integer optimization formulation in 4.2.

One interesting observation about Formulation (3) is that the constraint set (3b) is similar to the definition of Ξ. The following proposition formalizes this observation, which provides a geometric interpretation of (3b).

Proposition 2. *In bidual $\bar{\Delta}$, constraint set (3b) is equivalent to the following,*

$$v_i \in u_i \mathit{conv}(\Xi) + \mathit{cone}(\Xi),\ \forall i \in [k]. \tag{5}$$

Geometrically, this constraint says that the feasible region of the ith row of matrix V is the scaled polyhedron $u_i\text{conv}(\Xi) + \text{cone}(\Xi)$. Thus, constraint set (5) can be viewed as a *constraint propagator* that propagates each constraint of Ξ to the feasible space of each v_i with a scaling factor u_i.

This result allows us to represent each row v_i of matrix V as $u_i\xi_i + \xi_i'$ for some $\xi_i \in \text{conv}(\Xi)$ and $\xi_i' \in \text{cone}(\Xi)$. Then, the bidual formulation (3) has the following alternative form,

$$\bar{\Delta} = \max_{\xi' \in \text{cone}(\Xi)} \langle c, \xi' \rangle + \max_{\xi \in \text{conv}(\Xi)} \langle c, \xi \rangle + \sum_{i \in [k]} \max_{\xi' \in \text{cone}(\Xi)} \langle c_i, \xi_i' \rangle + \max_{u \in \mathcal{U}} \sum_{i \in [k]} u_i \left(\max_{\xi_i \in \text{conv}(\Xi)} \langle c_i, \xi_i \rangle \right).$$

By Proposition 1, the first and third terms above are equal to zero, which gives

$$\bar{\Delta} = \max_{\xi \in \text{conv}(\Xi)} \langle c, \xi \rangle + \max_{u \in \mathcal{U}} \sum_{i \in [k]} u_i \left(\max_{\xi_i \in \text{conv}(\Xi)} \langle c_i, \xi_i \rangle \right). \tag{6}$$

Thus, we can further replace (5) with $v_i \in u_i\text{conv}(\Xi)$ in the bidual $\bar{\Delta}$. With such intuitive interpretation of the constraints, it is expected that the feasible region of (3), denoted by $\mathfrak{P}$, has the following nice properties.

Proposition 3. *Any solution $(\xi, u, V) \in \mathfrak{P}$ is an extreme point if and only if $\xi \in \mathit{ext}(\Xi)$, $u \in \mathit{ext}(\mathcal{U})$, and for each row $v_i = u_i\xi_i$ of matrix V, either $u_i = 0$ or $\xi_i \in \mathit{ext}(\Xi)$.*

4.1 Zero-Adjustability Criteria

We use $\bar{\Delta}(u)$ to denote Formulation (6) with a fixed u. Then, $\mathcal{U}^\star := \arg\max_{u\in\mathcal{U}} \bar{\Delta}(u)$ is the set of optimal solutions in $\mathcal{U}$. According to (6), for a fixed u, $\bar{\Delta}(u)$ can be viewed as $k+1$ independent optimization problems, each of which is the *support function* (Rockafellar 1970, p. 28) over $\text{conv}(\Xi)$ with cost vector c or c_i. Based on this observation, we can use the following definition to characterize the zero-adjustability criterion.

Definition 5 (Normal Cone (Rockafellar 1970, p. 15))**.** *Vector c' is* normal *to a closed convex set Ξ at $\xi \in \Xi$ if $\langle c', \xi' - \xi\rangle \le 0$ for all $\xi' \in \Xi$. The set of all such vectors, $N_\Xi(\xi)$, is the* normal cone *to Ξ at ξ.*

When a set of cost vectors $\{c_i\}_{i\in L}$ belong to the same normal cone $N_\Xi(\xi^\star)$ for some $\xi^\star \in \Xi$, the optimization problems in $\{\max_{\xi\in\Xi}\langle c_i, \xi\rangle\}_{i\in L}$ share the same optimal solution $\xi^\star$. In this case, we call this family of problems *co-optimal*. It is clear that a family of maximization problems over the same Ξ is co-optimal if and only if the cost vectors belong to the same normal cone. With these definitions, we present the zero-adjustability theorem as follows.

Theorem 2. *$\Pi = (\Xi, \mathcal{U}, \bar{C})$ is zero-adjustable if and only if there exists some $\xi \in \Xi$ and $u \in \mathcal{U}^\star \cap ext(\mathcal{U})$ such that $c \in N_\Xi(\xi)$ and, for every $i \in [k]$, one of the following is true: (i) $u_i = 0$; (ii) $c_i \in N_\Xi(\xi)$.*

Proof. *By Proposition 2, Formulation (6) is equivalent to the bidual formulation (3) with the identities $v_i = u_i\xi_i$ for all $i \in [k]$. Then, every symmetric solution of (3) corresponds to a feasible solution $(\xi, u, \{\xi_i\}_{i\in[k]})$ of Formulation (6) that satifies $\xi = \xi_i$ for every index i such that $u_i > 0$. For sufficiency, we take u and ξ that satisfy the premise, and construct the solution $(\xi, u, \{\xi_i = \xi\}_{i\in[k]})$. This solution is optimal for (6) by the choice of u and ξ, thus corresponds to an optimal and symmetric solution for (3) since $\xi_i = \xi$ for all i. Then, by Corollary 1, Π is zero-adjustable. For necessity, according to Corollary 2, Π being zero-adjustable implies there exists a symmetric-optimal solution $(\xi^\star, u^\star, u^\star\xi^{\star\intercal})$ for (3) where $\xi^\star \in ext(\Xi)$ and $u^\star \in ext(\mathcal{U})$. Therefore, $(\xi^\star, u^\star, \{\xi_i = \xi^\star\}_{i\in[k]})$ is an optimal solution for (6). This further implies $\xi^\star$ is an optimal solution of the problems $\max_{\xi\in\Xi}\langle c, \xi\rangle$ and $\max_{\xi_i\in\Xi}\langle c_i, \xi_i\rangle$ for every index i such that $u_i > 0$,* i.e., *these problems are co-optimal. Thus, c and $\{c_i\}_{i\in L}$ belong to the same normal cone $N_\Xi(\xi^\star)$ where L labels all the nonzero entries in u.* □

The following corollary provides two sufficient criteria.

Corollary 3. $\Pi = (\Xi, \mathcal{U}, \bar{C})$ *is zero-adjustable if it satisfies either of the following conditions: (i)* $\{c\} \cup \{c_i\}_{i \in [k]} \subseteq N_\Xi(\xi)$ *for some* $\xi \in ext(\Xi)$*; (ii) for every* $u \in ext(\mathcal{U})$ *with nonzero entries labeled by* L_u*, we have* $\{c\} \cup \{c_i\}_{i \in L_u} \subseteq N_\Xi(\xi_u)$ *for some* $\xi_u \in ext(\Xi)$*.*

Figure 1 illustrate the relation between normal cone and zero-adjustability. Suppose all the rows in $\bar{C}$ are in the same normal cone, the adjustability gap is equal to zero according to the first condition in Corollary 3. Two trivial special cases follow directly: Ξ is a box set and rows in $\bar{C}$ are in the same orthant; Ξ is a L_1 ball and $\bar{C}$ has a *dominant column*, *i.e.*, the largest absolute value of each row in $\bar{C}$ is at the same entry and is of the same sign. Thus, in both of these cases, the corresponding problem Π is zero-adjustable.

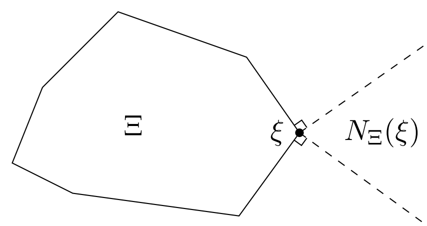

Figure 1: By Corollary 3, if all rows in $\bar{C}$ are in $N_\Xi(\xi)$, then Π is zero-adjustable.

For specific problems where the exact descriptions of both $\mathrm{ext}(\Xi)$ and $\mathrm{ext}(\mathcal{U})$ are accessible, Theorem 2 can often be used to produce more interesting optimality conditions for the specific problem at hand. In the next subsection, we extend the analytic capability of Theorem 2 and Corollary 3 using affine transformations.

4.2 Mixed Integer Linear Program Verifier for Zero-Adjustability

Using Theorem 2 and Corollary 3, we can verify the zero-adjustability of a class of problems that has certain special properties on the input $(\Xi, \mathcal{U}, C)$. In this subsection, we introduce an exact verifier for general inputs.

We assume that the values $\overline{\omega}_i := \max_{\xi \in \Xi} \langle c_i, \xi \rangle$ and $\underline{\omega}_i := \min_{\xi \in \Xi} \langle c_i, \xi \rangle$ can be efficiently computed, and use $\overline{\omega}$ and $\underline{\omega}$ to denote the corresponding vectors. Then, Formulation (6) can be equally written as

$$\bar{\Delta} = \max_{u \in \mathcal{U}} \langle \overline{\omega}, u \rangle + \overline{\omega}_0, \tag{7}$$

where $\overline{\omega}_0 := \max_{\xi \in \Xi} \langle c, \xi \rangle$ and $\mathcal{U}$ is the dual polyhedron $\{u \geq 0 \mid A^\intercal u = a\}$. Adding some extra constraints into (7), we obtain the following formulation.

$$\bar{\Delta}' := \max_{u,\xi,v,p_i} \langle \overline{\omega}, u \rangle + \overline{\omega}_0 \tag{8a}$$
$$\text{s.t. } A^\intercal u = a \tag{8b}$$
$$Mv \geq u \tag{8c}$$
$$v_0 = 1 \tag{8d}$$
$$\langle b, p_i \rangle \geq \underline{\omega}_i v_i, \quad \forall i \in \{0\} \cup [k] \tag{8e}$$
$$\langle b, p_i \rangle \leq \overline{\omega}_i v_i, \quad \forall i \in \{0\} \cup [k] \tag{8f}$$
$$\langle b, p_i \rangle \geq \langle c_i, \xi \rangle - \underline{\omega}_i (1 - v_i), \quad \forall i \in \{0\} \cup [k] \tag{8g}$$
$$\langle b, p_i \rangle \leq \langle c_i, \xi \rangle - \overline{\omega}_i (1 - v_i), \quad \forall i \in \{0\} \cup [k] \tag{8h}$$
$$B^\intercal p_i = c_i v_i, \quad \forall i \in \{0\} \cup [k] \tag{8i}$$
$$p_i \geq 0, \quad \forall i \in \{0\} \cup [k]. \tag{8j}$$
$$u \geq 0,\ \xi \in \Xi,\ v \in \{0,1\}^{k+1}. \tag{8k}$$

This is a mixed integer program with $(k+1)l+2k+n+1$ variables and $(n+5)(k+1)+m+l$ constraints (except for the nonnegativity or binary constraints) where m and n are the number of columns in the input matrices A and B, and k and l are the numbers of rows in the input matrices C and B, respectively. This formulation serves as a zero-adjustability verifier according to the following theorem.

