An exact method for a class of robust nonlinear optimization problems

Dimitris Bertsimas
Operations Research Center, Massachusetts Institute of Technology, United States, d.bertsim@mit.edu

Danique de Moor
Amsterdam Business School, University of Amsterdam, The Netherlands, d.demoor@uva.nl

Dick den Hertog
Amsterdam Business School, University of Amsterdam, The Netherlands, d.denhertog@uva.nl

Thodoris Koukouvinos
Operations Research Center, Massachusetts Institute of Technology, United States, tkoukouv@mit.edu

Jianzhe Zhen
ETH Zürich, Zürich, Switzerland, trevorzhen@gmail.com

We introduce a novel exact approach for addressing a broad spectrum of optimization problems with robust nonlinear constraints. These constraints are defined as sums of products of linear times concave (SLC) functions with respect to the uncertain parameters. Our approach synergizes a cutting set method with reformulation-perspectification techniques and branch and bound. We further extend the applicability of our approach to robust convex optimization, which can be reformulated as a problem involving a sum of linear times linear functions in the uncertain parameters, thus broadening the scope of existing literature. Numerical experiments on a robust convex geometric optimization problem and a robust linear optimization problem with data uncertainty and implementation error show that our approach can solve robust nonlinear problems that cannot be solved by existing methods in the literature. Moreover, a numerical experiment on a lot-sizing problem on a network demonstrates the efficiency of our method for two-stage ARO problems.

Key words: Reformulation-perspectification techniques, perspective functions, robust nonlinear optimization, adaptive robust optimization.

1. Introduction

In the last two decades, robust optimization (RO) has become a popular approach for solving optimization problems under uncertainty. Though introduced by Soyster (1973), the work of Ben-Tal and Nemirovski (1999), El Ghaoui and Lebret (1997), and El Ghaoui et al. (1998) together with improvements in efficient algorithms for conic and semidefinite optimization sparked significant interest in the field of RO in the late nineties. In contrast to stochastic programming, which requires the underlying probability distribution of uncertain parameters to be known or estimated, RO does not necessitate any information about this distribution. Instead, it employs an uncertainty set — a collection of scenarios where the solution is safeguarded. In RO, constraints are expected to hold for all realizations of the uncertain parameters within this uncertainty set, e.g.,

$$f(\boldsymbol{x}, \boldsymbol{z}) \leq 0, \quad \forall \boldsymbol{z} \in \mathcal{Z}, \tag{1}$$

where the vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ denote the decision variable and the uncertain parameter, respectively, and $\mathcal{Z}$ is the uncertainty set. For several types of constraint functions and uncertainty sets, this semi-infinite constraint (1) can be reformulated into a finite set of convex constraints, also referred to as the robust counterpart (RC). The first step in computing the RC is to rewrite (1) as:

$$\sup_{\boldsymbol{z} \in \mathcal{Z}} f(\boldsymbol{x}, \boldsymbol{z}) \leq 0. \tag{2}$$

In cases where the set $\mathcal{Z}$ is compact and convex, and the function f is convex in $\boldsymbol{x}$ and concave in $\boldsymbol{z}$, a computationally tractable Robust Constraint (RC) of (2) may be derived, as finding the worst-case scenario is equivalent to maximizing a concave function over $\mathcal{Z}$. The authors in Ben-Tal et al. (2015) proposed to calculate the support function of the uncertainty set and the concave conjugate of the nonlinear constraint function. This leads to a reformulation of (2) into a finite set of convex constraints.

If the function f is nonlinear and convex in both $\boldsymbol{x}$ and $\boldsymbol{z}$, finding an equivalent tractable reformulation is, in general, challenging since finding the worst-case scenario is equivalent to maximizing a convex function. In Bertsimas et al. (2023c), the authors propose a Reformulation-Perspectification Technique (RPT) based approach to solve such robust convex constraints with convex uncertainty sets. First, they reformulate constraint (1) as an uncertain linear constraint with bilinear uncertainty and nonconvex uncertainty set. Next, using partial or full RPT, they derive several convex outer approximations of the uncertainty set, thereby obtaining a robust linear constraint with a convex uncertainty set. Hence, well-established RO techniques (see Bertsimas and den Hertog (2022)) can be applied to finally obtain a computationally tractable safe approximation of (1). In this paper, we propose a method for computing the global optimal solution of robust convex optimization problems, thereby extending the existing literature.

When the function f is neither convex nor concave in the uncertain parameter $\boldsymbol{z}$, only for special cases can a computationally tractable safe approximation be found in the literature. Ben-Tal et al. (2015) show that if $f(\boldsymbol{x}, \boldsymbol{z})$ can be written as $\boldsymbol{g}(\boldsymbol{x})^\top \boldsymbol{h}(\boldsymbol{z})$ in which $\boldsymbol{h}(\boldsymbol{z})$ is nonconcave or $g(\boldsymbol{x})$ attains both positive and negative values, then by parametrization of $\boldsymbol{\zeta} = \boldsymbol{h}(\boldsymbol{z})$, we may replace the uncertainty set $\bar{\mathcal{Z}} = \{(\boldsymbol{z}, \boldsymbol{\zeta}) \mid \boldsymbol{z} \in \mathcal{Z}, \boldsymbol{\zeta} = \boldsymbol{h}(\boldsymbol{z})\}$ by its convex hull. Hence, in cases where the convex hull can be computed, a computationally tractable RC can be derived. One such example is a robust nonconcave quadratic constraint with an ellipsoidal uncertainty set. For this case, Ben-Tal and

Nemirovski (1998) derive an exact computationally tractable RC by showing that the convex hull of the parametrized uncertainty set can be written as a Linear Matrix Inequality (LMI). Finally, in the case of a conic quadratic inequality and structured norm-bounded uncertainty sets as well as uncertainty sets described by finitely many ellipsoids, there exists a computationally tractable safe approximation of the RC, see (Ben-Tal et al., 2009, Chapter 7). In this paper, we develop a method for computing the global optimal solution of problems where f is neither convex nor concave in the uncertain parameter $\boldsymbol{z}$ but sum of linear times concave (SLC).

In this paper, we propose a novel approach based on cutting sets as well as the Reformulation-Perspectification Technique with Branch and Bound (RPT-BB), recently introduced by Bertsimas et al. (2023a). Our approach is able to obtain the global optimal solution of optimization problems containing robust constraints that are neither convex nor concave but SLC with respect to the uncertain parameters $\boldsymbol{z}$. Namely, we assume that f can be written as the sum of functions that consist of the product of linear times a concave function with respect to the uncertain parameter $\boldsymbol{z}$.

In RO, the decisions are assumed to be here-and-now variables, i.e., they have to be determined before the uncertainty is realized. Adaptive robust optimization (ARO), introduced in Ben-Tal et al. (2004), relaxes this assumption by also considering wait-and-see variables which can be determined after more information on the uncertain parameters is known. A general fixed recourse two-stage adaptive RO constraint is given by

$$\boldsymbol{a}(\boldsymbol{z})^\top \boldsymbol{x} + \boldsymbol{s}^\top \boldsymbol{y}(\boldsymbol{z}) - b(\boldsymbol{z}) \leq 0, \quad \forall \boldsymbol{z} \in \mathcal{Z}, \tag{3}$$

where $\boldsymbol{x}$ is the here-and-now decision variable, $\boldsymbol{y}(\boldsymbol{z})$ is the second-stage wait-and-see variable, and $\boldsymbol{z}$ is the uncertain parameter taking values in an uncertainty set $\mathcal{Z}$. In general, fixed recourse two-stage RO problems are NP-hard (Ben-Tal et al., 2004).

Three approaches have been studied in the literature to make optimization problems containing constraints in the form of (3) computationally tractable. The first approach is to approximate the problem by restricting the wait-and-see variables, also called decision rules, to a certain class of functions. Ben-Tal et al. (2004) introduce affine decision rules (ADRs), restricting the wait-and-see variables to be affine functions of the uncertain parameters, and Xu and Burer (2018); Hanasusanto and Kuhn (2018) extend it to quadratic decision rules. Observe that imposing a quadratic decision rule to (3) leads to a static robust optimization constraint with quadratic uncertainty, in which case the resulting problem can be solved to global optimality with our proposed approach, in contrast to existing approaches that are only able to obtain an approximation (Xu and Burer, 2018; Hanasusanto and Kuhn, 2018). The second and third approach seek to find an exact solution using Bender's decomposition method (Thiele et al., 2010; Bertsimas et al., 2013) and a column-and-constraint generation procedure (Zeng and Zhao, 2013), respectively. Zeng and Zhao (2013) show that problems containing two-stage RO constraints can be reformulated into a static RO problem with bilinear uncertainty. Hence, our approach can solve such problems to global optimality.

Contributions

Our main contributions can be summarized as follows:

1. We extend the existing literature by proposing an exact method for solving a broad class of robust optimization problems including robust neither convex nor concave constraints, that are SLC in the uncertain parameters. We prove that, under mild assumptions about the constraints in the optimization problem, our method converges to the global optimum.
2. We apply our framework to robust convex optimization, which is a sub-class of problems involving SLC functions in the uncertain parameters, thereby extending the work of Bertsimas et al. (2023c) to obtain the *exact* optimal solution of robust convex optimization problems instead of an *approximation*. Moreover, we apply our approach to solve exactly two-stage adaptive optimization problems with general convex uncertainty set, by reformulating two-stage RO constraints into static RO constraints with bilinear uncertainty.
3. We demonstrate the effectiveness of our approach by extensive numerical experiments on problems that cannot be solved exactly by existing methods. More specifically, in terms of RO problems, we consider a robust geometric optimization problem and a linear program with data uncertainty and implementation error, and in terms of ARO problems, we consider a lot-sizing problem on a network.

The rest of the paper is structured as follows: In Section 2, we illustrate the class of problems that we address, in Section 3 we demonstrate the main building blocks of our method, in Section 4, we present numerical results and finally we summarize our key findings in Section 5.

Notation. The calligraphic letters $\mathcal{I}$, $\mathcal{J}$, $\mathcal{K}$, $\mathcal{L}$ and the corresponding capital Roman letters I, J, K, L are reserved for finite index sets and their respective cardinalities, i.e, $\mathcal{I}=\{1,\ldots,I\}$ etc. Moreover, we also use $[N]$, $N\in\mathbb{N}$, to denote the set of running indices $\{1,\ldots,N\}$. Let $\mathbb{R}^{m\times n}$ denote the set of real $m\times n$ matrices. We generally use bold faced characters such as $\boldsymbol{a}\in\mathbb{R}^n$ and $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ to represent vectors and matrices. We denote the vector of zeros by $\mathbf{0}$, the matrix of zeros by $\boldsymbol{O}$, and the vector of ones with $\boldsymbol{e}$. The *domain* of a function $f:\mathbb{R}^{n_\nu}\to[-\infty,+\infty]$ is defined as $\mathrm{dom}(f)=\{\boldsymbol{\nu}\in\mathbb{R}^{n_\nu}\mid f(\boldsymbol{\nu})<+\infty\}$. The function f is *proper* if $f(\boldsymbol{\nu})>-\infty$ for all $\boldsymbol{\nu}\in\mathbb{R}^{n_\nu}$ and $f(\boldsymbol{\nu})<+\infty$ for at least one $\boldsymbol{\nu}\in\mathbb{R}^{n_\nu}$, implying that $\mathrm{dom}(f)\neq\emptyset$. In addition, f is *closed* if f is lower semicontinuous and either $f(\boldsymbol{\nu})>-\infty$ for all $\boldsymbol{\nu}\in\mathbb{R}^{n_\nu}$ or $f(\boldsymbol{\nu})=-\infty$ for all $\boldsymbol{\nu}\in\mathbb{R}^{n_\nu}$. The *convex conjugate* of a function $f:\mathbb{R}^{n_\nu}\to[-\infty,+\infty]$ is the function $f^*:\mathbb{R}^{n_\nu}\to[-\infty,+\infty]$ defined through $f^*(\boldsymbol{w})=\sup_{\boldsymbol{\nu}}\left\{\boldsymbol{\nu}^\top\boldsymbol{w}-f(\boldsymbol{\nu})\right\}$. The convex conjugate $(f^*)^*$ of f^* is called the *biconjugate* of f and is abbreviated as f^{**}. The *concave conjugate* of a function $f:\mathbb{R}^{n_\nu}\to[-\infty,+\infty]$ is the function $f_*:\mathbb{R}^{n_\nu}\to[-\infty,+\infty]$ defined through $f_*(\boldsymbol{w})=\inf_{\boldsymbol{\nu}}\left\{\boldsymbol{\nu}^\top\boldsymbol{w}-f(\boldsymbol{\nu})\right\}$. The *indicator* function $\delta(\boldsymbol{\nu}|\mathcal{S})=0$ for all $\boldsymbol{\nu}\in\mathcal{S}$, and ∞ otherwise. Its convex conjugate form is known as the *support* function of the set $\mathcal{S}$, which is denoted as $\delta^*(\boldsymbol{y}|\mathcal{S})$ and by definition, we have $\delta^*(\boldsymbol{y}|\boldsymbol{S})=\sup_{\boldsymbol{\nu}\in\mathcal{S}}\{\boldsymbol{y}^\top\boldsymbol{\nu}\}$. The *perspective* $h:\mathbb{R}^{n_\nu}\times\mathbb{R}_+\to[-\infty,+\infty]$ of a proper, closed and convex function $f:\mathbb{R}^{n_\nu}\to(-\infty,+\infty)$ is defined for all $\boldsymbol{\nu}\in\mathbb{R}^{n_\nu}$ and $t\in\mathbb{R}_+$ as $h(\boldsymbol{\nu},t)=tf(\boldsymbol{\nu}/t)$ if $t>0$, and $h(\boldsymbol{\nu},0)=\delta^*_{\mathrm{dom}(f^*)}(\boldsymbol{\nu})$, where $\delta^*_{\mathrm{dom}(f^*)}$ denotes the *recession* function. For ease of exposition, we use $tf(\boldsymbol{\nu}/t)$ to denote the perspective function $h(\boldsymbol{\nu},t)$ for the rest of this paper.

2. Generic problem formulation

We consider the following generic robust nonlinear optimization problem

$$\begin{array}{ll}\min\limits_{\boldsymbol{x}} & f_0(\boldsymbol{x})\\ \text{s.t.} & f_k(\boldsymbol{x},\boldsymbol{z})\le 0,\quad \forall \boldsymbol{z}\in\mathcal{Z}_k,\ \forall k\in\mathcal{K},\\ & \boldsymbol{x}\in\mathcal{X},\end{array}\tag{4}$$

where the objective function $f_0:\mathbb{R}^{n_x}\to(-\infty,+\infty]$ is proper, closed and convex, and each function $f_k:\mathbb{R}^{n_x+n_z}\to(-\infty,+\infty]$ is given by

$$f_k(\boldsymbol{x},\boldsymbol{z})=f_{0k}(\boldsymbol{x},\boldsymbol{z})+\sum_{i\in\mathcal{I}_k}(q_{ik}-\boldsymbol{p}_{ik}^\top\boldsymbol{z})f_{ik}(\boldsymbol{x},\boldsymbol{z}),\quad \forall k\in\mathcal{K}.\tag{5}$$

Here, $\boldsymbol{p}_{ik}\in\mathbb{R}^{n_z}, q_{ik}\in\mathbb{R}$, and we assume that $f_{ik}(\boldsymbol{x},\boldsymbol{z})$ is proper, closed and convex in $\boldsymbol{x}$, and concave in $\boldsymbol{z}$ for every $i\in\{0\}\cup\mathcal{I}_k,\ k\in\mathcal{K}$. The uncertainty set $\mathcal{Z}_k$ is nonempty, compact and convex, which is given by

$$\mathcal{Z}_k=\left\{\boldsymbol{z}\in\mathbb{R}^{n_z}\mid \boldsymbol{D}_k^\top\boldsymbol{z}\le\boldsymbol{d}_k,\ g_{jk}(\boldsymbol{z})\le 0,\ \forall j\in\mathcal{J}_k\right\},\quad \forall k\in\mathcal{K},$$

where $\boldsymbol{D}_k\in\mathbb{R}^{n_z\times n_d}$, $\boldsymbol{d}_k\in\mathbb{R}^{n_d}$, and the function $g_{jk}:\mathbb{R}^{n_z}\to(-\infty,+\infty]$ is proper, closed, nonlinear, and convex for every $j\in\mathcal{J}_k$ and $k\in\mathcal{K}$. Further, the set $\mathcal{X}$ is compact and convex, given by

$$\mathcal{X}=\left\{\boldsymbol{x}\in\mathbb{R}^{n_x}\mid \boldsymbol{R}^\top\boldsymbol{x}\le\boldsymbol{r},\ h_j(\boldsymbol{x})\le 0,\ \forall j\in\mathcal{J}_x\right\},$$

where $\boldsymbol{R}\in\mathbb{R}^{n_x\times n_p}, \boldsymbol{r}\in\mathbb{R}^{n_p}$, and the nonlinear function $h_j:\mathbb{R}^{n_x}\to(-\infty,+\infty]$ is proper, closed and convex for all $j\in\mathcal{J}_x$. Moreover, we assume that (5) satisfies the following assumption:

Assumption 1. *If $f_{ik}(\boldsymbol{x},\boldsymbol{z})$ is nonlinear in $\boldsymbol{z}$ for some $i\in\mathcal{I}_k$ and $k\in\mathcal{K}$, then $q_{ik}-\boldsymbol{p}_{ik}^\top\boldsymbol{z}\ge 0$.*

Observe that under Assumption 1 each function $f_k(\boldsymbol{x},\boldsymbol{z})$ is convex in $\boldsymbol{x}$ as a nonnegative linear combination of convex functions is convex in $\boldsymbol{x}$. Therefore, for any fixed $\boldsymbol{z}$ that satisfies Assumption 1, Problem (4) is a convex optimization problem with a finite number of constraints. We next present examples of robust constraints that fit in problem format (4).

