

A Lagrangian Dual Method for Two-Stage Robust Optimization with Binary Uncertainties*

Anirudh Subramanyam

Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL

January 14, 2022

Abstract

This paper presents a new exact method to calculate worst-case parameter realizations in two-stage robust optimization problems with categorical or binary-valued uncertain data. Traditional exact algorithms for these problems, notably Benders decomposition and column-and-constraint generation, compute worst-case parameter realizations by solving mixed-integer bilinear optimization subproblems. However, their numerical solution can be computationally expensive not only due to their resulting large size after reformulating the bilinear terms, but also because decision-independent bounds on their variables are typically unknown. We propose an alternative Lagrangian dual method that circumvents these difficulties and is readily integrated in either algorithm. We specialize the method to problems where the binary parameters switch on or off constraints as these are commonly encountered in applications, and discuss extensions to problems that lack relatively complete recourse and to those with integer recourse. Numerical experiments provide evidence of significant computational improvements over existing methods.

Keywords: robust optimization, two-stage problems, Lagrangian dual, binary uncertainty

1 Motivation

Over the last two decades, robust optimization has emerged as one of the most popular approaches to address optimization problems with uncertain parameters. The key idea is to replace a probabilistic description of the unknown problem data with a deterministic set-based model, and identify solutions that are optimal under the worst-case data realizations from the latter so-called uncertainty set. The approach can not only circumvent some of the computational and modeling difficulties associated with traditional stochastic approaches, but can also offer probabilistic performance guarantees of its solutions via a careful choice of the problem formulation and uncertainty set. It has been successfully applied to problems in a wide variety of domains including energy [18], transportation [48], telecommunication [38], engineering design [12], and finance [21], to name but a few.

The rich gamut of applications have been enabled by several recent developments in robust optimization theory and algorithms. The rich methodology of static or one-stage robust optimization problems, where all decisions have to be taken before any uncertainties are resolved, is reviewed in

*This material is based upon work supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357.

[6, 23]. However, dynamic or multi-stage robust optimization problems, where some subset of the decisions can adapt to observations of unknown problem data, especially those involving categorical or discrete quantities, continue to pose theoretical and algorithmic challenges.

The goal of this paper is to address some of these challenges. Our motivation stems from the broad modeling power of categorical or binary-valued random variables, particularly in applications that are laced with uncertainty about inherently discrete or non-numerical quantities. Examples include:

1. *Optimization over networks or graph structures.* Binary parameters are natural candidates to model random node or link failures in networks. Depending on the application, the latter can represent connections that are either physical, as in the case of electric power [56], transportation [55] and supply chain [41] systems, or virtual, as in the case of telecommunication traffic engineering [40] and industrial control systems [53]. Related examples include adversarial network interdiction [47] and the design of survivable networks [37].
2. *Unit or service availability in manufacturing and service operations.* The unavailability (or failure) of machines, equipment, and units in a production scheduling environment [28, 22], or of particular services in the context of service operations [19] are naturally modeled as binary random variables.
3. *Discrete demand and preference models.* In many applications, customer demands are naturally discrete quantities. For example, logistics operators may not know *a priori* which customers to visit [1, 50]. Another example is the phenomenon of no-shows in queuing or appointment systems with scheduled arrivals, such as in healthcare clinics [35]. Finally, in discrete choice modeling and ranking-based portfolio management, the uncertain parameters naturally belong to discrete sets; for example, in the latter, unknown asset preferences can be modeled as uncertain rankings or permutations over all possible orderings [44].
4. *Cardinality-constrained uncertainty models.* One of the most popular uncertainty sets is the cardinality-constrained or Bertsimas-Sim model [11], which stipulates a budget on the number of uncertain parameters that can simultaneously deviate from their nominal values. This set has a natural interpretation as a discrete uncertainty set. In fact, any continuous-valued unknowns that are modeled using this set can be equivalently reformulated using a binary set. These models are similar to reliability and safety engineering concepts that are referred to as “ $N - k$ reliability” [13] or “ $N + k$ redundancy” [5], where the goal is to design a system with N components such that it remains operational under any combination of k failures.
5. *Scenario-based models of continuous unknowns.* Scenario-based uncertainty sets have been widely used in robust combinatorial optimization [39, 24] as well as other applications such as power systems engineering [33]. As before, such sets can be equivalently reformulated and interpreted as binary sets.
6. *Adversarial machine learning with categorical data.* Relevant applications of robust optimization with categorical uncertainty include logistic regression [46] and binary classification problems with noisy labels [15].