Theorem 3. $\Pi = (\Xi, \mathcal{U}, \bar{C})$ *is zero-adjustable if and only if* $z(\bar{\Delta}) = z(\bar{\Delta}')$.

Proof. We prove this claim with two steps. First, we will show $\bar{\Delta}'$ is equivalent to the following formulation $\bar{\Delta}''$. Then, we show that Π is zero-adjustable if and only if $z(\bar{\Delta}'') = z(\bar{\Delta})$.

$$\bar{\Delta}'' := \max_{u,\xi,v} \langle \overline{\omega}, u \rangle + \overline{\omega}_0 \tag{9a}$$
$$\text{s.t. } A^\intercal u = a \tag{9b}$$
$$Mv \geq u \tag{9c}$$
$$v_0 = 1 \tag{9d}$$
$$v_i \langle c_i, \xi' - \xi \rangle \leq 0, \quad \forall i \in \{0\} \cup [k], \xi' \in \Xi \tag{9e}$$
$$u \geq 0,\ \xi \in \Xi,\ v \in \{0,1\}^{k+1}. \tag{9f}$$

To prove the first, we rewrite Constraint (9e) as $\max_{\xi' \in \Xi} v_i \langle c_i, \xi' \rangle \leq v_i \langle c_i, \xi \rangle$ for each i. Then, dualizing the left-hand-side and removing the minimization sign will produce Constraint sets (8i), (8j), and $\langle b, p_i \rangle \leq v_i \langle c_i, \xi \rangle$ for all $i \in \{0\} \cup [k]$. Since v_i is binary and $\langle c_i, \xi \rangle$ is upper and lower bounded by $\overline{\omega}_i$ and $\underline{\omega}_i$, using classic linearization technique produces (8e)–(8h).

For the second step, we note that Constraints (9c) and (9d) imply that v indicates the nonzero entries in u, and Constraints (9e) and (9f) mean that all c_i's corresponding to nonzero u_i's are in the same normal cone $N_\Xi(\xi)$. Let $u'^\star$ be the optimal solution obtained by solving $\bar{\Delta}''$, and use $\mathcal{U}^\star$ and $\mathrm{ext}(\mathcal{U})$ to denote the optimal solutions and extreme points of $\mathcal{U}$ in problem $\bar{\Delta}$, respectively. Suppose $z(\bar{\Delta}) \neq z(\bar{\Delta}'')$, then either $\bar{\Delta}''$ is infeasible or $u'^\star \notin \mathcal{U}^\star$. Both cases violate the sufficient condition of Theorem 2, which implies Π is not zero-adjustable. On the other hand, suppose $z(\bar{\Delta}) = z(\bar{\Delta}'')$, then $u'^\star \in \mathcal{U}^\star$. If $u'^\star$ is also an extreme point of $\mathcal{U}$, we are done by Theorem 2. Otherwise, $u'^\star$ is contained in the convex combination of some solutions $\{u_i^\star\} \subseteq \mathcal{U}^\star \cap \mathrm{ext}(\mathcal{U})$. Select an arbitrary $u^\star \in \{u_i^\star\}$, it should be clear that the nonzero entries in $u^\star$ is a subset of the ones in $u'^\star$ since $\mathcal{U}$ is nonnegative. Thus, the feasibility of $u'^\star$ in $\bar{\Delta}''$ also implies the feasibility of $u^\star$ in $\bar{\Delta}''$, *i.e.*, c and c_i's that correspond to nonzero entries of $u^\star$ are still in the normal cone $N_\Xi(\xi)$. Thus, the claim follows according to Theorem 2. □ □

Therefore, given an arbitrary input $\Pi = (\Xi, \mathcal{U}, \bar{C})$, comparing $z(\bar{\Delta})$ and $z(\bar{\Delta}')$ verifies the zero-adjustability of Π.

Remark 2. *$\bar{\Delta}$ is a reformulation of $\bar{\Pi}$, yet $\bar{\Delta}'$ is* not *a reformulation of Π. Instead, it is derived from Theorem 2. Hence, we* cannot *ensure the identity $z(\bar{\Delta}') = z(\Pi)$ whenever zero-adjustability fails.*

5 Adjustability Ratio

In this section, we examine the cases where the adjustability gap may be non-zero. In 5.1, we provide a constructive approach to characterize a bound of adjustability ratio. In 5.2, we present an algorithmic procedure to estimate the tightest bound under this approach. To the authors' knowledge, this is the first characterization and algorithmic approach to quantify adjustability ratio in such a general setting.

Recall that in the proof of Theorem 1, we have shown $z(\Pi) = z(\Delta) \leq z(\bar{\Delta}) = z(\bar{\Pi})$. Thus, the objective value of the original problem $z(\Pi)$ can be lower bounded by solving Δ restricted to some subset of Ξ as the uncertainty set; the value of the constant policy problem $z(\bar{\Pi})$ can be upper bounded by solving $\bar{\Delta}_{\Xi'}$ with some polyhedron $\Xi' \supseteq \Xi$. When this superset is properly constructed to satisfy the condition of Theorem 2 or Corollary 3, it becomes possible to estimate the adjustability ratio $\delta_{\text{rel}}(\Pi)$. In this section, Ξ is assumed to be a *closed set*.

5.1 Bound on Adjustability Ratio

The following result provides a constructive way to bound $\delta_{\text{rel}}(\Pi)$.

Theorem 4. *Given $\Pi = (\Xi, \mathcal{U}, \bar{C})$ where $z(\Pi) > 0$ (or $z(\Pi) < 0$), if there exists some polyhedron $\Xi' \supseteq \Xi$, some $\xi' \in \Xi'$, and a scalar $K \geq 1$ (or $0 < K \leq 1$) that satisfy $\bar{C} \subseteq N_{\Xi'}(\xi')$ and $\xi'/K \in \Xi$, then, we have the bound $1 \leq \delta_{rel}(\Pi) \leq K$ (or $1 \geq \delta_{rel}(\Pi) \geq K$).*

Proof. *Let $\bar{\Pi}'$ and Π' be the RO and FARO formulations with input parameters $(\Xi', \mathcal{U}, \bar{C})$. Clearly, we have $z(\bar{\Pi}') \geq z(\bar{\Pi}) \geq z(\Pi)$, where the first inequality is due to $\Xi \subseteq \Xi'$. The premise also states that all rows of $\bar{C}$ belong to the same normal cone of Ξ', then according to Corollary 3, Π' is zero-adjustable. That is, the corresponding bidual is symmetrically optimal. Thus, the optimal value is $z(\Pi') = z(\bar{\Pi}') = \max_{u \in \mathcal{U}} \langle c, \xi' \rangle + \langle C, u\xi'^{\intercal} \rangle$. On the other hand, we have $\xi'/K \in \Xi$. Thus, for any $u \in \mathcal{U}$, the solution $(\xi'/K, u, u\xi'^{\intercal}/K)$ is feasible to Δ — the dual of Π. Thus, we have the following lower bound,*

$$z(\Pi) = z(\Delta) \geq \max_{u \in \mathcal{U}} \langle c, \xi'/K \rangle + \langle C, u\xi'^{\intercal}/K \rangle = z(\bar{\Pi}')/K.$$

The first equality has been shown in the proof of Theorem 1 and is true for any closed set Ξ by Remark 1. Combining all these inequalities and $K > 0$, we get $Kz(\Pi) \geq z(\bar{\Pi}') \geq z(\bar{\Pi}) \geq z(\Pi)$. Finally, when $z(\Pi) > 0$ and $K \geq 1$, we have $Kz(\Pi) \geq z(\bar{\Pi}) \geq z(\Pi) > 0$; when $z(\Pi) < 0$ and $0 < K \leq 1$, we also have $z(\Pi) \leq z(\bar{\Pi}) \leq Kz(\Pi) < 0$. In both cases, the adjustability ratio $\delta_{rel}(\Pi)$ is well-defined and can be computed directly, which gives to the desired bound K. □

Remark 3. *Theorem 4 does not require Ξ to be a polyhedron. In fact, Ξ can be an arbitrary closed set, such as a discrete, scenario-based uncertainty set, which can arise from practical/data-driven robust optimization problems (Bertsimas et al. 2018, Shang et al. 2017).*

Figure 2 provides the intuition of Theorem 4. Given a non-convex space Ξ with $\bar{C}$ consists of vectors c, c_1, c_2, we estimate an upper bound of δ_{rel} by constructing a polyhedron Ξ' such

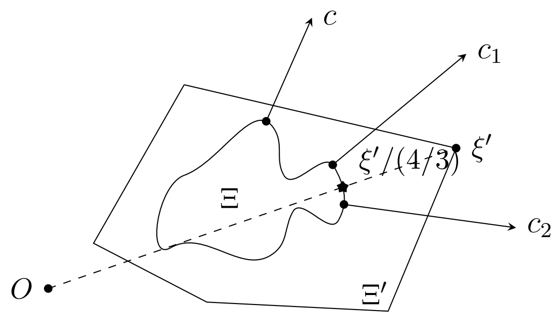

Figure 2: Bound on adjustability ratio for a non-convex Ξ. A better bound can be obtained by carefully constructing a tighter Ξ' (See Section 5.2).

that rows in $\bar{C}$ are in the same normal cone $N_{\Xi'}(\xi')$. Next, we scale and translate Ξ' to contain the original space Ξ. Then, any scalar k that satisfies $\xi'/k \in \Xi$ provides a valid upper bound of δ_{rel}. In this example, $4/3$ is the smallest upper bound under the constructed polyhedron Ξ'. However, a tighter bound can be obtained by delicately constructing a different Ξ'. The "optimal" type of polyhedron is called *anchor cone*, and will be introduced in Section 5.2.