Example 1 (Robust convex constraint). Consider the robust convex constraint

$$f(\boldsymbol{x},\boldsymbol{z})=h(\boldsymbol{T}(\boldsymbol{x})\boldsymbol{z}+\boldsymbol{t}(\boldsymbol{x}))\leq 0,\quad \forall \boldsymbol{z}\in\mathcal{Z},$$

where $h:\mathbb{R}^m\to(-\infty,+\infty]$ is a proper, closed, and convex function, and $\boldsymbol{T}:\mathbb{R}^{n_x}\to\mathbb{R}^{m\times n_z}$, $\boldsymbol{t}:\mathbb{R}^{n_x}\to\mathbb{R}^{n_z}$ are affine. Bertsimas et al. (2023c) show that this robust convex constraint can be equivalently written as the following SLC function:

$$\mathrm{Tr}\left(\boldsymbol{T}(\boldsymbol{x})^\top\boldsymbol{w}\boldsymbol{z}^\top\right)+\boldsymbol{t}(\boldsymbol{x})^\top\boldsymbol{w}-w_0\leq 0,\quad \forall(\boldsymbol{w},\boldsymbol{z})\in\Lambda,$$

where $\Lambda=\{(w_0,\boldsymbol{w},\boldsymbol{z})\mid \boldsymbol{z}\in\mathcal{Z}_k,\ \ \boldsymbol{w}\in\mathrm{dom}(h_k^*),\ \ h_k^*(\boldsymbol{w})\leq w_0\}$. Observe that a robust geometric constraint, that is,

$$\log\left(\exp\left((\boldsymbol{B}_1\boldsymbol{z}-\boldsymbol{e})^\top\boldsymbol{x}\right)+\exp\left((\boldsymbol{B}_2\boldsymbol{z}-\boldsymbol{e})^\top\boldsymbol{x}\right)\right)\leq 0,\quad \forall\boldsymbol{z}\in\mathcal{Z},$$

where the matrices $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are known, fits into this class of problems. This case is treated in the numerical experiment in Section 4.1. □

Example 2 (Robust quadratic nonconcave constraint). The quadratic constraint function

$$f(\boldsymbol{x},\boldsymbol{z})=\boldsymbol{z}^\top\boldsymbol{T}(\boldsymbol{x})\boldsymbol{z}+\boldsymbol{t}(\boldsymbol{x})^\top\boldsymbol{z}+\tau(\boldsymbol{x}),$$

where $\boldsymbol{T}:\mathbb{R}^{n_x}\to\mathbb{R}^{n_z\times n_z}$, $\boldsymbol{t}:\mathbb{R}^{n_x}\to\mathbb{R}^{n_z}$, $\tau:\mathbb{R}^{n_x}\to\mathbb{R}$ are all affine in $\boldsymbol{x}$, and $\boldsymbol{T}(\boldsymbol{x})$ not necessarily positive semidefinite, can be written as a sum of linear times linear functions. Note that a linear constraint with data uncertainty ($\boldsymbol{z}$) and implementation error ($\boldsymbol{v}$), that is,

$$(\boldsymbol{a}+\boldsymbol{z})^\top(\boldsymbol{x}+\boldsymbol{v})\leq b,\quad \forall(\boldsymbol{z},\boldsymbol{v})\in\mathcal{Z},$$

fits into this class of problems. This case is treated in the numerical experiment in Section 4.2. □

Example 3 (Robust nonconcave constraints). Another example of a robust nonconcave constraint is

$$f(\boldsymbol{x},\boldsymbol{z})=\sum_{i\in\mathcal{I}}(q_i-\boldsymbol{p}_i^\top\boldsymbol{z})\log\left(1+\boldsymbol{x}^\top\boldsymbol{a}_i(\boldsymbol{z})\right),$$

where $\boldsymbol{x}\in\mathbb{R}_+^{n_x},\boldsymbol{z}\in\mathbb{R}^{n_z}$ and $\boldsymbol{a}_i(\boldsymbol{z})\in\mathbb{R}_+^{n_x}$ is affine in $\boldsymbol{z}$. □

We next provide examples of adaptive optimization problems that fit in problem format (5).

Example 4 (Robust adaptive constraint with quadratic decision rules). Consider the following two-stage adaptive constraint with fixed recourse

$$f(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \boldsymbol{a}(\boldsymbol{z})^\top \boldsymbol{x} + \boldsymbol{s}^\top \boldsymbol{y}(\boldsymbol{z}) \le 0, \quad \forall \boldsymbol{z} \in \mathcal{Z}_k,$$

where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is a here-and-now variable, $\boldsymbol{y} \in \mathbb{R}^{n_y}$ is a wait-and-see variable and $\boldsymbol{a}(\boldsymbol{z}) \in \mathbb{R}^{n_x}$ is affine in $\boldsymbol{z} \in \mathbb{R}^{n_z}$. By imposing the quadratic decision rule $y_i^Q(\boldsymbol{z}) = \boldsymbol{z}^\top \boldsymbol{P}_i \boldsymbol{z} + \boldsymbol{\rho}_i^\top \boldsymbol{z} + \nu_i$, we obtain the constraint

$$\boldsymbol{a}(\boldsymbol{z})^\top \boldsymbol{x} + \sum_{i=1}^{n_y} s_i \left(\boldsymbol{z}^\top \boldsymbol{P}_i \boldsymbol{z} + \boldsymbol{\rho}_i^\top \boldsymbol{z} + \nu_i\right) \le 0, \quad \forall \boldsymbol{z} \in \mathcal{Z},$$

where $\boldsymbol{P}_i \in \mathbb{R}^{n_z \times n_z}$, $\boldsymbol{\rho}_i \in \mathbb{R}^{n_z}$, $i \in [n_y]$, and $\boldsymbol{\nu} \in \mathbb{R}^{n_y}$. Observe that in this case, the constraint function is the sum of linear times linear functions in $\boldsymbol{z}$ and therefore fits in problem format (5). □

Example 5 (Adaptive two-stage robust optimization problem). Consider the following adaptive two-stage robust optimization problem

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}\in\mathcal{X}} \max\limits_{\boldsymbol{z}\in\mathcal{Z}} \min\limits_{\boldsymbol{y}(\cdot)\ge 0} & \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{\zeta}^\top \boldsymbol{y}(\boldsymbol{z}) \\ \text{s.t.} & \boldsymbol{A}(\boldsymbol{z})\boldsymbol{x} + \boldsymbol{S}\boldsymbol{y}(\boldsymbol{z}) \le \boldsymbol{b}(\boldsymbol{z}), \quad \forall \boldsymbol{z} \in \mathcal{Z}, \end{array} \tag{6}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ denotes the here-and-now variable, $\boldsymbol{y}(\boldsymbol{z}) \in \mathbb{R}^{n_y}$ denotes the wait-and-see variable and $\boldsymbol{b}(\boldsymbol{z}) \in \mathbb{R}^L, \boldsymbol{A}(\boldsymbol{z}) \in \mathbb{R}^{L\times n_y}$ are both affine in $\boldsymbol{z} \in \mathbb{R}^{n_z}$. Further, we have $\boldsymbol{c} \in \mathbb{R}^{n_x}, \boldsymbol{\zeta} \in \mathbb{R}^{n_y}$, and $\boldsymbol{S} \in \mathbb{R}^{L\times n_y}$. Dualizing (6) over $\boldsymbol{y}(\cdot)$ we obtain

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}\in\mathcal{X}} & \boldsymbol{c}^\top \boldsymbol{x} + \tau \\ \text{s.t.} & \boldsymbol{\mu}^\top \boldsymbol{A}(\boldsymbol{z})\boldsymbol{x} - \boldsymbol{\mu}^\top \boldsymbol{b}(\boldsymbol{z}) \le \tau, \quad \forall (\boldsymbol{\mu}, \boldsymbol{z}) \in \mathcal{U}, \end{array} \tag{7}$$

where $\mathcal{U} = \{\boldsymbol{z} \in \mathbb{R}^{n_z}, \boldsymbol{\mu} \in \mathbb{R}^L_+ \mid \boldsymbol{z} \in \mathcal{Z},\ \boldsymbol{S}\boldsymbol{\mu}^\top + \boldsymbol{\zeta} \ge \boldsymbol{0}\}$. Observe that in this case, the constraint function is a disjoint bilinear constraint and therefore fits in problem format 4. We note that with the additional assumption that Problem (6) has relatively complete resource, i.e., for all $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{z} \in \mathcal{Z}$ the inner linear optimization problem is feasible (see Xu and Burer (2018)), the primal problem is feasible and thanks to strong duality the dual problem is bounded. □

3. Our approach

In this section, we describe our approach to obtaining the global optimal solution of Problem (4). Our framework combines a cutting set method, as described in Mutapcic and Boyd (2009), with RPT-BB. It comprises the following steps: First, we formulate the master problem as an instance of Problem (4), imposing each robust constraint for a finite subset of scenarios $\mathcal{S}_k \subseteq \mathcal{Z}_k$, that is,

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & f_0(\boldsymbol{x}) \\ \text{s.t.} & f_k(\boldsymbol{x},\boldsymbol{z}) \le 0, \quad \forall \boldsymbol{z} \in \mathcal{S}_k,\ \forall k \in \mathcal{K}, \\ & \boldsymbol{x} \in \mathcal{X}. \end{array} \tag{8}$$

We solve Problem (8) and obtain a solution $\boldsymbol{x}^*$. Then, we solve each k-th subproblem, that is,

$$\max_{\boldsymbol{z}\in\mathcal{Z}_k} \quad f_k(\boldsymbol{x}^*, \boldsymbol{z}), \tag{9}$$

utilizing RPT-BB, and obtain solution $\boldsymbol{z}_k^*$. If the optimal objective value of the k-th subproblem is positive, i.e., $f_k(\boldsymbol{x}^*, \boldsymbol{z}_k^*) > 0$, we add $\boldsymbol{z}_k^*$ to the finite subset of scenarios $\mathcal{S}_k$. If the optimal objective value of every subproblem is nonpositive, we conclude that $\boldsymbol{x}^*$ is the optimal solution of Problem (4) and terminate the method, else we repeat the previous steps. Our approach is presented in Algorithm 1. In the remainder of this section, we discuss important aspects of Algorithm 1 including how we solve every subproblem using RPT-BB in Section 3.1, the initialization in Section 3.2, and the convergence in Section 3.3.

Algorithm 1 Cutting set method for solving Problem (4).

Input: $\mathcal{S}_1^0, \ldots, \mathcal{S}_K^0$: Initial scenarios for each constraint.

Output: $\boldsymbol{x}^*$: Optimal solution of Problem (4).

1: Initialize $\mathcal{S}_1, \ldots, \mathcal{S}_K = \mathcal{S}_1^0, \ldots, \mathcal{S}_K^0$.
2: Solve the master problem (8) with input $\mathcal{S}_1, \ldots, \mathcal{S}_K$ and obtain optimal solution $\boldsymbol{x}^*$.
3: **for** $k \in \mathcal{K}$ **do**
4:     Solve the k-th subproblem (9) using RPT-BB with input $\boldsymbol{x}^*$ and obtain optimal solution $\boldsymbol{z}_k^*$.
5:     **if** $f_k(\boldsymbol{x}^*, \boldsymbol{z}_k^*) > 0$ **then**
6:         $\mathcal{S}_k = \mathcal{S}_k \cup \{\boldsymbol{z}_k^*\}$.
7:     **end if**
8: **end for**
9: **if** $f_k(\boldsymbol{x}^*, \boldsymbol{z}_k^*) \le 0,\ \forall k \in \mathcal{K}$ **then**
10:     Return the optimal solution $\boldsymbol{x}^*$.
11: **else**
12:     Repeat Steps 2-8.
13: **end if**

3.1. RPT-BB for solving subproblems

We use the technique proposed in Bertsimas et al. (2023a) to perspectify (5) and obtain the following equivalent reformulation

$$f_k(\boldsymbol{x}, \boldsymbol{z}) = f_{0k}(\boldsymbol{x}, \boldsymbol{z}) + \sum_{i\in\mathcal{I}_k} (q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}) f_{ik}\left(\boldsymbol{x}, \frac{q_{ik}\boldsymbol{z} - \boldsymbol{z}\boldsymbol{z}^\top \boldsymbol{p}_{ik}}{q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}}\right). \tag{10}$$

Linearizing the product terms $\boldsymbol{z}\boldsymbol{z}^\top$ by a matrix $\boldsymbol{U} \in \mathcal{S}^{n_x}$, we obtain the following objective for the k-th subproblem (9)

$$\max_{(\boldsymbol{z},\boldsymbol{U})\in\mathcal{U}_k} \quad f_{0k}(\boldsymbol{x},\boldsymbol{z}) + \sum_{i\in\mathcal{I}_k}(q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}) f_{ik}\left(\boldsymbol{x}, \frac{q_{ik}\boldsymbol{z} - \boldsymbol{U}\boldsymbol{p}_{ik}}{q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}}\right), \tag{11}$$

where $\mathcal{U}_k = \left\{\boldsymbol{z} \in \mathbb{R}^{n_z} \mid \boldsymbol{z} \in \mathcal{Z}_k,\ \boldsymbol{U} = \boldsymbol{z}\boldsymbol{z}^\top\right\}$. Observe that the set $\mathcal{U}_k$ is not convex, since it contains the quadratic equality constraint $\boldsymbol{U} = \boldsymbol{z}\boldsymbol{z}^\top$. We can obtain a tractable convex outer approximation Θ_k of $\mathcal{U}_k$ in the following way: We enlarge the set $\mathcal{Z}_k$ by first generating additional constraints from pairwise multiplications of the original constraints, and then, convexify all additional constraints by reformulating them in their perspective form and subsequently substitute all product terms $\boldsymbol{z}\boldsymbol{z}^\top$ by the newly introduced matrix $\boldsymbol{U} \in \mathbb{S}^{n_z}$. We refer to Bertsimas et al. (2023a) for the pairwise multiplication of a linear with a convex inequality and to Bertsimas et al. (2023b) for the pairwise multiplication of two cone inequalities. We finally obtain the following safe approximation for the k-th subproblem (9), which is jointly concave in the uncertain parameters $(\boldsymbol{z}, \boldsymbol{U})$:

$$\sup_{(\boldsymbol{z},\boldsymbol{U})\in\Theta_k} f_{0k}(\boldsymbol{x}^*,\boldsymbol{z}) + \sum_{i\in\mathcal{I}_k}(q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}) f_{ik}\left(\boldsymbol{x}^*, \frac{q_{ik}\boldsymbol{z} - \boldsymbol{U}\boldsymbol{p}_{ik}}{q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}}\right) \le 0. \tag{12}$$

Observe that the embedded optimization problem in (12) gives us an upper bound for the optimal objective value of Problem (9). Further, a lower bound can be obtained with local optimization algorithms, e.g., the mountain climbing procedure Tao and An (1997). RPT-BB combines these ways of obtaining good bounds with a branching mechanism leveraging the RPT solution, thus allowing for fast computation of the global optimal solution. The branch and bound scheme involves the generation of hyperplanes that bisect the feasible region. Subsequently, the root node is divided into two child nodes. At each child node, we solve Problem (9) using RPT in which the uncertainty set $\mathcal{Z}_k$ is intersected with one of the closed half spaces defined by the generated hyperplane. Unless, we have reached the optimal solution, the procedure is iteratively applied by selecting the node with the highest upper bound and generating a new hyperplane. We refer to Bertsimas et al. (2023a) for more details about the method.