Motivated by these applications, in Section 2, we first describe the problem formulations of interest, then provide an overview of existing algorithms for their solution, and then summarize the contributions of this paper.

2 Problem Formulation and Contributions

We study two different problem formulations. The first one, which we denote by \mathcal{P} , is fairly general where both the constraints and objective function are affected by uncertainty.

$$\begin{aligned} & \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}), \\ \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) = & \left[\begin{array}{l} \text{minimize}_{\mathbf{y} \in \mathcal{Y}} \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} \\ \text{subject to } \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\boldsymbol{\xi}) \end{array} \right]. \end{aligned} \quad (\mathcal{P})$$

Here, \mathbf{x} , \mathbf{y} and $\boldsymbol{\xi}$ are the first- and second-stage decision variables, and the vector of uncertain parameters, respectively. We will assume throughout the paper that the feasible decision sets $\mathcal{X} \subseteq \mathbb{R}^{n_1}$ and $\mathcal{Y} \subseteq \mathbb{R}^{n_2}$, and the uncertainty set $\Xi \subseteq \{0, 1\}^{n_p}$ are non-empty and mixed-integer linear programming (MILP) representable; \mathcal{X} is compact; $\mathbf{c} : \{0, 1\}^{n_p} \rightarrow \mathbb{R}^{n_1}$, $\mathbf{d} : \{0, 1\}^{n_p} \rightarrow \mathbb{R}^{n_2}$ and $\mathbf{h} : \{0, 1\}^{n_p} \rightarrow \mathbb{R}^m$ are vector-valued affine functions; and $\mathbf{T} \in \mathbb{R}^{m \times n_1}$ and $\mathbf{W} \in \mathbb{R}^{m \times n_2}$ are fixed matrices. Finally, we note that $\mathcal{Q} : \mathbb{R}^{n_1} \times \{0, 1\}^{n_p} \rightarrow \mathbb{R} \cup \{-\infty, +\infty\}$ is the second-stage value function, in which we also include the first-stage costs $\mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x}$ for ease of exposition. We adopt the convention that the optimal value of a minimization (maximization) problem is $-\infty$ ($+\infty$) if it is unbounded and $+\infty$ ($-\infty$) if it is infeasible.

The second formulation, which we denote by $\mathcal{P}_{\mathcal{I}}$, uses the uncertain parameters as binary indicators that switch on or off second-stage constraints.

$$\begin{aligned} & \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}), \\ \mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}) = & \left[\begin{array}{l} \text{minimize}_{\mathbf{y} \in \mathcal{Y}} \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} \\ \text{subject to } \mathbf{g}(\mathbf{x}, \mathbf{y}) \geq \mathbf{0} \\ \xi_j = 0 \implies g_i(\mathbf{x}, \mathbf{y}) = 0, \quad i \in \mathcal{I}_j^0, \quad j \in [n_p] \\ \xi_j = 1 \implies g_i(\mathbf{x}, \mathbf{y}) = 0, \quad i \in \mathcal{I}_j^1, \quad j \in [n_p] \end{array} \right]. \end{aligned} \quad (\mathcal{P}_{\mathcal{I}})$$

Here, $\mathbf{g} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^m$ is a vector-valued affine function, $\mathcal{I}_j^0, \mathcal{I}_j^1 \subseteq [m]$ are index sets. and $[n]$ denotes the set $\{1, 2, \dots, n\}$ for any positive integer n .