In the following three examples, we derive closed-form upper bounds of δ_{rel} to illustrate the usage of Theorem 4. In these examples, we assume $z(\Pi) > 0$ so that the ratio is well-defined.

Example 1 (Convex $\Xi \subseteq \mathbb{R}^n_+$; $\bar{C} \geq 0$)**.** *In this case, the adjustability ratio $\delta_{rel}(\Pi)$ is bounded by n. To derive this, we will first use Theorem 4 to show a more general result. We define $X^\star := \prod_{i\in[n]} \arg\max_{\xi\in\Xi} \xi_i$ where ξ_i is the ith entry of ξ. Thus, each element $x = (x_i)_{i\in[n]} \in X^\star$ is a tuple of vectors, each of which is a maximizer along the ith axis. We use $\mu(x)$ to denote the number of unique vectors in x,* i.e.*, the cardinality of the set $\{x_i\}_{i\in[n]}$. Then, we can establish the following.*

Corollary 4. *Given $\Pi = (\Xi, \mathcal{U}, \bar{C})$ where $\bar{C} \geq 0$ and $\Xi \subseteq \mathbb{R}^n_+$ is convex, we have $\delta_{rel}(\Pi) \leq \mu(x)$ for every $x \in X^\star$.*

This directly gives $\delta_{rel}(\Pi) \leq n$ as n is a trivial upper bound of $\mu(x)$. This result improves previous result of $O(n)$ from the literature (Bertsimas and Goyal 2010). It is also easy to show that n is a tight bound since a simplex Ξ will achieve this bound exactly. More generally, the class of budget sets $\Xi := \{\xi \in [0,1]^n \mid \langle 1, \xi\rangle \leq \beta n\}$ for $\beta \in [0,1]$ serves as a natural transition between a simplex (when $\gamma = 0$) and a box set (when $\gamma = 1$). Thus, the corresponding adjustability ratio upper bound also decreases from n to 1 continuously as γ sliding from 0 to 1. △

Example 2 (Convex Lattice $\Xi \subseteq \mathbb{R}^n_+$; $\bar{C} \geq 0$)**.** *With the extra lattice property,* i.e., $\xi_1 \vee \xi_2 \in \Xi$ *for every* $\xi_1, \xi_2 \in \Xi$ *where* $\vee$ *is the entry-wise maximum operator, the adjustability bound can be further tightened as* $\min(\dim(\Xi) + 1, n)$*. The* n *part has been established in the previous example. Suppose* $\dim(\Xi) + 1 < n$*, the following corollary shows that for every* $x \in X^\star$ *such that* $\dim(\Xi) + 1 < \mu(x) \leq n$*, we can construct another* $x' \in X^\star$ *with* $\mu(x') < \mu(x)$*.*

Corollary 5. *Given* $\Pi = (\Xi, \mathcal{U}, \bar{C})$ *where* $\bar{C} \geq 0$ *and* $\Xi \subseteq \mathbb{R}^n_+$ *is a convex lattice, suppose* $\dim(\Xi) + 1 < n$*, then for every* $x \in X^\star$ *such that* $\mu(x) > \dim(\Xi) + 1$*, there exists some* $x' \in X^\star$ *such that* $\mu(x') < \mu(x)$*.*

The proof can be found in Appendix A. This result along with Corollary 4 establish the bound $\min(\dim(\Xi) + 1, n)$*.* △

When we know the specific description of the uncertainty set Ξ, tighter bounds can be derived using Theorem 4.

Example 3 (Ellipsoidal Ξ; $\bar{C} \geq 0$)**.** *An ellipsoidal uncertainty set can be defined as* $\Xi = \{\xi \mid \sum_{i\in[n]} \xi_i^2/l_i^2 \leq 1\}$ *for some* $l = (l_i)_{i\in[n]} > 0$*. We take* Ξ' *as the box set that circumscribes* Ξ*,* i.e., $\Xi' = \prod_{i\in[n]}[-l_i, l_i]$*. Because* $\bar{C} \geq 0$*, all the row vectors of* $\bar{C}$ *belong to the normal cone* $N_{\Xi'}(l)$*. Then, the intersection point between the line segment* $[0, l]$ *and the b undary* $\partial\Xi = \{\xi \mid \sum_{i\in[n]} \xi_i^2/l_i^2 = 1\}$ *can be directly computed as* $l/\sqrt{n}$*. Applying Theorem 4, we get* $\delta_{rel}(\Pi) \leq \sqrt{n}$ *for ellipsoids that are centered at the origin.*

The same technique can be used to derive closed-form bounds for translated and/or rotated ellipsoidal uncertainty sets. For instance, for a given ellipsoidal set $\Xi = \{\xi \mid \sum_{i\in[n]} \xi_i^2/l_i^2 \leq 1\}$*, let* x *be the intersection point of the line segment* $[0, l]$ *and the boundary of* Ξ*, we can easily derive a tight bound of* δ_{rel} *for the translated ellipsoidal set* $\Xi' = \{\xi \mid \sum_{i\in[n]}(\xi_i - \lambda l_i)^2/l_i^2 \leq 1\}$ *for some* $\lambda \geq 0$*. We construct the tightest box set whose maximal point is* $\lambda l + l$ *to enclose* Ξ'*. Then, we get the following bound*

$$\delta_{rel} \leq \frac{\|\lambda l\|_2 + \|l\|_2}{\|\lambda l\|_2 + \|x\|_2} \leq \frac{\|\lambda l\|_2 + \|l\|_2}{\|\lambda l\|_2 + \frac{\|l\|_2}{\sqrt{n}}} = \frac{(\lambda+1)\sqrt{n}}{\lambda\sqrt{n}+1} = 1 + \frac{\sqrt{n}-1}{\lambda\sqrt{n}+1},$$

where the first inequality is obtained by direct computation (notice λl*,* l*, and* x *are in the same direction), the second is due to the result* $\|l\|_2/\|x\|_2 \leq \sqrt{n}$ *from the above example.* △

We can also apply Theorem 4 to other uncertainty sets and input matrix $\bar{C}$. In particular, it can be similarly computed that when Ξ is ellipsoidal and the jth column of $\bar{C}$ is a dominant column, δ_{rel} is bounded by $\|l\|_2/(2l_j)$; when Ξ is the budgeted set $\{\xi \in [-1,1]^m \mid \|\xi\|_1 \leq \Gamma\}$, δ_{rel} is bounded by n/Γ given $\bar{C} \geq 0$, and is bounded by Γ given $\bar{C}$ has a dominant column.

5.2 Algorithmic Estimation of Adjustability Ratio

We have shown that Theorem 4 can be used to analytically study the bounds for the adjustability ratio δ_{rel}. In this subsection, we formalize this idea into a numerical method, called the *anchor cone algorithm*, that produces the tightest bound for δ_{rel} accordingly. Throughout this subsection, we assume that the problem $\max_{\xi\in\Xi}\langle c, \xi\rangle$ can be efficiently solved for every possible vector c or there exists an oracle.

Anchor cone is a special class of polyhedrons that we choose to construct Ξ' in Theorem 4 for producing a valid bound. We define it as follows.

Definition 6 (Anchor Cone)**.** *Given a finite set of vectors $\mathcal{C} = \{c_i\}_{i\in L}$ and a point x_0, we define the corresponding anchor cone as $\mathfrak{A}_{\mathcal{C},x_0} := \{x \mid \langle c_i, x\rangle \leq \langle c_i, x_0\rangle,\ \forall i \in L\}$ where x_0 is called the* anchor *of $\mathfrak{A}_{\mathcal{C},x_0}$.*

By this definition, an anchor cone is a convex set constructed by anchoring a cone at x_0. This constitutes a more liberal use of the concept of *cone* than conventionally done, since our "cone" may not be anchored at the origin. We nonetheless keep this name for its geometric intuition. It has several interesting properties that will be used later. For cone($\mathcal{C}$), we use cone$^*(\mathcal{C})$ and cone$^\circ(\mathcal{C})$ to denote the corresponding dual and polar cones.

Proposition 4. *Every anchor cone $\mathfrak{A}_{\mathcal{C},x_0}$ has the following properties: (i) $\mathfrak{A}_{\mathcal{C},x_0} = \{x_0\} +$ cone$^\circ(\mathcal{C})$; (ii) $N_{\mathfrak{A}_{\mathcal{C},x_0}}(x_0) =$ cone$(\mathcal{C})$; (iii) constraints of $\mathfrak{A}_{\mathcal{C},x_0}$ that correspond to vectors in eray$(\mathcal{C})$ are sufficient to define $\mathfrak{A}_{\mathcal{C},x_0}$.*

Take J as the index set for eray($\bar{C}$) and let $\omega_j := \max_{\xi\in\Xi}\langle c_j, \xi\rangle$ for every $j \in J$, then, the *anchor cone formulation* is defined as,

$$\Lambda := \min_{\gamma,x,\xi\in\Xi} (\text{or } \max) \quad \gamma \tag{10a}$$

$$\text{s.t.} \quad \langle c_j, x_k\rangle \leq \langle c_j, \gamma\xi\rangle, \quad \forall j,k \in J, \tag{10b}$$

$$\langle c_j, x_j\rangle \geq \omega_j, \quad \forall j \in J, \tag{10c}$$

$$\gamma \geq 1 \text{ (or } \gamma \leq 1), \quad x \text{ free.} \tag{10d}$$

The minimization with $\gamma \geq 1$ and maximization with $\gamma \leq 1$ are designed for the two cases $z(\Pi) > 0$ and $z(\Pi) < 0$, respectively. The main idea of this formulation is to search for an element $\xi \in \Xi$ and an optimized positive scalar γ such that the anchor cone $\mathfrak{A}_{\bar{C},\gamma\xi}$ contains Ξ. The first constraint set says that every x_k belongs to the anchor cone $\mathfrak{A}_{\bar{C},\gamma\xi}$. The second constraint set ensures that $\mathfrak{A}_{\bar{C},\gamma\xi} \supseteq \Xi$, since, according to the last property in Proposition 4, the constraints labeled by J are sufficient to define $\mathfrak{A}_{\bar{C},\gamma\xi}$. Constraint set (10c) ensures every such constraint contain Ξ, so the anchor cone $\mathfrak{A}_{\bar{C},\gamma\xi}$ also contains Ξ.