3.2. Initializing Algorithm 1

An important part of Algorithm 1 is how to obtain the initial finite subset of scenarios $\mathcal{S}_1^0, \ldots, \mathcal{S}_K^0$. An obvious approach is to take random feasible scenarios for each $k \in \mathcal{K}$. In this section, we propose an intuitive initialization scheme leveraging RPT, which consists of the following steps: First, we apply RPT to each robust constraint and obtain in total K safe approximations that are concave in the uncertain parameters with a convex uncertainty set, see Section 3.1. Then, we compute the RC of the approximation by solving the embedded optimization problem in (12) for each $k \in \mathcal{K}$ using Theorem 1, for which we need the following assumption.

ASSUMPTION 2. *For any $k \in \mathcal{K}$, we assume that*

$$ri(dom f_{0k})(\boldsymbol{x},\cdot) \cap_{i\in\mathcal{I}} ri(dom f_{ik}(\boldsymbol{x},\cdot)) \neq \emptyset, \qquad \forall \boldsymbol{x} \in \mathcal{X}.$$

For ease of exposition, we define $\tilde{f}_{ik}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{U}) = (q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}) f_{ik}\left(\boldsymbol{x}, \frac{q_{ik}\boldsymbol{z} - \boldsymbol{U}\boldsymbol{p}_{ik}}{q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}}\right)$.

THEOREM 1. *If Assumption 2 holds, then $\boldsymbol{x} \in \mathcal{X}$ satisfies (12) if and only if $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{v}, w_{ik} \in \mathbb{R}^{n_z}$, $\boldsymbol{V}, \boldsymbol{W}_{ik} \in \mathbb{R}^{n_z \times n_z}$, $i \in \mathcal{I}_0$, satisfy the following system of convex constraints:*

$$\begin{cases} \delta^*(\boldsymbol{v},\boldsymbol{V} \mid \Theta_k) - (f_{0k})_*(\boldsymbol{x},\boldsymbol{w}_{0k}) + \sum_{i\in\mathcal{I}} \left(q_{ik}^2 \dfrac{(\boldsymbol{W}_{ik})_{11}}{(\boldsymbol{p}_{ik})_1^2} - q_{ik}\dfrac{(\boldsymbol{w}_{ik})_1}{(\boldsymbol{p}_{ik})_1} \right) \le 0, \\ \sum_{i\in\mathcal{I}\cup\{0\}} \boldsymbol{w}_{ik} = \boldsymbol{v}, \\ \sum_{i\in\mathcal{I}} \boldsymbol{W}_{ik} = \boldsymbol{V}, \\ \dfrac{(\boldsymbol{W}_{ik})_1}{(\boldsymbol{p}_k)_1} = \dfrac{(\boldsymbol{W}_{ik})_j}{(\boldsymbol{p}_k)_j}, & \forall i \in \mathcal{I}, \forall j \in [n_z]\setminus\{1\}, \\ q_{ik}\dfrac{(\boldsymbol{W}_{ik})_{11}}{(\boldsymbol{p}_{ik})_1^2} - \dfrac{(\boldsymbol{w}_{ik})_1}{(\boldsymbol{p}_{ik})_1} = q_{ik}\dfrac{(\boldsymbol{W}_{ik})_{1j}}{(\boldsymbol{p}_{ik})_1(\boldsymbol{p}_{ik})_j} - \dfrac{(\boldsymbol{w}_{ik})_j}{(\boldsymbol{p}_{ik})_j}, & \forall i \in \mathcal{I}, \forall j \in [n_z]\setminus\{1\}, \\ (f_{ik})_*\left(\boldsymbol{x}, \dfrac{(\boldsymbol{W}_{ik})_1}{(\boldsymbol{p}_{ik})_1}\right) + q_{ik}\dfrac{(\boldsymbol{W}_{ik})_{11}}{(\boldsymbol{p}_{ik})_1^2} - \dfrac{(\boldsymbol{w}_{ik})_1}{(\boldsymbol{p}_{ik})_1} \le 0, & \forall i \in \mathcal{I}, \end{cases}$$

where $(\boldsymbol{W}_{ik})_j$ denotes the j-th column of $\boldsymbol{W}_{ik}$, $(\boldsymbol{w}_{ik})_j$, and $(\boldsymbol{p}_{ik})_j$ denotes the j-th entry of $\boldsymbol{w}_{ik}$ and $\boldsymbol{p}_{ik}$, respectively, and in the first and last constraint the concave conjugate operation only applies to the second argument.

Proof. We have

$$\sup_{(\boldsymbol{z},\boldsymbol{U})\in\Theta_k} \left\{ f_{0k}(\boldsymbol{x},\boldsymbol{z}) + \sum_{i\in\mathcal{I}_k} \tilde{f}_{ik}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{U}) \right\} \le 0$$

$$\iff \sup_{\boldsymbol{z},\boldsymbol{U}} \left\{ \boldsymbol{0}^\top \boldsymbol{z} + \boldsymbol{O}^\top \boldsymbol{U} + f_{0k}(\boldsymbol{x},\boldsymbol{z}) + \sum_{i\in\mathcal{I}_k} \tilde{f}_{ik}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{U}) - \delta(\boldsymbol{z},\boldsymbol{U} \mid \Theta_k) \right\} \le 0 \tag{13a}$$

$$\iff \inf_{\substack{\boldsymbol{v},\boldsymbol{w}_{ik}\\ \boldsymbol{V},\boldsymbol{W}_{ik}}} \left\{ \begin{array}{c} \delta^*(\boldsymbol{v},\boldsymbol{V} \mid \Theta_k) - (f_{0k})_*(\boldsymbol{x},\boldsymbol{w}_{0k}) \\ -\sum_{i\in\mathcal{I}} (\tilde{f}_{ik})_*(\boldsymbol{x},\boldsymbol{w}_{ik},\boldsymbol{W}_{ik}) \end{array} \middle| \begin{array}{l} \sum_{i\in\mathcal{I}_0} \boldsymbol{w}_{ik} = \boldsymbol{v}, \\ \sum_{i\in\mathcal{I}_0} \boldsymbol{W}_{ik} = \boldsymbol{V} \end{array} \right\} \le 0 \tag{13b}$$

$$\iff \begin{cases} \delta^*(\boldsymbol{v},\boldsymbol{V} \mid \Theta_k) \le (f_{0k})_*(\boldsymbol{x},\boldsymbol{w}_{0k}) + \sum_{i\in\mathcal{I}} (\tilde{f}_{ik})_*(\boldsymbol{x},\boldsymbol{w}_{ik},\boldsymbol{W}_{ik}), \\ \sum_{i\in\mathcal{I}_0} \boldsymbol{w}_{ik} = \boldsymbol{v}, \\ \sum_{i\in\mathcal{I}_0} \boldsymbol{W}_{ik} = \boldsymbol{V}, \end{cases} \tag{13c}$$

where (13a) follows from the definition of the support function. Meanwhile, (13b) follows from (Rockafellar, 1970, Theorem 16.4), and its infimum is attained. This applies due to Assumption 2. Consequently, (13c) holds because the infimum of (13b) is attained. Filling in the expression for $(\tilde{f}_{ik})_*$ for all $i \in \mathcal{I}$, which can be obtained from Lemma 1, the proof follows.

We refer to (Bertsimas and den Hertog, 2022, Chapter 2) for an overview of how to compute the support function for several commonly used uncertainty sets.

Next, we solve Problem (4), with each robust constraint replaced by its tractable safe approximation, and obtain a solution $\boldsymbol{x}^*$. Then, for fixed $\boldsymbol{x}^*$, we solve the approximation of each k-th subproblem and obtain solutions $(\bar{\boldsymbol{z}}^k, \bar{\boldsymbol{U}}^k)$. Finally, for each $k \in \mathcal{K}$, we create the finite subset of scenarios $\bar{\mathcal{S}}_k^0$ given by

$$\bar{\mathcal{S}}_k^0 = \left\{ \bar{\boldsymbol{z}}^k, \frac{\bar{\boldsymbol{u}}_1^k}{\bar{z}_1^k}, \ldots, \frac{\bar{\boldsymbol{u}}_{n_z}^k}{\bar{z}_{n_z}^k} \right\}.$$

Bertsimas et al. (2023a) shows that only if $\boldsymbol{z} \in \mathbb{R}_+^{n_z}$, then $\bar{\mathcal{S}}_k^0 \subseteq \Theta_k$ is guaranteed. Hence we take the finite subset of feasible scenarios to be

$$\mathcal{S}_k^0 = \left\{ \boldsymbol{z}^k \in \bar{\mathcal{S}}_k^0 \mid \boldsymbol{z}^k \in \Theta_k \right\}.$$

We summarize our proposed initialization scheme in Algorithm 2.

Algorithm 2 Initialization scheme for Algorithm 1

Input: Problem (4)

Output: $\mathcal{S}_1^0, \ldots, \mathcal{S}_K^0$: Initial scenarios for each constraint of Problem (4)

for $k \in \mathcal{K}$ **do**
    Apply RPT to (9) and obtain the safe approximation (12)
    Apply Theorem 1 to obtain the RC of the embedded optimization problem in (12)
end for
Solve the reformulated problem and obtain a solution $\boldsymbol{x}^*$
for $k \in \mathcal{K}$ **do**
    For fixed $\boldsymbol{x}^*$ solve the embedded optimization problem in (12) and obtain $\overline{\boldsymbol{z}}^k, \overline{\boldsymbol{U}}^k$
    Take $\mathcal{S}_k^0$ to be the finite subset of feasible scenarios
end for
Return $\mathcal{S}_1^0, \ldots, \mathcal{S}_K^0$

If the support function $\delta^*(\boldsymbol{v}, \boldsymbol{V} \mid \Theta_k)$, $k \in \mathcal{K}$, cannot be easily computed, we can use an alternative approach based on Zhen et al. (2023) to compute the RC of the problem that arises when all constraints in (4) are approximated by (12), i.e.,

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & f_0(\boldsymbol{x}) \\ \text{s.t.} & \tilde{f}_k(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{U}_k) \leq 0, \quad \forall (\boldsymbol{z}_k, \boldsymbol{U}_k) \in \boldsymbol{\Theta}_k, \quad \forall k \in \mathcal{K}, \\ & \boldsymbol{x} \in \mathcal{X}, \end{array} \tag{14}$$

where $\tilde{f}_k(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{U}_k) = f_0(\boldsymbol{x}, \boldsymbol{z}_k) + \sum_{i\in\mathcal{I}} \tilde{f}_{ik}(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{U}_k)$. Recall that $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \boldsymbol{R}^\top \boldsymbol{x} \le \boldsymbol{r}, h_j(\boldsymbol{x}) \le 0,\ \forall j \in \mathcal{J}_x\}$.

THEOREM 2. *If the feasible region of* (14) *is nonempty and* $\mathcal{Z}_k$, $\forall k \in \mathcal{K}$, *admits a strict Slater point, then* (14) *has the same optimal value as*

$$
\begin{aligned}
\sup_{\boldsymbol{w}^k, \boldsymbol{y}, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{v}_k, \boldsymbol{V}_k} \quad & -f_0^*(\boldsymbol{w}^0) - \sum_{k\in\mathcal{K}} y_k \tilde{f}_k^* \left(\frac{\boldsymbol{w}^k}{y_k}, \frac{\boldsymbol{v}_k}{y_k}, \frac{\boldsymbol{V}_k}{y_k} \right) - \sum_{j\in\mathcal{J}_x} \eta_j h_j^* \left(\frac{\boldsymbol{w}^{K+j}}{\eta_j} \right) - \boldsymbol{\mu}^\top \boldsymbol{r} \\
\text{s.t.} \quad & \sum_{k\in\overline{\mathcal{K}}} \boldsymbol{w}^k + \boldsymbol{\mu}^\top \boldsymbol{R} = \boldsymbol{0}, \\
& y_k \delta \left(\left(\frac{\boldsymbol{v}_k}{y_k}, \frac{\boldsymbol{V}_k}{y_k} \right) \Big| \ \Theta_k \right) \le 0, \quad k \in \mathcal{K}, \\
& \boldsymbol{y} \ge \boldsymbol{0},\ \boldsymbol{\eta} \ge \boldsymbol{0},\ \boldsymbol{\mu} \ge \boldsymbol{0},
\end{aligned} \tag{15}
$$

where $\overline{\mathcal{K}} = \{0, 1, \ldots, K + J_x\}$, $\boldsymbol{w}^k \in \mathbb{R}^{n_x}$, $k \in \overline{\mathcal{K}}$, $\boldsymbol{y} \in \mathbb{R}_+^K$, $\boldsymbol{\eta} \in \mathbb{R}^{J_x}$, $\boldsymbol{\mu} \in \mathbb{R}^{n_p}$, $\boldsymbol{v}_k \in \mathbb{R}^{n_z}$, $\boldsymbol{V}_k \in \mathbb{R}^{n_z \times n_z}$, $k \in \mathcal{K}$, *and the convex conjugate operation is only applied to the first argument in* $\tilde{f}_k$ *for all* $k \in \mathcal{K}$.

Proof. Zhen et al. (2023, p. 4) show that the dual problem of (14) is given by

$$
\begin{aligned}
\sup_{\substack{\boldsymbol{w}^k, \boldsymbol{y}, \boldsymbol{\eta}, \\ \boldsymbol{\mu}, \boldsymbol{z}_k, \boldsymbol{U}_k}} \quad & -f_0^*(\boldsymbol{w}^0) - \sum_{k\in\mathcal{K}} y_k \tilde{f}_k^* \left(\frac{\boldsymbol{w}^k}{y_k}, \boldsymbol{z}_k, \boldsymbol{U}_k \right) - \sum_{j\in\mathcal{J}_x} \eta_j h_j^* \left(\frac{\boldsymbol{w}^{K+j}}{\eta_j} \right) - \boldsymbol{\mu}^\top \boldsymbol{r} \\
\text{s.t.} \quad & \sum_{k\in\overline{\mathcal{K}}} \boldsymbol{w}^k + \boldsymbol{\mu}^\top \boldsymbol{R} = \boldsymbol{0}, \\
& (\boldsymbol{z}_k, \boldsymbol{U}_k) \in \Theta_k, \quad \forall k \in \mathcal{K}, \\
& \boldsymbol{y} \ge \boldsymbol{0}, \boldsymbol{\eta} \ge \boldsymbol{0}, \boldsymbol{\mu} \ge \boldsymbol{0},
\end{aligned} \tag{16}
$$

where $\boldsymbol{w}^k \in \mathbb{R}^{n_x}, k \in \overline{\mathcal{K}}$, and the suprema of (14) and (16) coincide because the feasible region of (14) is nonempty and bounded by assumption (Zhen et al., 2023, Theorem 2(ii)). Furthermore, we know that the suprema of (16) and (15) coincide if (16) admits a strict Slater point (Zhen et al., 2023, Proposition 2(ii)).