The indicator formulation $\mathcal{P}_{\mathcal{I}}$ is ubiquitous in applications where the uncertain presence or absence of a quantity can modify the geometry of the feasible set. For example, in the context of network optimization, $g_i(\mathbf{x}, \mathbf{y})$ may represent flow of some commodity i through the j^{th} arc (or node) of the network, which must be set to 0 whenever the arc (or node) becomes unavailable due to random failure, as indicated by $\xi_j = 1$. As we highlighted previously, similar such structures arise, among others, in the context of unit (un-)availability in production scheduling, discrete demand locations in logistics, and customer no-shows in appointment scheduling. Although the indicator formulation $\mathcal{P}_{\mathcal{I}}$ can be viewed as a special case of the more general problem \mathcal{P} after performing an appropriate reformulation (e.g., big-M), we preserve the indicator structure of $\mathcal{P}_{\mathcal{I}}$, since it is more natural and intuitive. Perhaps more importantly, we will show that we can exploit the specific structure of $\mathcal{P}_{\mathcal{I}}$ to obtain computational improvements. Before we proceed, however, we make some important remarks about formulations \mathcal{P} and $\mathcal{P}_{\mathcal{I}}$.

Remark 1. *Problem \mathcal{P} can also accommodate random technology and recourse matrices, $\mathbf{T} : \{0, 1\}^{n_p} \rightarrow \mathbb{R}^{m \times n_1}$ and $\mathbf{W} : \{0, 1\}^{n_p} \rightarrow \mathbb{R}^{m \times n_2}$, whenever they are matrix-valued affine functions and the resulting bilinear terms, $\mathbf{T}(\boldsymbol{\xi})\mathbf{x}$ and $\mathbf{W}(\boldsymbol{\xi})\mathbf{y}$, can be exactly linearized (e.g., using Glover inequalities [26]). The reformulated problem can be readily expressed as an instance of \mathcal{P} .*

Remark 2. *The non-negativity constraint $\mathbf{g}(\mathbf{x}, \mathbf{y}) \geq \mathbf{0}$ in problem $\mathcal{P}_{\mathcal{I}}$ does not restrict generality. Indeed, suppose that for some fixed $i \in [m]$, the quantity $g_i(\mathbf{x}, \mathbf{y})$ is unconstrained, but must be set to 0 whenever $\xi_j = 1$. In such cases, we can augment the second-stage decision vector \mathbf{y} to $(\mathbf{y}, g_i^+, g_i^-)$, replace the indicator constraint in $\mathcal{P}_{\mathcal{I}}$ with $\xi_j = 1 \implies [g_i^+ = g_i^- = 0]$, and impose the additional constraints: $g_i^+, g_i^- \geq 0$ and $g_i(\mathbf{x}, \mathbf{y}) = g_i^+ - g_i^-$.*

The problem \mathcal{P} (or $\mathcal{P}_{\mathcal{I}}$) is said to satisfy the relatively complete recourse assumption whenever the second-stage problem is feasible for all first-stage decisions and uncertain parameter realizations; in other words, $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) < +\infty$ (or $\mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}) < +\infty$) for all $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\xi} \in \Xi$. Similarly, it is said to satisfy the sufficiently expensive recourse assumption whenever the second-stage value function is bounded from below for all first-stage decisions and parameters; in other words, $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) > -\infty$ (or $\mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}) > -\infty$) for all $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\xi} \in \Xi$. We do not need all of these assumptions but state them here for ease of exposition. We list our assumptions in Section 3. Regarding notation, we use \mathbb{R}_+ and \mathbb{Z}_+ to denote the non-negative reals and integers; and \mathbf{e}_j , \mathbf{e} , \mathbf{I} and $\mathbf{0}$ to denote the j^{th} canonical unit vector, the vector of ones, the identity matrix, and the zero vector/matrix, respectively; their dimensions should be clear from the context.