The next two theorems indicate that the anchor cone is the ideal polyhedron to produce the tightest bound of δ_{rel} under Theorem 4.

Theorem 5. *Given $\Pi = (\Xi, \mathcal{U}, \bar{C})$, let γ be any feasible solution of the corresponding anchor cone formulation. We have $\delta_{rel}(\Pi) \leq \gamma$ when $z(\Pi) > 0$ and $\delta_{rel}(\Pi) \geq \gamma$ when $z(\Pi) < 0$.*

Proof. *When $\bar{C}$ is full-rank, the anchor cone $\mathfrak{A}_{\bar{C},\gamma\xi}$ has the unique extreme point $\gamma\xi$. By the second property in Proposition 4, all vectors in $\bar{C}$ lead to $\gamma\xi$. Then, applying Theorem 4 proves the claim. When $\bar{C}$ is not full-rank, the anchor cone $\mathfrak{A}_{\bar{C},\gamma\xi}$ has no extreme point. However, the uncertain sets of both Π and $\bar{\Pi}'$ can be projected without changing the corresponding optimal objective values. That is, $z(\Pi)$ and $z(\bar{\Pi}')$ will not change if we replace Ξ and $\Xi' = \mathfrak{A}_{\bar{C},\gamma\xi}$ with $proj_{\bar{C}}(\Xi)$ and $proj_{\bar{C}}(\Xi')$ where $proj_{\bar{C}}(\cdot)$ projects the input onto the subspace spanned by $\bar{C}$. Then, it is clear that $proj_{\bar{C}}(\Xi')$ has the unique extreme $\gamma\xi$ where all the vectors in $\bar{C}$ are leading to. Thus, we can again apply Theorem 4 to prove the bound γ.* □

Theorem 6. *Given $\Pi = (\Xi, \mathcal{U}, \bar{C})$, let γ be the optimal value of (10) and let K be any bound calculated using Theorem 4 with some polyhedron Ξ', then, γ is a tighter bound than K.*

Proof. *We will show that every such Ξ' corresponds to a solution $(\gamma_0, \xi, \{x_j\}_{j\in J})$ of Formulation (10). Because $\Xi' \supseteq \Xi$, we can find $\{x_j\}_{j\in J} \subseteq \Xi'$ that satisfy constraint set (10c). Let $\xi' \in \Xi'$ be the extreme point that all vectors in $\bar{C}$ lead to. According to Theorem 4, $\xi'/K \in \Xi$. Let $\gamma_0 = K$ and $\xi = \xi'/K$, we have $\xi' = \gamma_0\xi$. Since all vectors in $\bar{C}$ are in the normal cone $N_{\Xi'}(\gamma_0\xi)$ and $\{x_j\}_{j\in J} \subseteq \Xi'$ by selection, we have $\langle c_j, x_k - \gamma_0\xi\rangle \leq 0$ for all $j, k \in J$ by the definition of normal cone, which is the same as constraint set (10b). Thus, $(K, \xi'/K, \{x_j\}_{j\in J})$ is a feasible solution of Formulation (10). This concludes the proof.* □

Notice that Formulation (10) is nonlinear due to the term $\gamma\xi$. However, we can still solve it efficiently using a binary line search algorithm on γ (see Algorithm 1) where at each

Algorithm 1: Calculating optimal bound for $\delta_{\text{rel}}(\Pi)$ given $z(\Pi) > 0$.

Data: Problem data $\Xi, \mathcal{U}, \bar{C}$; solution tolerance ϵ; M; number of steps T

initialization: $t \leftarrow 1$, $\underline{\gamma} \leftarrow 1$, $\overline{\gamma} \leftarrow M$, `hasSolution` = FALSE;

while $t < T$ *and* $\overline{\gamma} - \underline{\gamma} > \epsilon$ **do**

  $\gamma \leftarrow (\overline{\gamma} + \underline{\gamma})/2$;

  if *Anchor Cone Formulation is feasible with* γ **then**

    $\overline{\gamma} \leftarrow \gamma$; `hasSolution` $\leftarrow$ TRUE;

  end

  else

    $\underline{\gamma} \leftarrow \gamma$

  end

  $t \leftarrow t + 1$;

end

return γ, `hasSolution` ;

iteration, only a feasibility check is required. This binary line search algorithm is justified since given $z(\Pi) > 0$ ($z(\Pi) < 0$), the anchor cone $\mathfrak{A}_{\bar{C},\gamma\xi}$ is increasing (decreasing) on γ under the inclusion relation $\subseteq$. Therefore, the complexity of solving Formulation (10) is $O(\Lambda_\gamma \log_2 \frac{\bar{\gamma}-1}{\epsilon})$ where $\bar{\gamma}$ is some known upper bound of γ, scalar $\epsilon > 0$ is a given accuracy tolerance, and Λ_γ is the complexity of solving the anchor cone formulation Λ with a fixed γ. For instance, when Ξ is a polyhedron, Λ_γ is the time complexity of solving a linear program with $O(n \times |\text{eray}(\bar{C})|)$ variables and $O(|\text{eray}(\bar{C})|^2)$ constraints; when Ξ is convex, it is the complexity of solving a convex optimization with the same size. The next two propositions provide the feasibility/infeasibility criteria for Formulation (10).

Proposition 5. *Given* Ξ *is bounded, Formulation* (10) *is feasible if either* $\Xi \cap int(cone^*(\bar{C})) \neq \varnothing$ *or* $\Xi \subseteq int(cone^\circ(\bar{C}))$.

Implicitly, the first condition is associated with the case $z(\Pi) > 0$, while the second is for $z(\Pi) < 0$. These two conditions are not necessary. Hence, even this proposition is violated, Formulation (10) may still be feasible to produce a valid bound for the adjustability ratio.

Proposition 6. *Formulation* (10) *is infeasible if* $\dim(\Xi) + \dim(-cone(\bar{C}) \cap cone(\bar{C})) > n$.

The main tool for this proof is the equality $\dim(\text{cone}^\circ(\bar{C})) = n - \dim(\hat{\mathcal{C}})$ where $\hat{\mathcal{C}}$ denotes $-\text{cone}(\bar{C}) \cap \text{cone}(\bar{C})$. Thus, when vectors in $\bar{C}$ cannot lie in some halfspace, we have $\text{cone}^\circ(\bar{C}) = \{0\}$, which means the anchor cone is a single point. On the other hand, when vectors in $\bar{C}$ lie in the interior of some halfspace, $\hat{\mathcal{C}} = \{0\}$, which implies the anchor cone is full-dimensional.

Finally, we finish the section with a special case where the anchor cone formulation has an analytical solution.

Example 4 (cone($\bar{C}$) $= \mathbb{R}^n_+$ and $z(\Pi) > 0$)**.** *In this case, cone($\bar{C}$) is self-dual. Thus, for any $\xi \geq 0$, the anchor cone $\mathfrak{A}_{\bar{C},\xi}$ is simply obtained by removing all the lower bounds from a box set, which leaves the only extreme point ξ. Then, Formulation (10) reduces to*

$$\min_{\gamma \geq 1, x, \xi \in \Xi} \gamma \tag{11a}$$

$$s.t.\ x_j^k \leq \gamma \xi_j \quad \forall j, k \in J, \tag{11b}$$

$$x_j^j \geq \bar{\xi}_j \quad \forall j \in J. \tag{11c}$$

where $\bar{\xi}_j = \max_{\xi \in \Xi} \xi_j$ is the maximum value of the jth entry of ξ. Clearly, the variables x can be further reduced. The feasibility of the resulting formulation depends on Ξ. Suppose Ξ contains some element $\xi > 0$, this formulation is always feasible and bounded, which can be further reduced to

$$\min_{\xi \in \Xi} \max_j \frac{\bar{\xi}_j}{\xi_j} \iff \left(\max_{\xi \in \Xi} \min_j \frac{\xi_j}{\bar{\xi}_j} \right)^{-1}.$$

Depending on the uncertainty set Ξ, this can be computed directly to derive an analytical expression of the adjustability ratio bound. △

6 Computational Experiments

The experiments reported in this section were conducted on an N2 virtual machine instance on the Google Cloud Platform equipped with 16 virtual CPUs, 128 GB of RAM, and running on Linux x86_64, Debian 5.10. All the formulations and algorithms were implemented in Python 3.9 and solved using the commercial optimizer Gurobi 9.5.1. Each instance was solved under a time limit of 3,600 seconds and an optimality gap tolerance of 10^{-3}.

We focus on analyzing the efficiency of the anchor cone (AC) algorithm and the quality of the resulting bound by conducting comparison experiments on two sets of randomly generated instances. The baseline algorithm to compare with is the direct computation of the two bidual formulations (3) and (4) using Gurobi 9.5.1, which will be called the BD algorithm hereafter. In this algorithm, we first solve the linear program (3) exactly with a value $z(\bar{\Delta})$, then solve the bilinear program (4) with value $z(\bar{\Delta}^*)$ under the given optimality tolerance 10^{-3}. If both problems are solved within the time limit, $\gamma := z(\bar{\Delta})/z(\bar{\Delta}^*)$ is then an upper bound of the adjustability ratio with a gap less than the given tolerance 10^{-3}. We note that, in Gurobi 9.5.1, bilinear programs are solved using the state-of-art branch-and-bound algorithm based on the McCormick envelopes (McCormick 1976).

In each experiment, we record runtime t, optimality gap δ, and objective value γ (upper bound of adjustability ratio) of the AC and BD algorithms with subscripts *ac* and *bd*, respectively. Hence, the comparison ratios t_{ac}/t_{bd} and γ_{ac}/γ_{bd} can indicate the computation efficiency and approximation accuracy of the AC algorithm, where having small values in both comparison ratios implies an efficient and accurate performance.