For the remainder of the proof, we show that (16) indeed admits a strict Slater point. First, observe that $\mathcal{Z}_k$ admits a strict Slater point $\boldsymbol{z}_k^S$ by assumption. It can be verified that $(\boldsymbol{z}_k^S, \boldsymbol{z}_k^S(\boldsymbol{z}_k^S)^\top)$ is a strict Slater point of Θ_k for every $k \in \mathcal{K}$. Finally, it follows from (Zhen et al., 2023, Proposition C.6) that there exists a $((\boldsymbol{w}^k)^S, \boldsymbol{y}^S, \boldsymbol{\eta}^S, \boldsymbol{\mu}^S)$ such that $((\boldsymbol{w}^k)^S, \boldsymbol{y}^S, \boldsymbol{\eta}^S, \boldsymbol{u}^S, \boldsymbol{z}_k^S, \boldsymbol{z}_k^S(\boldsymbol{z}_k^S)^\top)$ constitutes a strict Slater point of (16).

Using Theorem 2, if the feasible region of (14) is nonempty, we can obtain the primal optimal solution $\boldsymbol{x}^*$ from the KKT conditions, which we can substitute in (12) to obtain an initial finite subset of scenarios as done in Steps 6 to 9 in Algorithm 2. We summarize the new steps of our initialization scheme in Algorithm 3. We note that Step 6 of Algorithm 3 is easily performed by most solvers. Observe that, as this initialization scheme is based on the dual, Algorithm 3 cannot be applied in the case we have integer variables. Hence, in that case, 2 has to be used as initialization scheme.

Algorithm 3 Initialization scheme for Algorithm 1 based on the dual

Input: Problem (4)

Output: $\mathcal{S}_1^0,\ldots,\mathcal{S}_K^0$: Initial scenarios for each constraint of Problem (4)

1: **for** $k\in\mathcal{K}$ **do**
2: Apply RPT to (9) and obtain the safe approximation (12)
3: **end for**
4: Solve the convex optimistic dual (14), and obtain a solution $(\boldsymbol{y}^*,(\boldsymbol{w}^k)^*,\boldsymbol{v}_k^*,\ \boldsymbol{V}_k^*)$
5: For fixed $(\boldsymbol{y}^*,(\boldsymbol{w}^k)^*,\boldsymbol{v}_k^*,\ \boldsymbol{V}_k^*)$, solve the KKT conditions and obtain $\boldsymbol{x}^*$
6: Do Steps 6-9 of Algorithm 2
7: Return $\mathcal{S}_1^0,\ldots,\mathcal{S}_K^0$

3.3. Convergence analysis of Algorithm 1

In this section, we prove the convergence of Algorithm 1 in case some additional assumptions are satisfied. We make the following assumptions:

Assumption 3. *The function $f_{ik}(\boldsymbol{x},\boldsymbol{z})$ is Lipschitz continuous in $\boldsymbol{x}$ for fixed $\boldsymbol{z}$, for every $i\in\mathcal{I}_0$, $k\in\mathcal{K}$.*

We note that the sets $\mathcal{X}$ and $\mathcal{Z}_k$, $k\in\mathcal{K}$ are bounded, since they are assumed to be compact.

Theorem 3. *If Assumption 3 holds, Algorithm 1 converges to the global optimum of Problem (4).*

Proof. The necessary conditions in order to have finite convergence are the following, see Mutapcic and Boyd (2009), Kelley Jr (1960):

1. Each subproblem can be solved exactly.
2. The constraint functions $f_k(\boldsymbol{x},\boldsymbol{z})$ are uniformly Lipschitz continuous over $\boldsymbol{x}$, i.e., there exists a constant L such that for every $k\in\mathcal{K}$ and every $\boldsymbol{z}\in\mathcal{Z}_k$ it satisfies

$$|f_k(\boldsymbol{x}_1,\boldsymbol{z})-f_k(\boldsymbol{x}_2,\boldsymbol{z})|\le L\|\boldsymbol{x}_1-\boldsymbol{x}_2\|.$$

Observe that 1. is satisfied since RPT-BB is an exact method for computing the global optimal solution (Bertsimas et al., 2023a, Theorem 2). We next show that 2. is also satisfied for Problem (4). First observe that since each set $\mathcal{Z}_k$ is bounded, there exists a constant M_k such that $\|\boldsymbol{z}\|\le M_k,\ \forall \boldsymbol{z}\in\mathcal{Z}_k$. Thus we have

$$\boldsymbol{p}_{ik}^\top\boldsymbol{z}\ge-\|\boldsymbol{p}_{ik}\|\|\boldsymbol{z}\|\implies-\boldsymbol{p}_{ik}^\top\boldsymbol{z}\le\|\boldsymbol{p}_{ik}\|\|\boldsymbol{z}\|\le\|\boldsymbol{p}_{ik}\|M_k.$$

Let L_{ik} denote the Lipschitz constant of the function f_{ik}. For any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}$, we have that $|f_k(\boldsymbol{x}_1, \boldsymbol{z}) - f_k(\boldsymbol{x}_2, \boldsymbol{z})| =$

$$\begin{aligned}
&\left| f_{0k}(\boldsymbol{x}_1, \boldsymbol{z}) - f_{0k}(\boldsymbol{x}_2, \boldsymbol{z}) + \sum_{i \in \mathcal{I}_k} (q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z})(f_{ik}(\boldsymbol{x}_1, \boldsymbol{z}) - f_{ik}(\boldsymbol{x}_2, \boldsymbol{z})) \right| \\
&\le |f_{0k}(\boldsymbol{x}_1, \boldsymbol{z}) - f_{0k}(\boldsymbol{x}_2, \boldsymbol{z})| + \left| \sum_{i \in \mathcal{I}_k} (q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z})(f_{ik}(\boldsymbol{x}_1, \boldsymbol{z}) - f_{ik}(\boldsymbol{x}_2, \boldsymbol{z})) \right| \\
&\le L_{0k} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\| + \sum_{i \in \mathcal{I}_k} (q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}) \, |f_{ik}(\boldsymbol{x}_1, \boldsymbol{z}) - f_{ik}(\boldsymbol{x}_2, \boldsymbol{z})| \\
&\le L_{0k} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\| + \sum_{i \in \mathcal{I}_k} (q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}) L_{ik} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \\
&\le L_{0k} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\| + \sum_{i \in \mathcal{I}_k} (q_{ik} + \|\boldsymbol{p}_{ik}\| M_k) L_{ik} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \\
&= \tilde{L}_k \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|,
\end{aligned}$$

where $\tilde{L}_k = L_{0k} + \sum_{i \in \mathcal{I}_k} (q_{ik} + \|\boldsymbol{p}_{ik}\| M_k) L_{ik}$. Let $L = \max \left\{ \tilde{L}_1, \ldots, \tilde{L}_K \right\}$. We then obtain

$$|f_k(\boldsymbol{x}_1, \boldsymbol{z}) - f_k(\boldsymbol{x}_2, \boldsymbol{z})| \le \tilde{L}_k \|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \le L \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|.$$

We note that the examples discussed in Section 2 satisfy the uniform Lipschitz continuity assumption.

4. Numerical experiments

In this section, we first demonstrate the effectiveness of our approach on two robust nonlinear problems that cannot be solved exactly by existing methods. To be more specific, we demonstrate the effectiveness on a robust convex geometric optimization problem and a linear optimization problem with data uncertainty and implementation error. Next, we demonstrate the efficiency of our approach on the two-stage ARO lot-sizing problem on a network. The first numerical experiment is performed on an Intel Core i7-8665U 1.90GHz Windows computer with 32.0GB of RAM. The computations are implemented using YALMIP Löfberg (2004) in MATLAB (R2023a). The second and third numerical experiment are performed on an Intel i9 2.3GHz CPU core with 16.0GB of RAM. The computations are implemented in Julia 1.5.3 and the Julia package `JuMP.jl` version 0.21.6. All computations are conducted with MOSEK 9.3.18 (MOSEK ApS, 2024).

4.1. Robust convex geometric optimization

We consider the robust geometric optimization problem from Bertsimas et al. (2023c) which contains robust geometric constraints. The problem is formulated as follows:

$$\begin{aligned}
\min_{\boldsymbol{x}} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\
\text{s.t.} \quad & \log \left(\exp \left((\boldsymbol{B}_i^{(1)} \boldsymbol{z} - \boldsymbol{e})^\top \boldsymbol{x} \right) + \exp \left((\boldsymbol{B}_i^{(2)} \boldsymbol{z} - \boldsymbol{e})^\top \boldsymbol{x} \right) \right) \le 0, \quad \forall \boldsymbol{z} \in \mathcal{Z}, i \in \mathcal{I}.
\end{aligned} \tag{17}$$

We utilize the uncertainty set $\mathcal{Z}$ introduced in Bertsimas et al. (2023c), which is the intersection of a hypercube and a norm ball uncertainty set, that is,

$$\mathcal{Z} = \{\|\boldsymbol{z}\|_\infty \leq 1,\ \|\boldsymbol{z}\|_2 \leq \gamma\},$$

where $\gamma = \sqrt[L]{\frac{2^L\Gamma(L/2+1)}{\pi^{L/2}}}$ and Γ denotes the gamma function, ensuring the volume of the hypercube coincides with the volume of the norm ball. When applying the cutting plane method, the master-problem is formulated as follows:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} & \log\left(\exp\left((\boldsymbol{B}_i^{(1)}\boldsymbol{z} - \boldsymbol{e})^\top \boldsymbol{x}\right) + \exp\left((\boldsymbol{B}_i^{(2)}\boldsymbol{z} - \boldsymbol{e})^\top \boldsymbol{x}\right)\right) \leq 0,\ \forall \boldsymbol{z}_i \in \mathcal{S}_i, i \in \mathcal{I}, \end{array} \tag{18}$$

where the set $\mathcal{S}_i$ contains a finite subset of scenarios for all $i \in \mathcal{I}$. The i-th subproblem, after applying the biconjugate trick, is formulated as follows:

$$\begin{array}{ll} \max\limits_{\boldsymbol{z},\boldsymbol{w}} & (\boldsymbol{z}^\top \boldsymbol{B}_i^{(1)\top}\boldsymbol{x} - \boldsymbol{e}^\top\boldsymbol{x})w_1 + (\boldsymbol{z}^\top \boldsymbol{B}_i^{(2)\top}\boldsymbol{x} - \boldsymbol{e}^\top\boldsymbol{x})w_2 - w_0 \\ \text{s.t.} & \boldsymbol{w} \in \mathcal{W}, \\ & \boldsymbol{z} \in \mathcal{Z}. \end{array} \tag{19}$$

Let $\mathcal{W}^1 = \{\boldsymbol{w} \in \mathbb{R}^3 : w_1, w_2 \geq 0,\ w_1 + w_2 = 1\}$. We have

$$\mathcal{W} = \left\{\boldsymbol{w} \in \mathcal{W}^1, \boldsymbol{t} \in \mathbb{R}^2,\ \ t_1 + t_2 \leq w_0,\ \ w_i \exp\left(\frac{-t_i}{w_i}\right) \leq 1,\ \ i \in \{1,2\}\right\}.$$

To ensure convergence of Algorithm 1, we impose the additional constraint $\|\boldsymbol{x}\|_2 \leq 10{,}000$. For the RC of (17) and the RPT reformulation of (19) we refer to Appendix B.1. Moreover, Assumption 3 is satisfied, as is demonstrated in Appendix C.1.

In Table 1 we illustrate the results of our approach for Problem (17) where we initialize Algorithm 1 from the nominal feasible solution. Using Algorithm 2 resulted in a longer computational time, as in this case the master problem takes longer to solve. We observe that the computational time increases when the dimension of the uncertain parameters increases, as well as when the number of robust constraints increases, as is also illustrated in Figure 2.

In Figure 1 we illustrate how the number of violated constraints changes in each iteration of Algorithm 1. We observe a large reduction in the number of violated constraints in the first 3 iterations and a slower reduction afterwards.

Finally, in Figure 3 we show how the number of cutting planes generated in Algorithm 1 varies with the dimension of uncertain parameters as well as the number of robust constraints. We observe that the number of generated cutting planes increases when the dimension of the uncertain parameters increases, while it does not change significantly when the number of robust constraints increases.

n_z	k	n_x	Nom Obj	Approach with nominal scenario		
				Opt	Time(s)	Cut Pl
10	20	100	-2954.91	-2290.52	10.50	10.60
10	50	100	-2807.53	-2101.26	19.05	10.50
20	50	100	-2792.30	-1607.22	32.96	12.90
30	50	100	-2954.91	-1079.30	53.72	23.00
10	40	1000	-8952.08	-8375.16	37.39	10.90
20	40	1000	-8995.21	-7949.64	43.75	11.20

Table 1 Comparison of the nominal problem with Algorithm 1 initialized from the nominal feasible solution (approach with nominal scenario), where the results reflect the average of 10 randomly generated instances. n_x and n_z refer to the dimension of the optimization variables and uncertain parameters respectively, and k refers to the number of uncertain constraints. Nom Obj denotes the objective value of the nominal problem without uncertainty, Opt denotes the optimal value, Time denotes the computational time of Algorithm 1 in seconds and Cut Pl denotes the number of cutting planes generated in Algorithm 1.

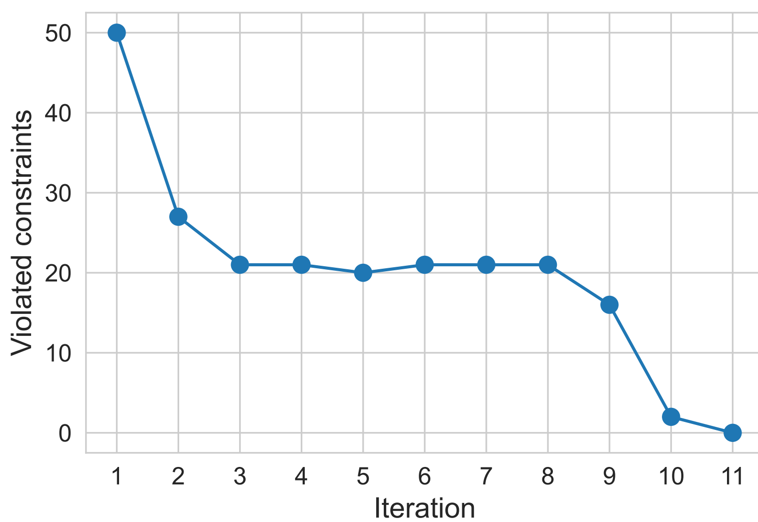

Figure 1 Number of violated constraints in each iteration of Algorithm 1 applied to Problem (17). We utilize $n_z = 10$ and $k = 50$.

4.2. Linear optimization with data uncertainty and implementation error

We consider a linear optimization problem with data uncertainty along with implementation error, formulated as follows:

$$\begin{array}{lll} \min\limits_{\boldsymbol{x}} & \boldsymbol{c}^\top \boldsymbol{x} & \\ \text{s.t.} & \boldsymbol{a}_i^\top \boldsymbol{x} \leq b_i, & \forall i \in [m_1], \\ & (\boldsymbol{a}_i + \boldsymbol{z})^\top (\boldsymbol{x} + \boldsymbol{v}) \leq b_i, & \forall i \in [m_2], \forall (\boldsymbol{z}, \boldsymbol{v}) \in \mathcal{Z}, \\ & \boldsymbol{x} \geq \boldsymbol{0}. & \end{array} \tag{20}$$

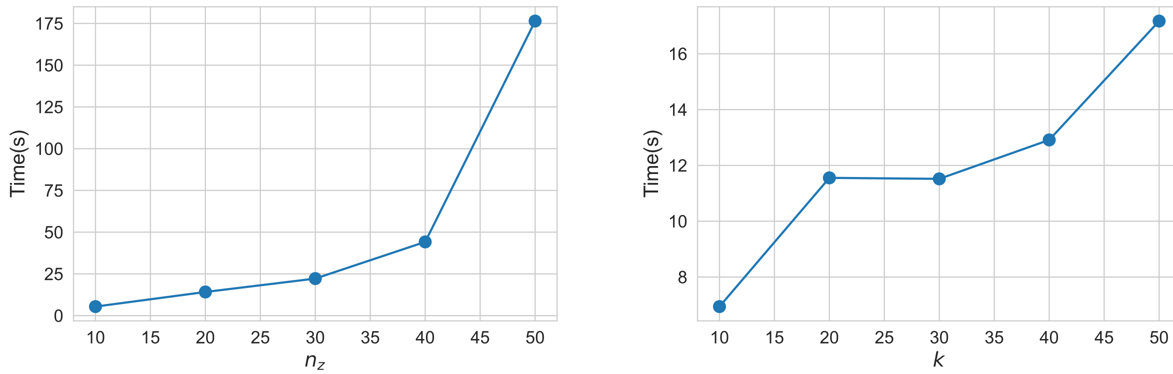

Figure 2 Computational time of Algorithm 1 applied to Problem (17), by varying n_z with fixed $k=10$ (left), as well as by varying k with fixed $n_z=10$ (right).