At time t of solving the bilinear program (4), we can retrieve the incumbent solution with value $z'_t(\bar{\Delta}^*)$. Then, the induced value $\gamma'_t := z(\bar{\Delta})/z'(\bar{\Delta}^*)$ is a valid upper bound for the adjustability ratio. To provide a fair comparison between the two algorithms and make the performance criteria more stringent for the AC algorithm, we also record the time $t'_{bd} := \min_{\gamma'_t \leq \gamma_{ac}} t$, *i.e.*, the earliest time that the BD algorithm obtains an incumbent bound γ'_t that is at least as good as γ_{ac}. Then, t_{ac}/t'_{bd} indicates the efficiency of the AC algorithm.

6.1 Test Instances

In terms of the adjustability ratio, it is obvious from Formulation (3) and (4) that the size of the polytope Ξ and the magnitude of rows in $\bar{C}$ are inessential as setting $\Xi' := \Xi/K_1$ and $\bar{C}' := \bar{C}/K_2$ for some $K_1, K_2 > 0$ will lead to the same ratio. Thus, we randomly generate polytopes Ξ within the hypercube $[-1, 1]^n$ and rows of $\bar{C}$ with lengths that are less than a unit. To guarantee $z(\Pi) > 0$ so that the adjustability ratio $\gamma \geq 1$, we ensure that (i) the polytopes contain at least one positive vector as their relative interior, (ii) $\bar{C} \geq 0$, and (iii) $\mathcal{U}$ is the product space of a set of simplices, *i.e.*, $a = 1$ and A is a diagonal block matrix where each diagonal block is an all-one vector of varied lengths.

More specifically, to generate Ξ, we use n to denote the environment dimension of Ξ and l to denote the number of constraints that define Ξ. First, we add the $2n$ constraints that define the hypercube $[-1, 1]^n$. Then, we randomly generate $l - 2n$ vectors b_i's within the hypercube and include the associated half-spaces $\langle b_i, \xi \rangle \leq \|b_i\|_2$ as constraints of Ξ. For $\mathcal{U}$, we use m to denote the number of simplices (*i.e.*, the environment dimension of $\mathcal{Y}$), then

Table 1: Number of variables and constraints in associated formulations.

Algorithm	Formulation	Variables	Linear Constraints	Bilinear Constraints
AC	(10) with a fixed γ	$n(k+1)+1$	$k(k+1)+l$	0
BD	(3)	$n(k+1)+n$	$l(k+1)+m$	0
	(4)	$n(k+1)+n$	$l+m$	nk

generate each simplex j with a dimension d_j. Hence, $k := \sum_{j\in[m]} d_j$ is the number of rows in A. This entire process produces a valid input $(\Xi, \mathcal{U}, \bar{C})$. Based on this generation scheme, we created two sets of instances as follows,

- S1: n ranges from $\{5, 10, \ldots, 25\} \cup \{30, 40, \ldots, 100\}$ and m/n ranges from $\{1, 2\}$. For each configuration (n, m), we randomly generate 10 instances where l/n is randomly selected from $\{3, \ldots, 9\}$ and each simplex has a random dimension from $\{1, \ldots, 5\}$. There are 26 configurations and 260 instances in total.

- S2: Ξ has an extra budget constraint $\langle 1, \xi \rangle \leq \beta n$ where $\beta \in (0, 1)$ controls the tightness of the budget. n ranges from $\{5, 10, \ldots, 50\}$, m is fixed at $\lfloor 1.5n \rfloor$, and β ranges from $\{0.1, 0.3, \ldots, 0.9\}$. For each configuration (n, m, β), we randomly generated 5 instances where l/n and each simplex's dimension are randomly selected from the same sets as before. There are 50 configuration and 250 instances in total.

In S1, we study the performance of the two algorithms on randomly generated uncertainty polytope Ξ. We are especially interested in the differences between running on small instances with $n \in \{5, 10, \ldots, 25\}$ and relatively large instances $n \in \{30, 40, \ldots, 100\}$. In S2, we focus on random polytopes with a tighter budget constraint. We intend to observe how this extra constraint would affect the performances of both algorithms.

Finally, for a configuration (n, m) in S1 with l and k defined above, the sizes of the associated formulations are listed in Table 1. In particular, the potentially largest instance in S1 will induce a Formulation (10) with 100,101 variables and 1,001,900 constraints, a Formulation (3) with 100,200 variables and 901,100 constraints, a Formulation (4) with the same number of variables, 1,100 linear constraints, and 100,000 bilinear constraints.

6.2 Results on Instance Set S1

Table 2 reports the results of the AC and BD algorithms on the instance set S1. All the values are averaged over ten instances in each configuration. According to column γ_{bd}, the adjustability ratios of these instances are quite small. Indeed, the maximum γ_{bd} across all configurations is mere 1.10. This implies that for a randomly generated polytope Ξ, the constant policy is likely to perform reasonably well, at least under our random generation scheme. The upper bound γ_{ac} computed by the AC algorithm is also close to the true adjustability ratio since the comparison ratios γ_{ac}/γ_{bd} are mostly within 1.5 except the small configuration $(5, 10)$. Another trend is that the approximation accuracy of the AC algorithm generally improves when the instance gets larger. For instance, when n is at the low level 5, the comparison ratios are the two largest values 1.41 and 2.04; when n is raised to 100, the corresponding comparison ratios decrease to 1.08 and 1.07. One contributing factor is that small instances are more likely to have "low-budget-constraints" that can induce a large upper bound for the AC algorithm, according to the discussion in Example 1.

In terms of runtime, the BD algorithm is quite inefficient according to the columns t_{bd}, t'_{bd}, and δ_{bd}. This is expected since Formulation (4) in the BD algorithm is a nonlinear program, thus could be NP-hard in general. In contrast, given any fixed tolerance, the AC algorithm is implemented as a binary search composed with a linear program in each iteration, thus is polynomially solvable. This is also consistent with the results in column t_{ac}. Another observation is that doubling the value m affects the runtime of the AC algorithm more apparently than the BD algorithm. This is due to the fact that k is of order $O(m)$. Thus, according to Table 1, the number of linear constraints in the AC and BD algorithms is of order $O(m^2)$ and $O(m)$. Finally, in all tested instances in S1, the AC algorithm only takes a fraction runtime of t'_{bd}, while the trend shows that the comparison ratio t_{ac}/t'_{bd} gets larger when the instance size increases, especially when m is at the high level.

6.3 Results on Instance Set S2

Instance set S2 differs from S1 mainly in that the uncertainty set of every instance in the former has an additional budget constraint $\langle 1, \xi \rangle \leq \beta n$ for some $\beta \in \{0.1, 0.3, \ldots, 0.9\}$. In this subsection, we analyze how this extra constraint will affect the performances of the two algorithms. The corresponding computational results are displayed in Table 3.

Table 2: Computational results of instance set S1. The AC algorithm provides a good estimation of the exact adjustability ratio since $\gamma_{ac}/\gamma_{bd} \approx 1$ in most cases.

Configuration		Adjustability Ratio			Time				Gap(%)
n	m	γ_{ac}	γ_{bd}	γ_{ac}/γ_{bd}	t_{ac}	t_{bd}	t'_{bd}	t_{ac}/t'_{bd}	δ_{bd}
5	5	1.46	1.04	1.41	0.08	0.92	0.92	0.09	0.00
	10	2.12	1.04	2.04	0.07	1.35	1.35	0.06	0.00
10	10	1.35	1.09	1.24	0.11	4.02	4.02	0.03	0.00
	20	1.30	1.07	1.22	0.29	4.80	4.80	0.06	0.02
15	15	1.35	1.08	1.24	0.25	7.58	7.55	0.03	0.02
	30	1.27	1.07	1.19	0.86	13.04	12.91	0.07	0.03
20	20	1.23	1.08	1.14	0.54	14.74	14.27	0.04	0.03
	40	1.30	1.09	1.19	2.09	30.78	28.15	0.07	0.05
25	25	1.14	1.05	1.08	1.12	16.24	15.91	0.07	0.00
	50	1.20	1.06	1.13	4.55	61.77	53.70	0.08	0.06
30	30	1.24	1.09	1.14	2.20	46.32	37.11	0.06	0.03
	60	1.24	1.09	1.14	9.38	188.86	107.84	0.09	0.08
40	40	1.18	1.08	1.09	5.72	156.65	87.19	0.07	0.04
	80	1.21	1.09	1.11	24.13	548.73	240.76	0.10	0.08
50	50	1.20	1.08	1.10	12.56	352.66	193.87	0.06	0.07
	100	1.20	1.08	1.11	64.39	1,451.30	390.57	0.16	0.06
60	60	1.16	1.08	1.07	26.72	972.07	315.28	0.08	0.19
	120	1.17	1.08	1.08	124.11	1,252.17	295.53	0.42	0.08
70	70	1.20	1.10	1.09	51.00	3,200.20	1,029.84	0.05	0.80
	140	1.18	1.08	1.09	242.02	2,509.92	431.02	0.56	1.16
80	80	1.18	1.09	1.08	85.68	2,714.96	1,420.24	0.06	1.46
	160	1.14	1.07	1.07	408.25	2,326.94	420.63	0.97	0.25
90	90	1.13	1.07	1.05	144.08	2,412.98	356.60	0.40	0.86
	180	1.16	1.08	1.08	796.25	2,776.07	850.84	0.94	1.83
100	100	1.14	1.08	1.06	228.27	2,629.08	589.30	0.39	1.33
	200	1.15	1.07	1.07	1,256.71	3,463.28	1,307.23	0.96	1.29

The first observation is that both upper bounds γ_{ac} and γ_{bd} become worse as the budget decreases, yet the upper bound from the AC algorithm deteriorates faster according to the comparison ratio γ_{ac}/γ_{bd}. Nonetheless, in most cases, γ_{ac} still provides a tight estimate of the adjustability ratio. For instance, under most configurations, γ_{ac}/γ_{bd} is within 2 and the largest comparison ratio is 3.03 at the configuration with the smallest size $(5, 7)$ and lowest budget 0.1. Another assurance regarding the approximation quality of the AC algorithm is that at each fixed β, the ratio γ_{ac}/γ_{bd} is trending downward, implying that the upper bound γ_{ac} gets better when the instance size increases.