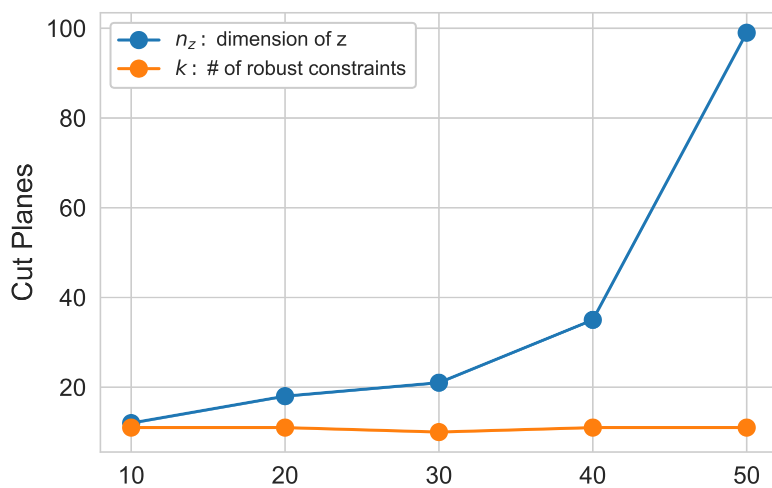

Figure 3 Number of cutting planes generated in Algorithm 1 applied to Problem (17), by varying n_z with fixed $k=10$ (blue line) as well as varying k with fixed $n_z=10$ (orange line).

When applying Algorithm 1, we have the following master-problem:

$$\begin{array}{lll} \min\limits_{\boldsymbol{x}} & \boldsymbol{c}^\top \boldsymbol{x} & \\ \text{s.t.} & \boldsymbol{a}_i^\top \boldsymbol{x} \leq b_i, & \forall i \in [m_1], \\ & (\boldsymbol{a}_i + \boldsymbol{z})^\top (\boldsymbol{x} + \boldsymbol{v}) \leq b_i, & \forall i \in [m_2],\ \forall (\boldsymbol{z}, \boldsymbol{v}) \in \mathcal{S}_i, \\ & \boldsymbol{x} \geq \boldsymbol{0}, & \end{array} \tag{21}$$

where $\mathcal{S}_i$ denotes the list of finite scenarios for the i-th robust constraint. Moreover, we have the following i-th subproblem:

$$\begin{array}{ll} \max\limits_{\boldsymbol{z}, \boldsymbol{v}} & \boldsymbol{a}_i^\top \boldsymbol{x} + \boldsymbol{a}_i^\top \boldsymbol{v} + \boldsymbol{z}^\top \boldsymbol{x} + \boldsymbol{z}^\top \boldsymbol{v} \\ \text{s.t.} & (\boldsymbol{z}, \boldsymbol{v}) \in \mathcal{Z}. \end{array} \tag{22}$$

We consider the uncertainty sets

$$\mathcal{Z}^1 = \{(\boldsymbol{z}, \boldsymbol{v}) : \|\boldsymbol{z}\|_\infty \le \rho,\ \|\boldsymbol{v}\|_1 \le \theta\},$$
$$\mathcal{Z}^2 = \{(\boldsymbol{z}, \boldsymbol{v}) : \|\boldsymbol{z}\|_\infty \le \rho,\ \|\boldsymbol{v}\|_\infty \le \rho,\ \|\boldsymbol{v}\|_1 \le \theta\}.$$

Our proposed initialization scheme with Algorithm 2 is discussed in Appendix B.2. Moreover, Assumption 3 is satisfied as is shown in Appendix C.2.

In Table 2, we illustrate results of our approach for Problem (20) using uncertainty set $\mathcal{Z}^1$ and in Tables 3 and 4 using uncertainty set $\mathcal{Z}^2$ for different values of ρ and θ. We initialize Algorithm 1 either from a random feasible solution $\boldsymbol{z}^0$ or from Algorithm 2. In case of random initialization we report the average result over 10 random choices of $\boldsymbol{z}^0$. The data are generated as $\boldsymbol{a}_i \sim [0,1]^{n_x}$ and $b_i \sim [10, 20]$ and each entry consists of the average over 10 such randomly generated instances. Throughout we take $m_1 = n_x - k$ and $m_2 = k$.

n_z	k	n_x	Nom Obj	Our Approach w random			Our Approach w init		
				Opt	Time(s)	Cut Pl	Opt	Time(s)	Cut Pl
10	5	10	-27.05	-16.91	2.45	6.28	-16.91	1.73	4.10
20	10	20	-26.72	-15.73	55.50	14.89	-15.73	49.51	12.11
30	15	30	-33.86	-18.86	982.83	39.19	-18.86	911.57	34.92
40	20	40	-31.62	-16.83	5211.67	57.33	-16.83	4556.77	44.50

Table 2 Algorithm 1 results for Problem (20) with uncertainty set $\mathcal{Z}^1$, initialized either from a random feasible solution or from Algorithm 2, where the results reflect the average of 10 randomly generated instances. We fix $\rho = \theta = 0.5$. n_x and n_z refer to the dimension of the optimization variables and uncertain parameters, respectively, and k refers to the number of uncertain constraints. Opt denotes the optimal value, Time denotes the computational time of Algorithm 1 in seconds including the initialization and Cut Pl denotes the number of cutting planes generated in Algorithm 1. Finally, Nom Obj denotes the optimal objective value of the nominal problem without uncertainty.

From Tables 2, 3 and 4, we observe that our approach is efficient in problems of dimensions $n_z = 10, 20, 30$. We notice a significant increase in computational time from $n_z = 30$ to $n_z = 40$. Further, we note that additional experiments for $n_z = 50$ have shown an increase in computational time to more than 10000 seconds. We also observe that using Algorithm 2 for initialization can improve the overall computational time, see for example instances $n_z = 20$ and $n_z = 30$ in Table 4, however this is not always the case, see for example instances $n_z = 20, 30$ in Table 3. The reason is that Algorithm 2 requires the solution of additional optimization problems which can increase the overall computational time. Moreover, we notice that when initializing with Algorithm 2 the number of cutting planes generated in Algorithm 1 is always smaller than the case of random initialization.

n_z	k	n_x	Nom Obj	Our Approach w random			Our Approach w init		
				Opt	Time(s)	Cut Pl	Opt	Time(s)	Cut Pl
10	5	10	-27.05	-24.48	3.83	1.35	-24.48	2.95	0.31
20	10	20	-26.72	-23.97	63.75	4.06	-23.97	69.56	0.98
30	15	30	-33.86	-29.23	601.89	10.86	-29.23	765.11	4.82
40	20	40	-31.62	-26.77	4188.26	20.89	-26.77	2729.33	8.15

Table 3 Algorithm 1 results for Problem (20) with uncertainty set $\mathcal{Z}^2$, initialized either from a random feasible solution or from Algorithm 2, where the results reflect the average of 10 randomly generated instances. We fix $\rho = 0.1$ and $\theta = 0.5$. n_x and n_z refer to the dimension of the optimization variables and uncertain parameters, respectively, and k refers to the number of uncertain constraints. Opt denotes the optimal value, Time denotes the computational time of Algorithm 1 in seconds including the initialization and Cut Pl denotes the number of cutting planes generated in Algorithm 1. Finally, Nom Obj denotes the optimal objective value of the nominal problem without uncertainty.

n_z	k	n_x	Nom Obj	Our Approach w random			Our Approach w init		
				Opt	Time(s)	Cut Pl	Opt	Time(s)	Cut Pl
10	5	10	-27.05	-18.20	6.53	4.74	18.20	5.24	2.81
20	10	20	-26.72	-16.71	90.68	12.97	-16.71	89.06	4.72
30	15	30	-33.86	-20.07	1299.55	30.41	-20.07	1210.03	18.01
40	20	40	-31.62	-18.78	6712.56	48.29	-18.78	7001.68	43.89

Table 4 Algorithm 1 results for Problem (20) with uncertainty set $\mathcal{Z}^2$, initialized either from a random feasible solution or from Algorithm 2, where the results reflect the average of 10 randomly generated instances. We fix $\rho = 0.4$ and $\theta = 0.8$. n_x and n_z refer to the dimension of the optimization variables and uncertain parameters, respectively, and k refers to the number of uncertain constraints. Opt denotes the optimal value, Time denotes the computational time of Algorithm 1 in seconds including the initialization and Cut Pl denotes the number of cutting planes generated in Algorithm 1. Finally, Nom Obj denotes the optimal objective value of the nominal problem without uncertainty.

Next, in Figure 4 we illustrate how the number of violated constraints changes in each iteration of Algorithm 1.

Further, in Figure 5 we illustrate how the computational time of Algorithm 1 varies with the dimension of the uncertain parameters (n_z), as well as the number of robust constraints (k). We observe that the computational time increases in both cases, with that of n_z being more significant. More precisely, by varying n_z from 10 to 40, we observe an increase in computational time from approximately 10 seconds to approximately 5000 seconds, while by varying k from 10 to 40 we observe an increase in computational time from approximately 10 seconds to approximately 50 seconds.

In Figure 6 we show how the number of cutting planes generated in Algorithm 1 varies with the dimension of the uncertain parameters as well as the number of robust constraints. We observe that the number of generated cutting planes increases when the dimension of the uncertain parameters

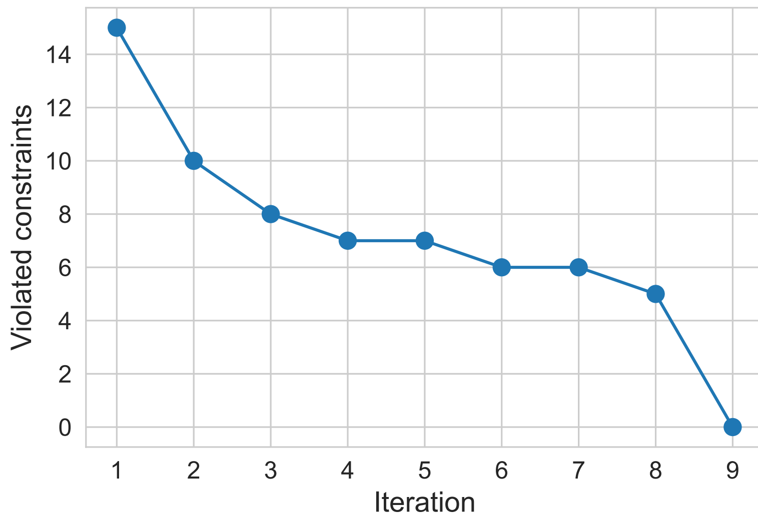

Figure 4 Number of violated constraints in each iteration of Algorithm 1 for Problem (20) with uncertainty set $\mathcal{Z}^1$ and $n_z = 40, k = 20, \theta = 0.5, \rho = 0.5$.

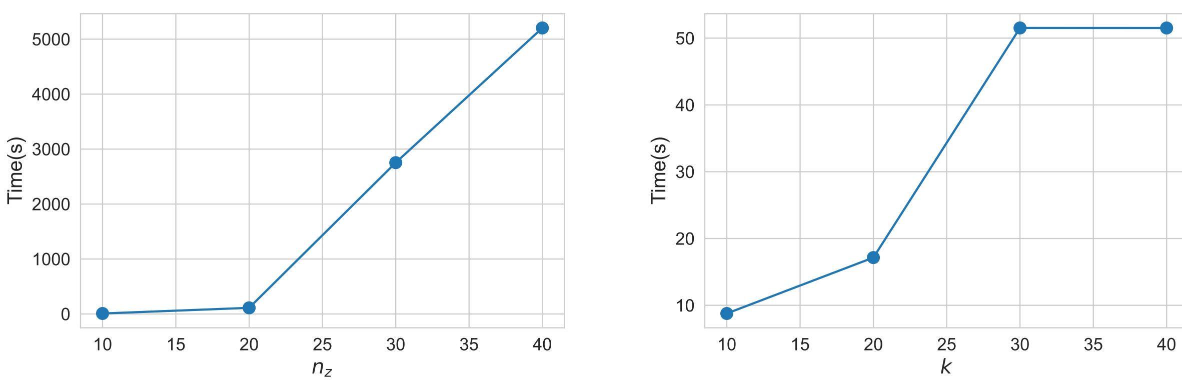

Figure 5 Computational time of Algorithm 1 applied to Problem (20), by varying n_z with fixed $k = 10$ (left), as well as by varying k with fixed $n_z = 10$ (right). We utilize uncertainty set $\mathcal{Z}^2$ and $\rho = 0.5, \theta = 1$.

increases, while it does not change significantly when the number of robust constraints increases, an observation that aligns with Figure 3.

Further, in Figures 7 and 8, we illustrate how the computational time and number of generated cutting planes of Algorithm 1 vary with the size of the uncertainty set, respectively.

From Figures 7 and 8, we observe that both the computational time of Algorithm 1 and the number of generated cutting planes increase, as the size of the uncertainty set increases.

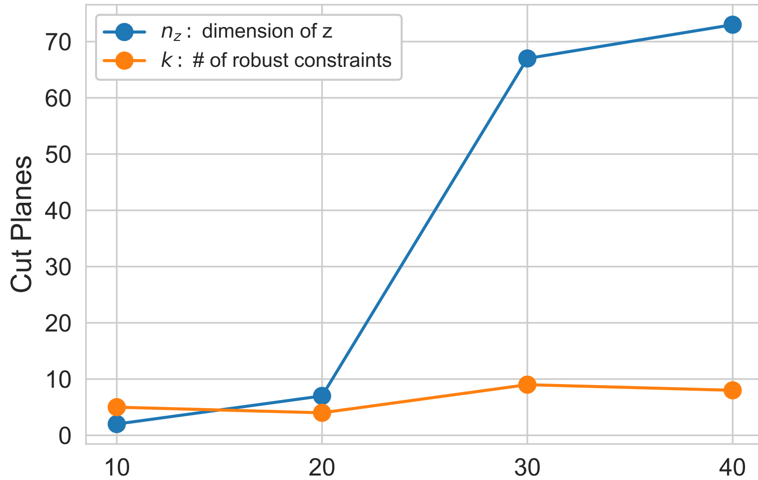

Figure 6 Number of cutting planes generated in Algorithm 1 applied to Problem (20), by varying n_z with fixed $k = 10$ (blue line) as well as by varying k with fixed $n_z = 10$ (orange line). We utilize uncertainty set $\mathcal{Z}^2$ and $\rho = 0.5, \theta = 1$.

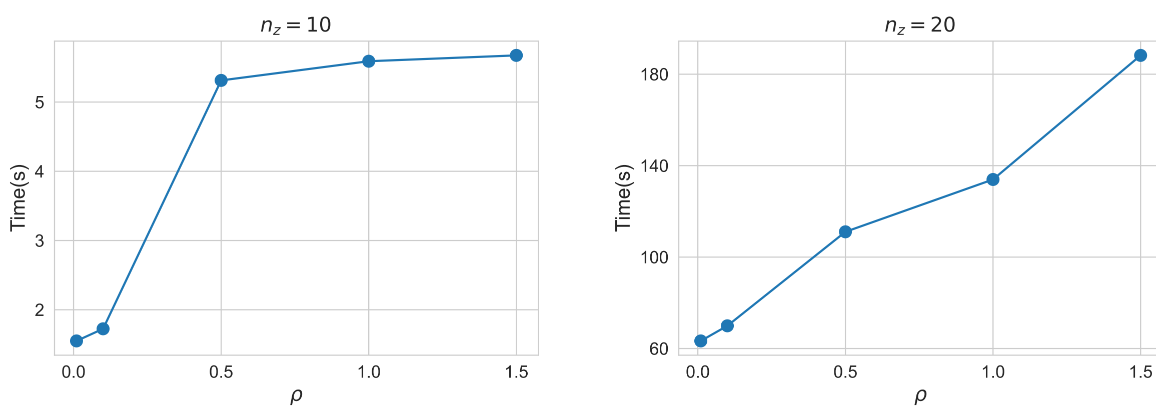

Figure 7 Computational time of Algorithm 1 for Problem (20) with ρ, for fixed $\theta = 2$.

4.3. Adaptive lot sizing

In the classical lot sizing problem in a network we have variables $x_i, i \in [N]$, encoding the stock allocation, with unit cost c_i and maximum capacity V_i, that have to be determined prior to knowing the realization of the demand at each location. After realizing the demand $\boldsymbol{z}$, we decide how much stock to transport from store i to store j, denoted by y_{ij}, with unit cost t_{ij}, in order to meet demand. The problem of minimizing total cost is formulated as follows Zhen et al. (2018), Xu and Burer

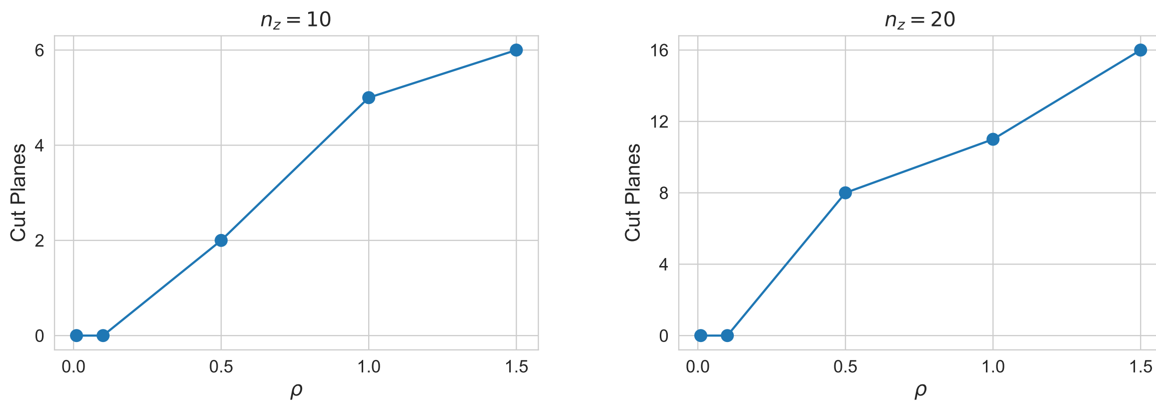

Figure 8 Number of cutting planes generated in Algorithm 1 for Problem (20) with ρ, for fixed $\theta = 2$.

(2018):

$$\min_{\boldsymbol{x},\boldsymbol{y}()} \max_{\boldsymbol{z}\in\mathcal{Z}} \quad \boldsymbol{c}^\top \boldsymbol{x} + \sum_{i,j} t_{ij} y_{ij}(\boldsymbol{z}) \tag{23a}$$

$$\text{s.t.} \quad \sum_j y_{ji}(\boldsymbol{z}) - \sum_j y_{ij}(\boldsymbol{z}) \geq z_i - x_i, \qquad \forall \boldsymbol{z} \in \mathcal{Z},\ i \in [N], \tag{23b}$$

$$y_{ij}(\boldsymbol{z}) \geq 0, \qquad \forall \boldsymbol{z} \in \mathcal{Z}, i, j \in [N], \tag{23c}$$

$$x_i \leq V_i, \qquad i \in [N], \tag{23d}$$

$$x_i \geq 0, \qquad i \in [N]. \tag{23e}$$

Let $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^N_+ : x_i \leq V_i,\ \forall i \in [N]\}$. We consider the following min-max-min formulation

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{z}\in\mathcal{Z}} \min_{\boldsymbol{y}\geq \boldsymbol{0}} \quad \boldsymbol{c}^\top \boldsymbol{x} + \sum_{i,j} t_{ij} y_{ij} \tag{24a}$$

$$\text{s.t.} \quad \sum_j y_{ji} - \sum_j y_{ij} \geq z_i - x_i, \qquad \forall \boldsymbol{z} \in \mathcal{Z},\ i \in [N]. \tag{24b}$$

We introduce an epigraph variable α for the second stage cost in the objective and obtain the following problem:

$$\min_{\boldsymbol{x}\in\mathcal{X},\alpha} \max_{\boldsymbol{z}\in\mathcal{Z}} \min_{\boldsymbol{y}\geq \boldsymbol{0}} \quad \boldsymbol{c}^\top \boldsymbol{x} + \alpha \tag{25a}$$

$$\text{s.t.} \quad \sum_{i,j} t_{ij} y_{ij} \leq \alpha, \tag{25b}$$

$$\sum_j y_{ji} - \sum_j y_{ij} \geq z_i - x_i, \qquad \forall \boldsymbol{z} \in \mathcal{Z},\ i \in [N]. \tag{25c}$$

The dual of the inner minimization problem for $\boldsymbol{y}$ is as follows:

$$\max_{\boldsymbol{p}\geq \boldsymbol{0}, s\geq 0} \quad \boldsymbol{p}^\top \boldsymbol{z} - \boldsymbol{p}^\top \boldsymbol{x} - s\alpha \tag{26a}$$

$$\text{s.t.} \quad s t_{ij} + p_i - p_j \geq 0, \qquad \forall i, j \in [N],\ i \neq j. \tag{26b}$$

Let $\mathcal{P} = \{(\boldsymbol{p}, s) \geq \boldsymbol{0} : st_{ij} + p_i - p_j \geq 0,\ \forall i, j \in [N],\ i \neq j\}$. We have the following problem:

$$\min_{\boldsymbol{x}\in\mathcal{X},\alpha} \max_{\boldsymbol{z}\in\mathcal{Z}} \max_{(\boldsymbol{p},s)\in\mathcal{P}} \boldsymbol{c}^\top \boldsymbol{x} + \alpha + \boldsymbol{p}^\top \boldsymbol{z} - \boldsymbol{p}^\top \boldsymbol{x} - s\alpha.$$

By combining the two maximization problems we obtain the following problem:

$$\begin{aligned}
\min_{\boldsymbol{x}\geq\boldsymbol{0},\alpha} \quad & \boldsymbol{c}^\top \boldsymbol{x} + \alpha && && \text{(27a)}\\
\text{s.t.} \quad & \boldsymbol{p}^\top \boldsymbol{z} - \boldsymbol{p}^\top \boldsymbol{x} - s\alpha \leq 0, && \forall (\boldsymbol{z}, \boldsymbol{p}, s) \in \mathcal{Z} \times \mathcal{P}, && \text{(27b)}\\
& x_i \leq V_i, && i \in [N]. && \text{(27c)}
\end{aligned}$$

Observe that the constraints for the dual variables $\tilde{\boldsymbol{p}} = (\boldsymbol{p}, s)$ can be written using vector notation as $\boldsymbol{B}\tilde{\boldsymbol{p}} \leq \boldsymbol{0}$ and therefore following the approach from Bertsimas and de Ruiter (2016), we can add the constraint $\sum_{i=1}^n p_i + s = 1$, at no additional cost. We assume the box-budget uncertainty set with non-negativity constraints for $\boldsymbol{z}$, that is, $\mathcal{Z} = \{\boldsymbol{z} \in \mathbb{R}^N_+ : \|\boldsymbol{z}\|_\infty \leq \rho,\ \|\boldsymbol{z}\|_1 \leq \theta\}$. The subproblem that needs to be solved in order to add scenarios in this case, is as follows:

$$\begin{aligned}
\max_{\boldsymbol{z},\boldsymbol{p},s} \quad & \boldsymbol{p}^\top \boldsymbol{z} - \boldsymbol{p}^\top \boldsymbol{x} - s\alpha && && \\
\text{s.t.} \quad & st_{ij} + p_i - p_j \geq 0, && \forall i, j \in [N],\ i \neq j, && \text{(28a)}\\
& \sum_{i=1}^N p_i + s = 1, && && \text{(28b)}\\
& z_i \leq \rho, && \forall i \in [N], && \text{(28c)}\\
& \sum_{i=1}^N z_i \leq \theta, && && \text{(28d)}\\
& \boldsymbol{z}, \boldsymbol{p}, s \geq 0. && && \text{(28e)}
\end{aligned}$$

Our proposed initialization scheme with Algorithm 2 is discussed in Appendix B.3. Moreover, Assumption 3 is satisfied as is shown in Appendix C.3.

We next provide results for our approach as well as the LDR approach in Table 5. From Table 5, we observe that our approach can efficiently solve problems up to $N = 17$ in less than one hour. Further, we notice a significant change in the computational time when increasing N from 15 to 17. We note that for $N \geq 18$ the computational time increases to more than one hour. The efficiency of our proposed method aligns with the Column-and-Constraint Generation (CCG) approach by Zeng and Zhao (2013). For a detailed examination of the CCG-based approach's efficiency, we refer to Zhen and de Ruiter (2019). Our study did not incorporate the CCG approach, as it is primarily designed for two-stage robust linear problems with polyhedral uncertainties, which makes it unsuitable for nonlinear problems. In contrast, our method is adaptable and effectively addresses two-stage nonlinear problems, even when dealing with non-polyhedral uncertainty sets.

N	LDR	Our Approach		
		Opt	Time(s)	Cut Pl
5	1170.72	974.31	6.31	5.30
10	1790.71	1337.69	264.85	7.60
15	2363.22	1661.83	1661.51	6.70
17	2535.32	1669.34	2985.58	6.40

Table 5 Algorithm 1 results for Problem (23) with uncertainty set $\mathcal{Z}$, initialized from Algorithm 2, where the results reflect the average of 10 randomly generated instances. Opt denotes the optimal value, Time denotes the computational time of Algorithm 1 in seconds including the initialization and Cut Pl denotes the number of cutting planes generated in Algorithm 1. Finally, LDR denotes the optimal objective value of the problem solved with linear decision rules.

5. Conclusions

In summary, we have developed a method for globally solving optimization problems containing robust constraints, which are SLC with respect to uncertain parameters. By utilizing a cutting set method, we apply the RPT-BB approach of Bertsimas et al. (2023a) to solve each subproblem exactly. Our method is further extended to two-stage ARO problems with fixed recourse. We have demonstrated that our approach converges to the global optimum under mild assumptions and proposed various initialization strategies that leverage the problem's structure. The effectiveness of our method is confirmed through numerical experiments on diverse problems, including a robust geometric optimization problem, a linear optimization problem with data uncertainty and implementation error, and an adaptive robust lot-sizing problem on a network. We found that our approach is computationally feasible for practical-sized instances across all considered problems. Additionally, our proposed initialization schemes frequently enhance the computational efficiency of our method. A disadvantage of our approach is that the number of uncertain parameters is squared. An interesting topic for further research is to develop variants of the RPT-BB approach that avoid this increase in the number of variables.

Acknowledgments

The second author of this paper is funded by NWO grant 406.18.EB.003.

References

A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4): 769–805, 1998.

A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.

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

A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

A. Ben-Tal, D. den Hertog, and J.-Ph. Vial. Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical Programming*, 149:265–299, 2015.

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

D. Bertsimas and D. den Hertog. *Robust and Adaptive Optimization*. Dynamic Ideas LLC, Belmont, 2022.

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

D. Bertsimas, D. de Moor, D. den Hertog, T. Koukouvinos, and J. Zhen. A novel algorithm for a broad class of nonconvex optimization problems. *Optimization Online*, 2023a. URL http://www.optimization-online.org/DB_FILE/2021/07/8491.pdf.

D. Bertsimas, D. de Moor, D. den Hertog, T. Koukouvinos, and J. Zhen. Cone product reformulation for global optimization. *Optimization Online*, 2023b. URL https://optimization-online.org/?p=24323.

D. Bertsimas, D. den Hertog, J. Pauphilet, and J. Zhen. Robust convex optimization: A new perspective that unifies and extends. *Mathematical Programming*, 200(2):877–918, 2023c.

L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

L. El Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9(1):33–52, 1998.

B.L. Gorissen, D. den Hertog, and M. Reusken. Hidden convexity in a class of optimization problems with bilinear terms. *Optimization Online*, 2022. URL https://optimization-online.org/?p=19067.

G. Hanasusanto and D. Kuhn. Conic programming reformulations of two-stage distributionally robust linear programs over wasserstein balls. *Operations Research*, 66(3):849–869, 2018.

J. E. Kelley Jr. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 10.1.*, 2024. URL http://docs.mosek.com/10.1/toolbox/index.html.

A. Mutapcic and S. Boyd. Cutting-set methods for robust convex optimization with pessimizing oracles. *Optimization Methods & Software*, 24(3):381–406, 2009.

R.T. Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics and Physics. Princeton University Press, 1970.

A. Soyster. Convex programming with set-inclusive constraints. *Operations Research*, 21(5):1154–1157, 1973.

P. D. Tao and L.T.H. An. Convex analysis approach to d. c. programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355, 1997.

A. Thiele, T. Terry, and M. Epelman. Robust linear optimization with recourse. *Optimization Online*, 2010. URL https://optimization-online.org/?p=10712.

G. Xu and A. Burer. A copositive approach for two-stage adjustable robust optimization with uncertain right-hand sides. *Computational Optimization and Applications*, 70(1):33–59, 2018.

B. Zeng and L. Zhao. Solving two-stage robust optimization problems using a column and-constraint generation method. *Operations Research Letters*, 41(5):457–461, 2013.

J. Zhen and F. de Ruiter. A primal-dual perspective on adaptive robust linear optimization. *Optimization Online*, 2019. URL https://optimization-online.org/?p=16060.

J. Zhen, D. den Hertog, and M. Sim. Adjustable robust optimization via fourier–motzkin elimination. *Operations Research*, 66(4):1086–1100, 2018.

J. Zhen, D. Kuhn, and W. Wiesemann. A unified theory of robust and distributionally robust optimization via the primal-worst-equals-dual-best principle. *Operations Research*, 2023.

Appendix

A. Lemma A.1

Lemma 1. *Let*

$$\tilde{f}_{ik}(\boldsymbol{z},\boldsymbol{U}) = (q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}) f_{ik}\left(\frac{q_{ik}\boldsymbol{z} - \boldsymbol{U} p_{ik}}{q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}}\right).$$

Then the conjugate of $\tilde{f}_{ik}$, $(\tilde{f}_{ik})_*(\boldsymbol{w},\boldsymbol{W})$, equals

$$\left\{ -q_{ik}^2 \frac{W_{11}}{(p_{ik})_1^2} + q_{ik}\frac{w_1}{(p_{ik})_1} \left| \begin{array}{l} \frac{\boldsymbol{W}_1}{(p_{ik})_1} = \frac{\boldsymbol{W}_j}{(p_{ik})_j}, \quad j \in [n_z^{'}], \\ q_k \frac{W_{11}}{(p_{ik})_1^2} - \frac{w_1}{(p_{ik})_1} = q_{ik}\frac{W_{1j}}{(p_{ik})_1 (p_{ik})_j} - \frac{w_j}{(p_{ik})_j},\ j \in [n_z^{'}], \\ (f_{ik})_*\left(\frac{\boldsymbol{W}_1}{(p_{ik})_1}\right) + q_{ik}\frac{W_{11}}{(p_{ik})_1^2} - \frac{w_1}{(p_{ik})_1} \le 0 \end{array} \right. \right\},$$

where $[n_z^{'}] = [n_z] \setminus \{1\}$.