For the runtime, it is apparent that exactly solving the BD algorithm becomes more difficult compared to similar configurations in Table 2. Even at the small-sized configuration $(20, 30, 0.1)$, the average runtime hits the limit and has an average optimality gap 13%, while

Table 3: Computational results of instance set S2. The AC algorithm provides a good estimation of the exact adjustability ratio since $\gamma_{ac}/\gamma_{bd} \geq 1.5$ in most cases and the quality gets better for larger instances and higher budget.

Configuration			Adjustability Ratio			Time				Gap(%)
n	m	β	γ_{ac}	γ_{bd}	γ_{ac}/γ_{bd}	t_{ac}	t_{bd}	t'_{bd}	t_{ac}/t'_{bd}	δ_{bd}
5	7	0.1	4.38	1.45	3.03	0.14	1.12	0.10	1.38	0.04
		0.3	1.75	1.11	1.57	0.06	1.11	0.10	0.58	0.00
		0.5	1.58	1.13	1.40	0.06	1.09	0.08	0.71	0.00
		0.7	1.61	1.06	1.51	0.06	1.05	0.05	1.20	0.00
		0.9	3.43	1.26	2.72	0.06	1.06	0.06	0.89	0.00
10	15	0.1	5.26	1.80	2.93	0.24	6.68	0.61	0.39	0.00
		0.3	2.26	1.35	1.67	0.21	5.87	0.61	0.34	0.02
		0.5	1.53	1.14	1.34	0.20	4.99	0.58	0.35	0.02
		0.7	1.35	1.12	1.20	0.18	4.97	0.58	0.31	0.02
		0.9	1.51	1.08	1.40	0.19	4.74	0.48	0.41	0.00
15	22	0.1	5.44	1.97	2.77	0.83	314.93	2.09	0.40	0.00
		0.3	2.15	1.42	1.51	0.60	41.13	2.17	0.28	0.00
		0.5	1.55	1.15	1.34	0.58	13.68	1.96	0.29	0.04
		0.7	1.36	1.08	1.26	0.48	12.30	1.98	0.24	0.02
		0.9	1.19	1.06	1.12	0.42	11.45	1.93	0.22	0.02
20	30	0.1	5.19	2.02	2.57	2.04	3,600.01	6.09	0.34	12.52
		0.3	2.29	1.54	1.48	1.71	2,967.15	6.25	0.27	5.80
		0.5	1.61	1.28	1.26	1.62	1,121.28	6.13	0.26	0.02
		0.7	1.28	1.10	1.17	1.25	29.26	6.35	0.20	0.02
		0.9	1.21	1.06	1.14	1.32	24.21	5.73	0.23	0.00
25	37	0.1	5.54	2.19	2.53	5.11	3,600.02	13.46	0.38	38.24
		0.3	2.27	1.57	1.44	3.78	3,600.02	13.44	0.28	14.04
		0.5	1.57	1.28	1.23	3.51	2,130.29	13.69	0.26	1.66
		0.7	1.36	1.17	1.16	3.01	769.88	13.19	0.23	0.02
		0.9	1.27	1.08	1.18	2.63	51.21	12.00	0.22	0.06
30	45	0.1	5.40	2.29	2.36	8.21	3,600.07	24.96	0.33	51.88
		0.3	2.31	1.64	1.40	7.06	3,600.07	25.70	0.27	21.84
		0.5	1.61	1.33	1.21	7.21	3,600.04	22.83	0.32	6.86
		0.7	1.28	1.12	1.14	5.76	913.72	23.69	0.24	0.04
		0.9	1.21	1.08	1.12	5.46	104.78	21.86	0.25	0.06
35	52	0.1	5.36	2.44	2.20	14.03	3,600.09	41.76	0.34	66.46
		0.3	2.37	1.70	1.40	11.79	3,600.07	49.64	0.24	30.38
		0.5	1.63	1.35	1.20	11.56	3,600.09	51.41	0.22	10.68
		0.7	1.27	1.14	1.11	9.58	2,440.49	43.69	0.22	1.12
		0.9	1.18	1.07	1.11	8.79	155.11	48.17	0.18	0.04
40	60	0.1	5.61	2.55	2.20	25.19	3,600.11	83.89	0.30	76.18
		0.3	2.32	1.69	1.37	20.28	3,600.08	72.37	0.28	29.64
		0.5	1.66	1.40	1.19	20.28	3,600.10	81.40	0.25	14.58
		0.7	1.27	1.16	1.10	16.50	3,600.08	79.84	0.21	2.02
		0.9	1.27	1.10	1.16	14.59	633.71	85.24	0.17	0.08
45	67	0.1	5.66	2.61	2.17	37.07	3,600.11	143.38	0.26	81.48
		0.3	2.33	1.72	1.36	30.70	3,600.14	161.74	0.19	33.12
		0.5	1.61	1.37	1.18	31.47	3,600.08	151.43	0.21	13.50
		0.7	1.30	1.15	1.13	25.06	2,471.93	143.51	0.17	1.66
		0.9	1.18	1.09	1.09	24.24	1,058.17	159.14	0.15	0.08
50	75	0.1	5.64	2.78	2.03	55.84	3,600.13	128.40	0.43	95.14
		0.3	2.35	1.75	1.35	48.14	3,600.16	136.99	0.35	35.70
		0.5	1.66	1.42	1.17	48.38	3,600.14	139.41	0.35	16.92
		0.7	1.31	1.19	1.10	39.80	3,600.17	138.71	0.29	4.30
		0.9	1.22	1.10	1.11	35.29	2,254.26	172.78	0.20	0.10

the total runtime is just 30.78 at the similar configuration $(20, 40)$ in the instance set S1. This means a tight budget constraint raises the overall difficulty of the problem. This phenomenon is also demonstrated in the AC algorithm, albeit not so severely, where the runtime γ_{ac} decreases as β increases. Finally, compared to the BD algorithm, the AC algorithm is still efficient since, in most cases, the comparison ratio t_{ac}/t'_{bd} is a small fraction, especially when the instance size is large.

In conclusion, in both sets of instances, we demonstrate that the AC algorithm can produce a tight upper bound for the adjustability ratio within a short execution time. Moreover, the binary search used in the AC algorithm can be easily paralleled into a $(p+1)$-ary search where p is the number of computer cores. In this case, the complexity becomes $O(\Lambda_\gamma \log_p \frac{\bar{\gamma}-1}{\epsilon})$ with $\bar{\gamma}$ as some known upper bound of γ, scalar ϵ as a given tolerance, and Λ_γ as the complexity of solving the anchor cone formulation Λ with a fixed γ.

7 Conclusion and Future Extensions

We set out to answer two questions: can we identify the zero-adjustability conditions? And can we quantify the adjustability ratio?

For dynamic robust optimization problems, these questions are fundamental for understanding the conservativeness of RO solutions and the approximation performances of various policy families. From the game-theoretic perspective, these questions are posed in the same spirit that von Neumann studied minimax (in)equality in two-person zero-sum games. Such classical minimax setting does not allow strategy dependence. In this sense, our results extend the classical results, at least in a linear setting. The practical motivation of this setup comes from a plethora of interdiction and defender-attacker games.

Previous literature regarding adjustability provided intriguing but restrictive findings. The most general work to date had some key assumptions, for example, requiring the right hand side to be positive and the constraints to have only the “greater than or equal to” direction. These assumptions precluded the study of many problems, such as network optimization with arc capacities and resource allocation problems with budget constraints.

In this work, we dropped these assumptions and developed a general framework to analyze and quantify adjustability. This framework, including a unified theory and two algorithms, provided a set of tools for decision-makers to study the adjustability of the specific problem at hand. For instance, the theorem of alternatives (Theorem 2) gave a geometrically intuitive characterization for zero-adjustability; Theorem 4 introduced a constructive procedure for bounding adjustability ratio based on an "outer-approximation" idea; the anchor cone formulation (10) provided an efficient method to either derive analytical expressions or conduct numerical computations for adjustability ratio bounds.

For future work, it would be interesting to examine if Theorem 4 is, in general, the "optimal" way to bound adjustability ratio given an arbitrary problem setup. If the answer is affirmative, then one could examine more specialized settings to derive context-specific managerial and policy insights. In addition, the theoretical framework and new proof techniques may be of interest for researchers that work on areas mentioned in Section 1.1.

A Additional Proofs

Proposition 1. *The feasibility assumption implies (i) for every* $\xi \in \Xi$ *and* $u \in \mathbb{R}_+^k$*, if* $A^\intercal u = 0$ *then* $\langle C\xi, u\rangle \leq 0$*; (ii) for every* $\xi \in$ *cone*(Ξ)*,* $\langle c_i, \xi\rangle \leq 0$*. The boundedness assumption entails (i) there exists some* $u \in \mathbb{R}_+^k$ *such that* $A^\intercal u = a$*; (ii) for every* $\xi \in$ *cone*(Ξ)*,* $\langle c, \xi\rangle \leq 0$*.*

Proof. *For simplicity, we use* $\mathcal{Y}_\xi := \{y \in \mathcal{Y} \mid Ay \geq C\xi\}$ *to denote the solution space relative to a fixed* $\xi \in \Xi$*. The feasibility of* Π *means that for every* $\xi \in \Xi$*, there is a feasible* y*. This implies two things. First, the dual of the inner problem* $\min_{y\in\mathcal{Y}_\xi} f(\xi, y)$ *has a bounded objective value for every* $\xi \in \Xi$*. The rays of the dual problem is the set* $\{u \in \mathbb{R}_+^k \mid A^\intercal u = 0\}$*, which proves the first statement for feasibility. Second, suppose* $\langle c_i, \xi\rangle > 0$ *for some* $i \in [k]$ *and some* $\xi \in$ *cone*(Ξ)*, then, for every possible* y*, there always exists a sufficiently large* $\lambda > 0$ *such that* $\lambda\xi \in$ *cone*(Ξ) *violates the corresponding constraint in (1b). This contradicts the feasibility assumption. This proves the second statement regarding feasibility.*

The boundedness of Π *also implies two things. First, there exists some* $\xi \in \Xi$ *such that the optimal objective value of the inner problem is bounded. Then, the corresponding dual feasibility yields the first statement of boundedness. Second, suppose* $\langle c, \xi\rangle > 0$ *for some* $\xi \in$ *cone*(Ξ)*, then the term in the objective function (1a) is unbounded.* □