Proof. We have that $(\tilde{f}_{ik})_*(\boldsymbol{v},\boldsymbol{V}) =$

$$\begin{aligned}
&\inf_{\boldsymbol{z},\boldsymbol{U}} \left\{ \boldsymbol{v}^\top \boldsymbol{z} + \mathrm{Tr}(\boldsymbol{U}\boldsymbol{V}^\top) - (q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}) f_{ik}\left(\frac{q_{ik}\boldsymbol{z} - \boldsymbol{U}\boldsymbol{p}_{ik}}{q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}}\right)\right\} \\
&= \inf_{\boldsymbol{z},\boldsymbol{U}} (q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}) \left\{ \frac{\boldsymbol{v}^\top \boldsymbol{z}}{q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}} + \frac{\mathrm{Tr}(\boldsymbol{U}\boldsymbol{V}^\top)}{q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}} - f_{ik}\left(\frac{q_{ik}\boldsymbol{z} - \boldsymbol{U}\boldsymbol{p}_{ik}}{q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z}}\right)\right\} \\
&= \inf_{\boldsymbol{z},\boldsymbol{U},t_{ik}} \left\{ t_{ik}\left(\frac{\boldsymbol{v}^\top \boldsymbol{z}}{t_{ik}} + \frac{\mathrm{Tr}(\boldsymbol{U}\boldsymbol{V}^\top)}{t_{ik}} - f_{ik}\left(\frac{q_{ik}\boldsymbol{z} - \boldsymbol{U}\boldsymbol{p}_{ik}}{t_{ik}}\right)\right) \;\middle|\; q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z} = t_{ik} \right\} \\
&= \inf_{\boldsymbol{z},\boldsymbol{U},t_{ik},\boldsymbol{w}_{ik}} \left\{ t_{ik}\left(\frac{\boldsymbol{v}^\top \boldsymbol{z}}{t_{ik}} + \frac{\mathrm{Tr}(\boldsymbol{U}\boldsymbol{V}^\top)}{t_{ik}} - f_{ik}\left(\frac{\boldsymbol{w}_{ik}}{t_{ik}}\right)\right) \;\middle|\; \begin{array}{l} q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z} = t_{ik}, \\ q_{ik}\boldsymbol{z} - \boldsymbol{U}\boldsymbol{p}_{ik} = \boldsymbol{w}_{ik} \end{array} \right\} \\
&= \sup_{y_{ik},\boldsymbol{\lambda}_{ik}} \inf_{\boldsymbol{z},\boldsymbol{U},t_{ik},\boldsymbol{w}_{ik}} \left\{ t_{ik}\left(\frac{\boldsymbol{v}^\top \boldsymbol{z}}{t_{ik}} + \frac{\mathrm{Tr}(\boldsymbol{U}\boldsymbol{V}^\top)}{t_{ik}} - f_{ik}\left(\frac{\boldsymbol{w}_{ik}}{t_{ik}}\right)\right) \right. \\
&\qquad\qquad \left. - y_{ik}(q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z} - t_{ik}) - \boldsymbol{\lambda}_{ik}^\top (q_{ik}\boldsymbol{z} - \boldsymbol{U}\boldsymbol{p}_{ik} - \boldsymbol{w}_{ik}) \right\} \\
&= \sup_{y_{ik},\boldsymbol{\lambda}_{ik}} \inf_{\boldsymbol{z},\boldsymbol{U},t_{ik},\boldsymbol{\rho}_{ik}} \left\{ t_{ik}\left(\frac{\boldsymbol{v}^\top \boldsymbol{z}}{t_{ik}} + \frac{\mathrm{Tr}(\boldsymbol{U}\boldsymbol{V}^\top)}{t_{ik}} + \boldsymbol{\lambda}^\top \boldsymbol{\rho}_{ik} - f_{ik}(\boldsymbol{\rho}_{ik})\right) \right. \\
&\qquad\qquad \left. - y(q_{ik} - \boldsymbol{p}_{ik}^\top \boldsymbol{z} - t_{ik}) - \boldsymbol{\lambda}_{ik}^\top (q_{ik}\boldsymbol{z} - \boldsymbol{U}\boldsymbol{p}_{ik}) \right\} \\
&= \sup_{y_{ik},\boldsymbol{\lambda}_{ik}} \inf_{\boldsymbol{z},\boldsymbol{U},t_{ik},\boldsymbol{\rho}_{ik}} \left\{ t_{ik}(\frac{\boldsymbol{v}^\top \boldsymbol{z}}{t_{ik}} + \frac{\mathrm{Tr}(\boldsymbol{U}\boldsymbol{V}^\top)}{t_{ik}} + \boldsymbol{\lambda}_{ik}^\top \boldsymbol{\rho}_{ik} - f_{ik}(\boldsymbol{\rho}_{ik}) + y_{ik}\boldsymbol{p}_{ik}^\top \frac{\boldsymbol{z}}{t_{ik}} \right. \\
&\qquad\qquad \left. - q_{ik}\boldsymbol{\lambda}^\top \frac{\boldsymbol{z}}{t_{ik}}) - y_{ik}(q_{ik} - t_{ik}) + \boldsymbol{\lambda}_{ik}^\top \boldsymbol{U}\boldsymbol{p}_{ik} \right\} \\
&= \sup_{y_{ik},\boldsymbol{\lambda}_{ik}} \inf_{\boldsymbol{r}_{ik},\boldsymbol{U},t_{ik},\boldsymbol{\rho}_{ik}} \left\{ t_{ik}(\boldsymbol{v}^\top \boldsymbol{r} + \frac{\mathrm{Tr}(\boldsymbol{U}\boldsymbol{V}^\top)}{t_{ik}} + \boldsymbol{\lambda}_{ik}^\top \boldsymbol{\rho}_{ik} - f_{ik}(\boldsymbol{\rho}_{ik}) + y_{ik}\boldsymbol{p}_{ik}^\top \boldsymbol{r}_{ik} \right. \\
&\qquad\qquad \left. - q_{ik}\boldsymbol{\lambda}^\top \boldsymbol{r}_{ik}) - y_{ik}(q_{ik} - t_{ik}) + \boldsymbol{\lambda}_{ik}^\top \boldsymbol{U}\boldsymbol{p}_{ik} \right\} \\
&= \sup_{y_{ik},\boldsymbol{\lambda}_{ik}} \inf_{\boldsymbol{r}_{ik},\boldsymbol{U},t_{ik},\boldsymbol{\rho}_{ik}} \{ t_{ik}(\boldsymbol{v}^\top \boldsymbol{r}_{ik} + \frac{\mathrm{Tr}(\boldsymbol{U}\boldsymbol{V}^\top)}{t_{ik}} + \boldsymbol{\lambda}_{ik}^\top \boldsymbol{\rho}_{ik} - f_{ik}(\boldsymbol{\rho}_{ik}) + y_{ik}\boldsymbol{p}_{ik}^\top \boldsymbol{r}_{ik} \\
&\qquad\qquad - q_{ik}\boldsymbol{\lambda}_{ik}^\top \boldsymbol{r}_{ik} + y_{ik}) - y_{ik}q_{ik} + \boldsymbol{\lambda}_{ik}^\top \boldsymbol{U}\boldsymbol{p}_{ik} \} \\
&= \sup_{y_{ik},\boldsymbol{\lambda}_{ik}} \inf_{\boldsymbol{r}_{ik},\boldsymbol{W}_{ik},t_{ik},\boldsymbol{\rho}_{ik}} \{ t_{ik}(\boldsymbol{v}^\top \boldsymbol{r}_{ik} + \mathrm{Tr}(\boldsymbol{W_{ik}}\boldsymbol{V}^\top) + \boldsymbol{\lambda}^\top \boldsymbol{\rho}_{ik} - f_{ik}(\boldsymbol{\rho}_{ik}) + y_{ik}\boldsymbol{p}_{ik}^\top \boldsymbol{r}_{ik}
\end{aligned}$$

$$
\begin{aligned}
&\qquad\qquad\qquad - q_{ik}\boldsymbol{\lambda}_{ik}^\top \boldsymbol{r}_{ik} + y_{ik} + \boldsymbol{\lambda}_{ik}^\top \boldsymbol{W}_{ik}\boldsymbol{p}_{ik}) - y_{ik}q_{ik}\} \\
&= \sup_{y_{ik},\boldsymbol{\lambda}_{ik}} \inf_{\boldsymbol{r}_{ik},\boldsymbol{W}_{ik},t_{ik},\boldsymbol{\rho}_{ik}} \left\{ t_{ik}(\boldsymbol{r}_{ik}^\top(\boldsymbol{v} + y_{ik}\boldsymbol{p}_{ik} - q_{ik}\boldsymbol{\lambda}_{ik}) + \mathrm{Tr}\left(\boldsymbol{W}_{ik}(\boldsymbol{V}^\top + \boldsymbol{p}_{ik}\boldsymbol{\lambda}_{ik}^\top)\right) \right. \\
&\qquad\qquad\qquad \left. + \boldsymbol{\lambda}_{ik}^\top \boldsymbol{\rho}_{ik} - f_{ik}(\boldsymbol{\rho}_{ik}) + y_{ik} - y_{ik}q_{ik} \right\} \\
&= \sup_{y_{ik},\boldsymbol{\lambda}_{ik}} \left\{ -y_{ik}q_{ik} \middle| \boldsymbol{v} + y_{ik}\boldsymbol{p}_{ik} = q_{ik}\boldsymbol{\lambda}_{ik},\ \ \boldsymbol{V} = -\boldsymbol{\lambda}_{ik}\boldsymbol{p}_{ik}^\top,\ \ (f_{ik})_*(\boldsymbol{\lambda}_{ik}) + y_{ik} \geq 0 \right\} \\
&= \left\{ q_{ik}^2 \frac{V_{11}}{(p_{ik})_1^2} + q_{ik}\frac{v_1}{(p_{ik})_1} \middle| \begin{array}{l} \frac{\boldsymbol{V}_1}{(p_{ik})_1} = \frac{\boldsymbol{V}_j}{(p_{ik})_j}, \quad j \in [n_z'] \\ q_{ik}\frac{V_{11}}{(p_{ik})_1^2} + \frac{v_1}{(p_{ik})_1} = q_{ik}\frac{V_{1j}}{(p_{ik})_1(p_{ik})_j} + \frac{v_j}{(p_{ik})_j}, \quad j \in [n_z'] \\ (f_{ik})_*\left(\frac{-\boldsymbol{V}_1}{(p_{ik})_1}\right) - q_{ik}\frac{V_{11}}{(p_{ik})_1^2} - \frac{v_1}{(p_{ik})_1} \geq 0 \end{array} \right\}.
\end{aligned}
$$

B. RC and RPT-formulation of numerical experiments

B.1. Robust convex geometric optimization

Bertsimas et al. (2023c) show that (17) can be approximated by

$$
\begin{array}{ll}
\max_{\boldsymbol{x}} & -c^\top \boldsymbol{x} \\
\text{s.t.} & (-\boldsymbol{e}w_1 + \boldsymbol{B}_i^{(1)}\boldsymbol{v}_1)^\top \boldsymbol{x} + (-\boldsymbol{e}w_2 + \boldsymbol{B}_i^{(2)}\boldsymbol{v}_2)^\top \boldsymbol{x} \\
& \qquad\qquad -w_0 \leq 0, \quad \forall (\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_0) \in \Theta, \quad \forall i \in \mathcal{I},
\end{array} \tag{29}
$$

where Θ is the approximated uncertainty set, given by (see Bertsimas et al. (2023c))

$$
\Theta = \left\{ \begin{array}{c} \boldsymbol{w} \in \mathbb{R}^3, \\ \boldsymbol{t} \in \mathbb{R}^2, \\ \boldsymbol{v}_1 \in \mathbb{R}^{n_z}, \\ \boldsymbol{v}_2 \in \mathbb{R}^{n_z}, \\ \boldsymbol{v}_0 \in \mathbb{R}^{n_z} \\ \boldsymbol{z} \in \mathbb{R}^{n_z}, \\ \boldsymbol{\lambda}_1 \in \mathbb{R}^{n_z}, \\ \boldsymbol{\lambda}_2 \in \mathbb{R}^{n_z}, \end{array} : \begin{array}{l} \boldsymbol{D}\boldsymbol{z} \leq \boldsymbol{d}, \quad \|\boldsymbol{z}\|_2 \leq \gamma, \\ w_1 + w_2 = 1, \quad w_1, w_2 \geq 0, \quad t_1 + t_2 \leq w_0, \\ w_i \exp\left(\frac{-t_i}{w_i}\right) \leq 1,\ i \in \{1,2\}, \\ \boldsymbol{D}\boldsymbol{v}_i \leq \boldsymbol{d}w_i, \quad \|\boldsymbol{v}_i\|_2 \leq w_i\gamma, \quad i \in \{1,2\}, \\ \boldsymbol{v}_1 + \boldsymbol{v}_2 = \boldsymbol{z}, \\ (d_\ell w_1 - \boldsymbol{D}_\ell \boldsymbol{v}_1)\exp\left(\dfrac{-\boldsymbol{d}_\ell t_1 + \boldsymbol{D}_\ell \boldsymbol{\lambda}_1}{\boldsymbol{d}_\ell w_1 - \boldsymbol{D}_\ell \boldsymbol{v}_1}\right) \leq \boldsymbol{d}_\ell - \boldsymbol{D}_\ell \boldsymbol{z},\ \ell \in \mathcal{L}, \\ (d_\ell w_2 - \boldsymbol{D}_\ell \boldsymbol{v}_2)\exp\left(\dfrac{-\boldsymbol{d}_\ell t_2 + \boldsymbol{D}_\ell \boldsymbol{\lambda}_2}{\boldsymbol{d}_\ell w_2 - \boldsymbol{D}_\ell \boldsymbol{v}_2}\right) \leq \boldsymbol{d}_\ell - \boldsymbol{D}_\ell \boldsymbol{z},\ \ell \in \mathcal{L}, \\ \boldsymbol{D}(\boldsymbol{v}_0 - \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2) \leq \boldsymbol{d}(w_0 - t_1 - t_2), \\ \|\boldsymbol{v}_0 - \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\|_2 \leq \gamma(w_0 - t_1 - t_2). \end{array} \right\}
$$

Here $\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \lambda_1$ and λ_2 linearize the product terms $w_0\boldsymbol{z}, w_1\boldsymbol{z}, w_2\boldsymbol{z}, t_1\boldsymbol{z}$ and $t_2\boldsymbol{z}$, respectively. Bertsimas et al. (2023b) show that by utilizing the the Taylor expansion, the additional redundant constraints $1 \geq w_1 - t_1$, $1 \geq w_2 - t_2$ can be generated and then multiplied with the constraints in Θ to obtain a new approximated uncertainty set Θ'. To compute the RC of Problem (29) we take the dual over $\boldsymbol{x}$ and obtain:

$$
\begin{array}{ll}
\min_{\lambda \geq 0, \boldsymbol{y} \geq 0, \boldsymbol{w}, \boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{z}} & \sum_{i \in \mathcal{I}} w_0 y_i + \alpha\lambda \\
\text{s.t.} & \|-\boldsymbol{c} + \sum_{i \in \mathcal{I}} \left(y_i w_1 \boldsymbol{e} + y_i w_2 \boldsymbol{e} - \boldsymbol{B}_i^1 y_i \boldsymbol{v}_1 - \boldsymbol{B}_i^2 y_i \boldsymbol{v}_2\right)\|_2 \leq \lambda, \\
& (\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_0) \in \Theta'.
\end{array}
$$

By substituting the product terms $y_iw_0, y_iw_1, y_iw_2, y_i\boldsymbol{v}_0, y_i\boldsymbol{v}_1, y_i\boldsymbol{v}_2, y_i\boldsymbol{z}$ by p_0^i, p_1^i, p_2^i, $\boldsymbol{q}_0^i$, $\boldsymbol{q}_1^i$, $\boldsymbol{q}_2^i$, $\boldsymbol{s}^i$, and taking the perspective function of all constraints with respect to y_i we obtain the following equivalent convex reformulation as done in Gorissen et al. (2022):

$$
\begin{aligned}
\min_{\lambda\geq 0, \boldsymbol{y}\geq 0, \boldsymbol{w}, \boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{z}} \quad & \sum_{i\in\mathcal{I}} p_0^i + \alpha\lambda \\
\text{s.t.} \quad & \|-\boldsymbol{c} + \sum_{i\in\mathcal{I}} \left(p_1^i\boldsymbol{e} + p_2^i\boldsymbol{e} - \boldsymbol{B}_i^1\boldsymbol{q}_1^i - \boldsymbol{B}_i^2\boldsymbol{q}_2^i\right)\|_2 \leq \lambda, \\
& y_i g_k\left(\frac{\boldsymbol{p}^i}{y_i}, \frac{\boldsymbol{s}^i}{y_i}, \frac{\boldsymbol{q}_1^i}{y_i}, \frac{\boldsymbol{q}_2^i}{y_i}, \frac{\boldsymbol{q}_0^i}{y_i}\right) \leq 0, \quad i\in\mathcal{I}, k\in\mathcal{K},
\end{aligned}
$$

where g_k represents the k-th constraint in the uncertainty set Θ'. After substituting g_k with each constraint and linearizing all product terms we obtain the final RC.