Corollary 2. Π *is zero-adjustable if and only if there exists a symmetric-optimal solution* $(\xi^\star, u^\star, u^\star\xi^{\star\intercal})$ *of the bidual* $\bar{\Delta}$ *such that* $\xi^\star \in$ *ext*(Ξ) *and* $u^\star \in$ *ext*$(\mathcal{U})$*.*

Proof. *Corollary 1 establishes the "if" part. We only need to show the "only if" part. Given* *a symmetric-optimal solution* $(\xi^\star, u^\star, u^\star \xi^{\star\intercal})$*, suppose* $\xi^\star \notin ext(\Xi)$ *or* $u^\star \notin ext(\mathcal{U})$*, we can construct a new solution* $(\xi', u', u'\xi'^\intercal)$ *in the desired form. With loss of generality, suppose* $\xi^\star \notin ext(\Xi)$*, we can fix the variables* u *at* $u^\star$ *in the symmetric bidual* $\bar{\Delta}^*$*, then the resulting problem is a linear program with decision variables* ξ*. Because the original problem* Π *is feasible and bounded, there is an optimal extreme point* ξ' *such that the new solution* $(\xi', u^\star, u^\star \xi'^\intercal)$ *preserves the objective value. Now, suppose* $u^\star \in ext(\mathcal{U})$*, we simply set* $u' = u^\star$*. Otherwise, we fix* ξ *at* ξ' *and use the same method to identify some* $u' \in ext(\mathcal{U})$ *that preserves the objective value. Then, the constructed solution is a symmetric-optimal solution of* $\bar{\Delta}$ *where both parts* u' *and* ξ' *are extreme points.* □

Proposition 2. *In bidual* $\bar{\Delta}$*, constraint set (3b) is equivalent to the following,*

$$v_i \in u_i conv(\Xi) + cone(\Xi), \ \forall i \in [k]. \tag{5}$$

Proof. *Constraint set (3b) can be written as* $Bv_i \leq u_i b$ *for all* i*. Every* $v_i \in u_i conv(\Xi) + cone(\Xi)$ *can be written as* $u_i \xi_i + \xi_i'$ *for some* $\xi_i \in conv(\Xi)$ *and* $\xi_i' \in cone(\Xi)$*.* $Bv_i \leq u_i b$ *is then satisfied, because* u_i *is nonnegative,* $B\xi_i \leq b$*, and* $B\xi_i' \leq 0$*. Conversely, when* $u_i = 0$*,* v_i *belongs to* $cone(\Xi) = 0 conv(\Xi) + cone(\Xi)$*; when* $u_i > 0$*,* $v_i / u_i \in \Xi$*, which is the same as* $v_i \in u_i conv(\Xi) + cone(\Xi)$*.* □

Proposition 3. *Any solution* $(\xi, u, V) \in \mathfrak{P}$ *is an extreme point if and only if* $\xi \in ext(\Xi)$*,* $u \in ext(\mathcal{U})$*, and for each row* $v_i = u_i \xi_i$ *of matrix* V*, either* $u_i = 0$ *or* $\xi_i \in ext(\Xi)$*.*

Proof. *For sufficiency, we take a solution* $(\xi, u, V) \in \mathfrak{P}$ *that satisfies the requirement. Towards a contradiction, we assume there exists two distinct feasible solutions* (ξ', u', V') *and* (ξ'', u'', V'') *such that* $0.5(\xi', u', V') + 0.5(\xi'', u', V'') = (\xi, u, V)$*. Focusing on the equalities that correspond to* u*, we have* $0.5u' + 0.5u'' = u$*. Because* u *is an extreme point by selection, we have* $u = u' = u''$*. Similarly, we also have* $\xi = \xi' = \xi''$*. Next, we focus on the equalities corresponding to each row of* V*. By Proposition 2, each row* v_i, v_i', v_i'' *can be represented as* $u_i \xi_i, u_i' \xi_i', u_i'' \xi_i''$ *with some* $\xi_i, \xi_i', \xi_i'' \in \Xi$*. Thus, for a fixed row index* i*, we have* $0.5 u_i' \xi_i' + 0.5 u_i'' \xi_i'' = u_i \xi_i$*. Suppose* $u_i = 0$*, then* $v_i = v_i' = v_i'' = 0$*. Otherwise, we can cancel* u_i *from the above equalities since* $u = u' = u''$ *by the previous argument, which gives* $0.5\xi_i' + 0.5\xi_i'' = \xi_i$*. This also implies* $v_i = v_i' = v_i''$ *since* ξ_i *is an extreme point by choice. In either case, we have* $(\xi', u', V') = (\xi'', u'', V'')$*, which leads to the desired contradiction.*

For necessity, we prove the contrapositive. Take any (ξ, u, V) that does not satisfy the requirement, then either $\xi \notin ext(\Xi)$, $u \notin ext(\mathcal{U})$, or at some row $v_i = u_i\xi_i$ of matrix V such that $u_i \neq 0$, we have $\xi_i \notin ext(\Xi)$. Every possible case implies that the associated vector x (x represents ξ, u, or some v_i) can be written as $x = 0.5x' + 0.5x''$ for some distinct, feasible x' and x''. Then, replacing the vector x in (ξ, u, V) with x' and x'' respectively gives two distinct feasible solutions in $\mathfrak{P}$ whose convex combination contains the solution (ξ, u, V). Thus, (ξ, u, V) cannot be an extreme point of $\mathfrak{P}$. □

Corollary 3. *$\Pi = (\Xi, \mathcal{U}, \bar{C})$ is zero-adjustable if it satisfies either of the following conditions: (i) $\{c\} \cup \{c_i\}_{i\in[k]} \subseteq N_\Xi(\xi)$ for some $\xi \in ext(\Xi)$; (ii) for every $u \in ext(\mathcal{U})$ with nonzero entries labeled by L_u, we have $\{c\} \cup \{c_i\}_{i\in L_u} \subseteq N_\Xi(\xi_u)$ for some $\xi_u \in ext(\Xi)$.*

Proof. *The first statement is a trivial consequence of Theorem 2. For the second, Proposition 3 implies there must exist some $u^\star \in \mathcal{U}^\star$ such that $u^\star$ is also an extreme point of $\mathcal{U}$. Then, by Theorem 2, the second statement also implies zero-adjustability.* □

Corollary 4. *Given $\Pi = (\Xi, \mathcal{U}, \bar{C})$ where $\bar{C} \geq 0$ and $\Xi \subseteq \mathbb{R}^n_+$ is convex, we have $\delta_{rel}(\Pi) \leq \mu(x)$ for every $x \in X^\star$.*

Proof. *Take any $x = (x_i)_{i\in[n]} \in X^\star$, let $\{x_j\}_{j\in J}$ contains all the unique vectors in x, thus $|J| = \mu(x)$. Then, the element $\xi = \sum_{j\in J} x_j/\mu(x) \in \Xi$ since Ξ is convex. Moreover, due to the choice of x and its nonnegativity, we have $\sum_{j\in J} x_j \geq (\max_{\xi\in\Xi} \xi_i)_{i\in[n]}$. Thus, the box set Ξ' uniquely determined by the maximum point $\xi' = \sum_{j\in J} x_j$ contains Ξ. Applying Theorem 4 with $K = \mu(x)$, we established the desired bound.* □

Corollary 5. *Given $\Pi = (\Xi, \mathcal{U}, \bar{C})$ where $\bar{C} \geq 0$ and $\Xi \subseteq \mathbb{R}^n_+$ is a convex lattice, suppose $\dim(\Xi) + 1 < n$, then for every $x \in X^\star$ such that $\mu(x) > \dim(\Xi) + 1$, there exists some $x' \in X^\star$ such that $\mu(x') < \mu(x)$.*

Proof. *For each $i \in [n]$, x_i is selected as a maximizer of $\max_{\xi\in\Xi} \xi_i$, which means x_i is located at the hyperplane $H_i := \{\xi_i = \max_{\xi\in\Xi} \xi_i\}$. Let $\{x_j\}_{j\in J}$ denotes the unique vectors in x, we define $\mathcal{H}_j := \{H_i \mid x_i = x_j\}$ to collect all the hyperplanes that select x_j as their maximizer. Then, there are two possible cases for the unique elements in x: (i) every unique vector x_j is not located on any hyperplane associated with other element $x_{j'}$,* i.e., *for all $j' \in J \setminus \{j\}$ and $H \in \mathcal{H}_{j'}$, $x_j \notin H$; (ii) $x_j \in H \in \mathcal{H}_{j'}$ for some $j' \neq j$. In the former case, the convex combination of $\{x_j\}_{j\in J}$, denoted by $Conv(x)$, forms a polytope of dimension $\mu(x) - 1$. To see* *this, there exists a rigid transformation that maps the maximum vector* $(\max_{\xi\in\Xi}\xi_i)_{i\in[n]}$ *to the origin and each hyperplane* H_i *to the trivial hyperplane* $\xi_i = 0$*. Under this transformation,* x_j*'s are located on nonnegative sections of distinct hyperplanes, thus they are affinely independent. This implies* $\dim(\Xi) \geq \mu(x) - 1$ *since Conv*$(x) \subseteq \Xi$*, which contradicts to the assumption* $\dim(\Xi) + 1 < \mu(x)$*. Thus, only the latter case is possible. In this case,* x_j *and* $x_{j'}$ *are the selected maximizer associated with hyperplanes in* $\mathcal{H}_i$ *and* $\mathcal{H}_j$*, respectively. Because* Ξ *is a lattice,* $x_j \vee x_{j'} \in \Xi$ *and is a maximizer that is located on all hyperplanes in* $\mathcal{H}_i \cup \mathcal{H}_j$*. Thus, replacing both* x_j *and* $x_{j'}$ *in* x *with* $x_j \vee x_{j'}$ *produces an element* $x' \in X^\star$ *such that* $\mu(x') \leq \mu(x) - 1$*. This completes the argument.* □

Proposition 4. *Every anchor cone* $\mathfrak{A}_{\mathcal{C},x_0}$ *has the following properties: (i)* $\mathfrak{A}_{\mathcal{C},x_0} = \{x_0\} + cone^\circ(\mathcal{C})$*; (ii)* $N_{\mathfrak{A}_{\mathcal{C},x_0}}(x_0) = cone(\mathcal{C})$*; (iii) constraints of* $\mathfrak{A}_{\mathcal{C},x_0}$ *that correspond to vectors in eray*$(\mathcal{C})$ *are sufficient to define* $\mathfrak{A}_{\mathcal{C},x_0}$*.*