B.2. Linear optimization with data uncertainty and implementation error

$\mathcal{Z} = \mathcal{Z}_1$ We linearize products among uncertain parameters in the LHS of each uncertain constraint and multiply all constraints in $\mathcal{Z}$. Observe that the constraint $\|\boldsymbol{z}\|_\infty \leq \rho$ can be written as $\boldsymbol{C}\boldsymbol{z} \leq \boldsymbol{d}$, where $\boldsymbol{C} = \begin{bmatrix}\boldsymbol{I}\\ -\boldsymbol{I}\end{bmatrix}$ and $\boldsymbol{d} = \begin{bmatrix}\rho\boldsymbol{e}\\ \rho\boldsymbol{e}\end{bmatrix}$. Further, the constraint $\|\boldsymbol{v}\|_1 \leq \theta$ can be linearized as $\sum_i s_i \leq \theta$, $-\boldsymbol{s} \leq \boldsymbol{v} \leq \boldsymbol{s}$. We then multiply all constraints and add new variables to linearize products $\boldsymbol{Z} = \boldsymbol{z}\boldsymbol{z}^\top$, $\boldsymbol{V} = \boldsymbol{v}\boldsymbol{v}^\top$, $\boldsymbol{S} = \boldsymbol{s}\boldsymbol{s}^\top$, $\boldsymbol{W} = \boldsymbol{z}\boldsymbol{v}^\top$, $\boldsymbol{T} = \boldsymbol{z}\boldsymbol{s}^\top$ and $\boldsymbol{P} = \boldsymbol{v}\boldsymbol{s}^\top$. We obtain the following i-th subproblem:

$$
\begin{aligned}
\max_{\boldsymbol{z},\boldsymbol{v}} \quad & \boldsymbol{a}_i^\top\boldsymbol{x} + \boldsymbol{a}_i^\top\boldsymbol{v} + \boldsymbol{z}^\top\boldsymbol{x} + \mathrm{Tr}(\boldsymbol{W}) \\
\text{s.t.} \quad & \boldsymbol{C}\boldsymbol{z} \leq \boldsymbol{d}, \quad \sum_i s_i \leq \theta, \quad \boldsymbol{v} \leq \boldsymbol{s}, \quad -\boldsymbol{v} \leq \boldsymbol{s}, && \text{(30a)} \\
& d_id_j - d_i\boldsymbol{c}_j^\top\boldsymbol{z} - d_j\boldsymbol{c}_i^\top\boldsymbol{z} + \boldsymbol{c}_i^\top\boldsymbol{Z}\boldsymbol{c}_j \geq 0, \quad i,j\in[L], && \text{(30b)} \\
& \theta^2 - 2\theta\sum_i s_i + \sum_{i,j} S_{ij} \geq 0, && \text{(30c)} \\
& \theta d_i - \theta\boldsymbol{c}_i^\top\boldsymbol{z} - d_i\sum_j s_j + \boldsymbol{c}_i^\top\sum_j \boldsymbol{T}_j \geq 0, \quad i\in[L], && \text{(30d)} \\
& \theta v_j - \sum_i P_{ji} \leq \theta s_j - \sum_i S_{ij}, \quad j\in[n], && \text{(30e)} \\
& \sum_i P_{ji} - \theta v_j \leq \theta s_j - \sum_i S_{ij}, \quad j\in[n], && \text{(30f)} \\
& d_is_j - d_iv_j + \boldsymbol{c}_i^\top\boldsymbol{W}_j - \boldsymbol{c}_i^\top\boldsymbol{T}_j \geq 0, \quad i\in[L], j\in[n], && \text{(30g)} \\
& d_is_j + d_iv_j - \boldsymbol{c}_i^\top\boldsymbol{W}_j - \boldsymbol{c}_i^\top\boldsymbol{T}_j \geq 0, \quad i\in[L], j\in[n], && \text{(30h)} \\
& S_{ij} + P_{ji} - P_{ij} - V_{ij} \geq 0, \quad S_{ij} - P_{ji} - P_{ij} + V_{ij} \geq 0, \quad i,j\in[n], && \text{(30i)} \\
& S_{ij} + P_{ji} + P_{ij} + V_{ij} \geq 0, \quad i,j\in[n],. && \text{(30j)}
\end{aligned}
$$

The dual of Problem (30) is then derived. We then obtain the RC by replacing the LHS of each constraint with the objective of the derived dual and further adding the dual constraints.

$\mathcal{Z} = \mathcal{Z}_2$ The constraint $\|\boldsymbol{v}\|_1 \le \theta$ can be linearized as $\sum_i s_i \le \theta$, $-\boldsymbol{s} \le \boldsymbol{v} \le \boldsymbol{s}$. We obtain the following i-th subproblem:

$$\begin{aligned}
\max_{\boldsymbol{z},\boldsymbol{v}} \quad & \boldsymbol{a}_i^\top \boldsymbol{x} + \boldsymbol{a}_i^\top \boldsymbol{v} + \boldsymbol{z}^\top \boldsymbol{x} + \mathrm{Tr}(\boldsymbol{W}) && && (31a)\\
\text{s.t.} \quad & (30a) - (30j), \quad \boldsymbol{C}\boldsymbol{v} \le \boldsymbol{d}, && && (31b)\\
& d_i d_j - d_i \boldsymbol{c}_j^\top \boldsymbol{v} - d_j \boldsymbol{c}_i^\top \boldsymbol{v} + \boldsymbol{c}_i^\top \boldsymbol{V} \boldsymbol{c}_j \ge 0, && i,j \in [L], && (31c)\\
& d_i d_j - d_i \boldsymbol{c}_j^\top \boldsymbol{v} - d_j \boldsymbol{c}_i^\top \boldsymbol{z} + \boldsymbol{c}_i^\top \boldsymbol{W} \boldsymbol{c}_j \ge 0, && i,j \in [L], && (31d)\\
& \theta d_i - \theta \boldsymbol{c}_i^\top \boldsymbol{v} - d_i \sum_j s_j + \boldsymbol{c}_i^\top \sum_j \boldsymbol{P}_j \ge 0, && i \in [L], && (31e)\\
& d_i s_j - d_i v_j + \boldsymbol{c}_i^\top \boldsymbol{V}_j - \boldsymbol{c}_i^\top \boldsymbol{P}_j \ge 0, && i \in [L], j \in [n], && (31f)\\
& d_i s_j + d_i v_j - \boldsymbol{c}_i^\top \boldsymbol{V}_j - \boldsymbol{c}_i^\top \boldsymbol{P}_j \ge 0, && i \in [L], j \in [n]. && (31g)
\end{aligned}$$

The dual of Problem (31) is then derived in order to obtain the final RC.

B.3. Adaptive lot sizing

Since $\boldsymbol{z} \ge \boldsymbol{0}$, the uncertainty set can be written as $\mathcal{Z} = \{\boldsymbol{z} \in \mathbb{R}_+^N : \boldsymbol{C}\boldsymbol{z} \le \boldsymbol{d}\}$, where $\boldsymbol{C} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{e}^\top \end{bmatrix}$ and $\boldsymbol{d} = \begin{bmatrix} \rho \boldsymbol{e} \\ \theta \end{bmatrix}$. In addition, we use the notation $\boldsymbol{p} = (\boldsymbol{p}, s)$, $\boldsymbol{x} = (\boldsymbol{x}, \alpha)$ and write the linear constraints $st_{ij} + p_i - p_j \ge 0$ as $\boldsymbol{A}\boldsymbol{p} \le \boldsymbol{0}$. We linearize $\boldsymbol{z}\boldsymbol{z}^\top$ with $\boldsymbol{Z}$, $\boldsymbol{p}\boldsymbol{p}^\top$ with $\boldsymbol{P}$ and $\boldsymbol{z}\boldsymbol{p}^\top$ with $\boldsymbol{W}$ and obtain

$$\begin{aligned}
\max_{\boldsymbol{z},\boldsymbol{p},\boldsymbol{Z},\boldsymbol{P},\boldsymbol{W} \ge \boldsymbol{0}} \quad & \sum_{i=1}^N W_{ii} - \boldsymbol{p}^\top \boldsymbol{x} && && (32a)\\
\text{s.t.} \quad & \boldsymbol{A}\boldsymbol{p} \le \boldsymbol{0}, \quad \boldsymbol{C}\boldsymbol{z} \le \boldsymbol{d}, \quad \boldsymbol{e}^\top \boldsymbol{p} = 1, && && (32b)\\
& \boldsymbol{A}\boldsymbol{w}_j \le \boldsymbol{0}, \quad \boldsymbol{C}\boldsymbol{Z}_j \le z_j \boldsymbol{d}, && j \in [N], && (32c)\\
& \boldsymbol{A}\boldsymbol{P}_j \le \boldsymbol{0}, \quad \boldsymbol{C}\boldsymbol{W}_j \le p_j \boldsymbol{d}, && j \in [N+1], && (32d)\\
& d_i d_j - d_i \boldsymbol{c}_j^\top \boldsymbol{z} - d_j \boldsymbol{c}_i^\top \boldsymbol{z} + \boldsymbol{c}_i^\top \boldsymbol{Z} \boldsymbol{c}_j \ge 0, && i,j \in [L_z], && (32e)\\
& \boldsymbol{a}_i^\top \boldsymbol{P} \boldsymbol{a}_j \ge 0, && i,j \in [L_p], && (32f)\\
& -d_j \boldsymbol{a}_i^\top \boldsymbol{p} + \boldsymbol{c}_j^\top \boldsymbol{W} \boldsymbol{a}_i \ge 0, && i \in [L_p],\ j \in [L_z], && (32g)\\
& \boldsymbol{W}\boldsymbol{e} = \boldsymbol{z}, \quad \boldsymbol{P}\boldsymbol{e} = \boldsymbol{p}. && && (32h)
\end{aligned}$$

The dual of Problem (32) is then derived. We then obtain the RC by replacing the LHS of each constraint with the objective of the derived dual and further adding the dual constraints.

C. Convergence conditions for numerical experiments

C.1. Robust convex geometric optimization

We note that by definition, all sets $\mathcal{X}, \mathcal{Z}, \mathcal{W}$ are bounded. Moreover, in this problem we have $f_i(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{w}) = w_1(\boldsymbol{B}_i^{(1)}\boldsymbol{z} - \boldsymbol{e})^\top \boldsymbol{x} + w_2(\boldsymbol{B}_i^{(2)}\boldsymbol{z} - \boldsymbol{e})^\top \boldsymbol{x} - w_0$. We have

$$|f_i(\boldsymbol{x}_1, \boldsymbol{z}, \boldsymbol{w}) - f_i(\boldsymbol{x}_2, \boldsymbol{z}, \boldsymbol{w})| \le \left| w_1(\boldsymbol{B}_i^{(1)}\boldsymbol{z} - \boldsymbol{e})^\top (\boldsymbol{x}_1 - \boldsymbol{x}_2) \right|$$

$$+\left|w_2(\boldsymbol{B}_i^{(2)}\boldsymbol{z}-\boldsymbol{e})^\top(\boldsymbol{x}_1-\boldsymbol{x}_2)\right| \le (w_1\|\boldsymbol{B}_i^{(1)}\boldsymbol{z}-\boldsymbol{e}\|+w_2\|\boldsymbol{B}_i^{(2)}\boldsymbol{z}-\boldsymbol{e}\|)\|\boldsymbol{x}_1-\boldsymbol{x}_2\|$$
$$\le w_1(\gamma\|\boldsymbol{B}_i^{(1)}\|+\|\boldsymbol{e}\|)\|\boldsymbol{x}_1-\boldsymbol{x}_2\|+w_2(\gamma\|\boldsymbol{B}_i^{(2)}\|+\|\boldsymbol{e}\|)\|\boldsymbol{x}_1-\boldsymbol{x}_2\|.$$

Observe that in this case we have $L=\gamma\max_i\left(\max\left(\|\boldsymbol{B}_i^{(1)}\|,\|\boldsymbol{B}_i^{(2)}\|\right)\right)+\|\boldsymbol{e}\|$.

C.2. Linear optimization with data uncertainty and implementation error

Since the data are generated as $\boldsymbol{a}_i\sim[0,1]$ and $b_i\sim[10,20]$ and $\boldsymbol{x}\ge\boldsymbol{0}$ it follows that $\mathcal{X}$ is bounded. Moreover, by definition, $\mathcal{Z}$ is also bounded. In this problem we have $f_i(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v})=\boldsymbol{a}_i^\top\boldsymbol{x}+\boldsymbol{a}_i^\top\boldsymbol{v}+\boldsymbol{z}^\top\boldsymbol{x}+\boldsymbol{z}^\top\boldsymbol{v}$. We then obtain

$$|f_i(\boldsymbol{x}_1,\boldsymbol{z},\boldsymbol{v})-f_i(\boldsymbol{x}_2,\boldsymbol{z},\boldsymbol{v})|\le\left|\boldsymbol{a}_i^\top(\boldsymbol{x}_1-\boldsymbol{x}_2)\right|+\left|\boldsymbol{z}^\top(\boldsymbol{x}_1-\boldsymbol{x}_2)\right|$$
$$\le(\|\boldsymbol{a}_i\|+\sqrt{n}\|\boldsymbol{z}\|_\infty)\|\boldsymbol{x}_1-\boldsymbol{x}_2\|\le(\|\boldsymbol{a}_i\|+\rho\sqrt{n})\|\boldsymbol{x}_1-\boldsymbol{x}_2\|.$$

Observe that in this case we have $L=\max_i\|\boldsymbol{a}_i\|+\rho\sqrt{n}$.

C.3. Adaptive lot sizing

Let $\tilde{\boldsymbol{x}}=(\boldsymbol{x},\alpha)$ and $\tilde{\boldsymbol{p}}=(\boldsymbol{p},s)$. In this problem we have $f_i(\tilde{\boldsymbol{x}},\boldsymbol{z},\tilde{\boldsymbol{p}})=\tilde{\boldsymbol{p}}^\top(\boldsymbol{z},0)-\tilde{\boldsymbol{p}}^\top\tilde{\boldsymbol{x}}$. We then obtain

$$|f_i(\tilde{\boldsymbol{x}_1},\boldsymbol{z},\tilde{\boldsymbol{p}})-f_i(\tilde{\boldsymbol{x}_2},\boldsymbol{z},\tilde{\boldsymbol{p}})|\le\|\tilde{\boldsymbol{p}}\|_2\|\tilde{\boldsymbol{x}_1}-\tilde{\boldsymbol{x}_2}\|_2\le\|\tilde{\boldsymbol{p}}\|_1\|\tilde{\boldsymbol{x}_1}-\tilde{\boldsymbol{x}_2}\|_2\le\|\tilde{\boldsymbol{x}_1}-\tilde{\boldsymbol{x}_2}\|.$$

Observe that in this case we have $L=1$.