Proof. *Both 1 and 2 can be verified directly from the definitions. For 3, every* $c_i \in \mathcal{C} \setminus eray(\mathcal{C})$ *can be written as a conic combination of the extreme rays in eray*$(\mathcal{C})$*. Then, combining the constraints* $\{\langle c_j, x\rangle \leq \langle c_j, x_0\rangle\}_{c_j\in eray(\mathcal{C})}$ *with the same coefficients produces the constraint associated with* c_i*.* □

Proposition 5. *Given* Ξ *is bounded, Formulation (10) is feasible if either* $\Xi \cap int(cone^*(\bar{C})) \neq \varnothing$ *or* $\Xi \subseteq int(cone^\circ(\bar{C}))$*.*

Proof. *For the first case, pick any* $\xi \in \Xi \cap int(cone^*(\bar{C})) \neq \varnothing$*. By the choice of* ξ*, we have* $\langle c_j, \gamma\xi\rangle > 0 = \langle c_j, 0\rangle$ *for all* $j \in J$ *and* $\gamma > 0$*. This implies* $0 \in int(\mathfrak{A}_{\bar{C},\gamma\xi})$*, which further implies* $\lim_{\gamma\to\infty}\mathfrak{A}_{\bar{C},\gamma\xi} = \mathbb{R}^n$*. Thus, for any bounded* Ξ*, there exists a sufficiently large scalar* γ_1 *so that* $\mathfrak{A}_{\bar{C},\gamma_1\xi} \supseteq \Xi$*, which implies that* γ_1 *is a feasible solution to Formulation (10). For the second case, notice that for any* ξ*,* $\lim_{\lambda\to 0}\mathfrak{A}_{\bar{C},\lambda\xi} = cone^\circ(\bar{C})$*. Thus, if* $\Xi \subseteq int(cone^\circ(\bar{C}))$ *and* Ξ *is bounded, there always exists some* $\lambda_2 > 0$ *so that* $\mathfrak{A}_{\bar{C},\lambda_2\xi} \supseteq \Xi$*.* □

Proposition 6. *Formulation (10) is infeasible if* $\dim(\Xi) + \dim(-cone(\bar{C}) \cap cone(\bar{C})) > n$*.*

Proof. *We use* $\hat{\mathcal{C}}$ *to denote the set* $-cone(\bar{C}) \cap cone(\bar{C})$*. First, notice that* $\hat{\mathcal{C}}$ *is a subspace. Moreover, the dimension of the polar cone* $\dim(cone^\circ(\bar{C})) = n - \dim\hat{\mathcal{C}}$*, which can be proved using induction. Finally, the first property in Proposition 4 implies* $\dim(\mathfrak{A}_{\bar{C},x_0}) = \dim(cone^\circ(\bar{C}))$*, which along with the inequality in the claim imply that* $\dim(\Xi) > \dim(\mathfrak{A}_{\bar{C},x_0})$*. Therefore, the anchor cone* $\mathfrak{A}_{\bar{C},x_0}$ *can never enclose the target space* Ξ*.* □

References

Lu An, Aranya Chakrabortty, and Alexandra Duel-Hallen. A stackelberg security investment game for voltage stability of power systems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3359–3364. IEEE, 2020.

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

Amir Ardestani-Jaafari and Erick Delage. Linearized robust counterparts of two-stage robust optimization problems with applications in operations management. *INFORMS Journal on Computing*, 33(3):1138–1161, 2021.

Pranjal Awasthi, Vineet Goyal, and Brian Y Lu. On the adaptivity gap in two-stage robust linear optimization under uncertain packing constraints. *Mathematical Programming*, 173(1):313–352, 2019.

Dimitra Bampou and Daniel Kuhn. Scenario-free stochastic programming with polynomial decision rules. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7806–7812. IEEE, 2011.

Chaithanya Bandi, Eojin Han, and Omid Nohadani. Sustainable inventory with robust periodic-affine policies and application to medical supply chains. *Management Science*, 65(10):4636–4655, 2019.

Chaithanya Bandi, Nikolaos Trichakis, and Phebe Vayanos. Robust multiclass queuing theory for wait time estimation in resource allocation systems. *Management Science*, 65(1):152–187, 2019.

Aharon Ben-Tal and Arkadi Nemirovski. Robust optimization–methodology and applications. *Mathematical programming*, 92(3):453–480, 2002.

Aharon Ben-Tal, Alexander Goryashko, Elana Guslitzer, and Arkadi Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical programming*, 99(2):351–376, 2004.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

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

Dimitris Bertsimas and Vineet Goyal. On the power and limitations of affine policies in two-stage adaptive optimization. *Mathematical programming*, 134(2):491–531, 2012.

Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Mathematical programming*, 98(1):49–71, 2003.

Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

Dimitris Bertsimas and Aurélie Thiele. A robust optimization approach to inventory theory. *Operations research*, 54(1):150–168, 2006.

Dimitris J Bertsimas, Patrick Jaillet, and Amedeo R Odoni. A priori optimization. *Operations Research*, 38(6):1019–1033, 1990.

Dimitris Bertsimas, Dan A Iancu, and Pablo A Parrilo. Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research*, 35(2):363–394, 2010.

Dimitris Bertsimas, Vineet Goyal, and Xu Andy 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.

Dimitris Bertsimas, Vineet Goyal, and Brian Y Lu. A tight characterization of the performance of static solutions in two-stage adjustable robust linear optimization. *Mathematical Programming*, 150(2):281–319, 2015.

Dimitris Bertsimas, Vishal Gupta, and Nathan Kallus. Data-driven robust optimization. *Mathematical Programming*, 167(2):235–292, 2018.

Xin Chen and Yuhan Zhang. Uncertain linear programs: Extended affinely adjustable robust counterparts. *Operations Research*, 57(6):1469–1482, 2009.

Xin Chen, Melvyn Sim, Peng Sun, and Jiawei Zhang. A linear decision-based approximation approach to stochastic programming. *Operations Research*, 56(2):344–357, 2008.

Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

Omar El Housni and Vineet Goyal. On the optimality of affine policies for budgeted uncertainty sets. *Mathematics of Operations Research*, 46(2):674–711, 2021.

S Rasoul Etesami and Tamer Başar. Dynamic games in cyber-physical security: An overview. *Dynamic Games and Applications*, 9(4):884–913, 2019.

Laurent El Ghaoui, Maksim Oks, and Francois Oustry. Worst-case value-at-risk and robust portfolio optimization: A conic programming approach. *Operations research*, 51(4):543–556, 2003.

Ali Haddad-Sisakht and Sarah M Ryan. Conditions under which adjustability lowers the cost of a robust linear program. *Annals of Operations Research*, 269(1):185–204, 2018.

Grani A Hanasusanto, Daniel Kuhn, and Wolfram Wiesemann. K-adaptability in two-stage robust binary programming. *Operations Research*, 63(4):877–891, 2015.

Dan A. Iancu, Mayank Sharma, and Maxim Sviridenko. Supermodularity and affine policies in dynamic robust optimization. *Operations Research*, 61(4):941–956, 2013.

Dan Andrei Iancu, Nikolaos Trichakis, and Do Young Yoon. Monitoring with limited information. *Management Science*, 67(7):4233–4251, 2021.

Eitan Israeli and R Kevin Wood. Shortest-path network interdiction. *Networks: An International Journal*, 40(2):97–111, 2002.

Renata A Konrad, Andrew C Trapp, Timothy M Palmbach, and Jeffrey S Blom. Overcoming human trafficking via operations research and analytics: Opportunities for methods, models, and applications. *European Journal of Operational Research*, 259(2):733–745, 2017.

Dmytro Korzhyk, Zhengyu Yin, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe. Stackelberg vs. nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research*, 41:297–327, 2011.

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

Ahmadreza Marandi and Dick Den Hertog. When are static and adjustable robust optimization problems with constraint-wise uncertainty equivalent? *Mathematical programming*, 170(2):555–568, 2018.

Timothy C Matisziw and Alan T Murray. Modeling s–t path availability to support disaster vulnerability assessment of network infrastructure. *Computers & Operations Research*, 36(1):16–26, 2009.

Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Mehran Poursoltani and Erick Delage. Adjustable robust optimization reformulations of two-stage worst-case regret minimization problems. *Operations Research*, 2021.

R. Tyrrell Rockafellar. *Convex analysis.* Princeton Mathematical Series. Princeton University Press, Princeton, N. J., 1970.

Chao Shang, Xiaolin Huang, and Fengqi You. Data-driven robust optimization based on kernel learning. *Computers & Chemical Engineering*, 106:464–479, 2017.

Hanif D Sherali and Amine Alameddine. A new reformulation-linearization technique for bilinear programming problems. *Journal of Global optimization*, 2(4):379–410, 1992.

Ruifeng Shi, Shaopeng Li, Penghui Zhang, and Kwang Y Lee. Integration of renewable energy sources and electric vehicles in v2g network with adjustable robust optimization. *Renewable Energy*, 153:1067–1080, 2020.

David Simchi-Levi, Nikolaos Trichakis, and Peter Yun Zhang. Designing response supply chain against bioattacks. *Operations Research*, 67(5):1246–1268, 2019.

Ioannis Tsaknakis, Mingyi Hong, and Shuzhong Zhang. Minimax problems with coupled linear constraints: Computational complexity, duality and solution methods. *arXiv preprint arXiv:2110.11210*, 2021.

Ningji Wei, Jose L Walteros, McKenzie R Worden, and Héctor J Ortiz-Peña. A resiliency analysis of information distribution policies over mobile ad hoc networks. *Optimization Letters*, 15(4):1081–1103, 2021.

Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

Jianzhe Zhen, Dick Den Hertog, and Melvyn Sim. Adjustable robust optimization via fourier–motzkin elimination. *Operations Research*, 66(4):1086–1100, 2018.

İhsan Yanıkoğlu, Bram L. Gorissen, and Dick den Hertog. A survey of adjustable robust optimization. *European Journal of Operational Research*, 277(3):799–813, 2019.