2.1 Literature review

In the presence of binary uncertainties, the two-stage robust optimization problem \mathcal{P} is NP-hard even if there are no first-stage decisions (i.e., \mathcal{X} is a singleton) and the second-stage problem is a two-dimensional linear program with only uncertain objective coefficients (i.e., $\mathcal{Y} = \mathbb{R}_+^2$ and $\mathbf{h}(\boldsymbol{\xi})$ is deterministic) [51]. Indeed, it is solvable in polynomial time only in few special cases, such as when the uncertainty set Ξ has a small inner description (i.e., in terms of a polynomial number of extreme points) or when $\mathbf{h}(\boldsymbol{\xi}) = \mathbf{0}$ and the matrices describing the constraints of Ξ and the slopes of the affine function $\mathbf{d}(\boldsymbol{\xi})$ are totally unimodular [43, 51].

Problem \mathcal{P} simplifies substantially in the absence of binary uncertainties, when the uncertainty set Ξ is linear programming (LP) representable. In this case, existing algorithms for its approximate solution include:

- i)* decision rule approaches [7, 10, 59, 25], which approximate the optimal second-stage decisions \mathbf{y} using parametric classes of linear or nonlinear functions of the uncertain parameters $\boldsymbol{\xi}$;
- ii)* uncertainty set partitioning approaches [45, 9], which partition the uncertainty set Ξ into simple subsets (e.g., hyperrectangles) and approximate \mathbf{y} by affine or constant functions of $\boldsymbol{\xi}$ over each subset;
- iii)* finite- or K -adaptability approaches [8, 31, 14, 49], which calculate a small, fixed number of second-stage decisions $\mathbf{y}_1, \dots, \mathbf{y}_K$ in the first-stage, and subsequently select the best of these in response to a given realization of $\boldsymbol{\xi}$ in the second-stage; and,
- iv)* convexification approaches [30, 2, 54], which first reformulate the infinite-dimensional two-stage problem as a finite-dimensional (but still intractable) one-stage problem and then approximate the latter using lift-and-project techniques (e.g., reformulation-linearization technique or using copositive programming).

Several of these methods, particularly ones that are based on K -adaptability [27, 49] and convexification [43, 34, 51], have also been extended to problems with mixed-integer uncertainties.

In addition to the above approximate schemes, a number of exact methods have also been proposed for problem \mathcal{P} under various simplifying conditions. When uncertainty affects only the

objective function (i.e., $\mathbf{h}(\boldsymbol{\xi})$ is deterministic), the first- and second-stage decisions are binary (i.e., $\mathcal{X} \subseteq \{0,1\}^{n_1}$ and $\mathcal{Y} \subseteq \{0,1\}^{n_2}$) and Ξ is a polytope, [36] have proposed efficient algorithms to calculate lower bounds on the optimal value of \mathcal{P} that is then integrated in a branch-and-bound scheme. A similar approach but allowing also some continuous decisions in the first- and second-stage has been proposed in [3].

When uncertainty affects only the second-stage constraints' right-hand sides (i.e., $\mathbf{d}(\boldsymbol{\xi})$ is deterministic), the second-stage decisions are continuous (i.e., \mathcal{Y} is LP representable) and Ξ is the projection of a binary set (e.g., a cardinality-constrained set), a number of schemes based on Benders decomposition have been developed; e.g., see [52, 4]. A variant of this scheme that can solve problems with an exponential but finite number of uncertain parameter realizations that are described implicitly via block-structured binary uncertainty sets, has been recently developed in [32]. An alternative exact approach based on semi-infinite programming techniques and known as column-and-constraint-generation was proposed in [58, 57]. Notably, this algorithm can also address problems where both the second-stage feasible region \mathcal{Y} and the uncertainty set Ξ are MILP representable.

Before we proceed to discuss the Benders decomposition and column-and-constraint generation algorithms in some detail, we note that alternative approximation schemes for \mathcal{P} and $\mathcal{P}_{\mathcal{I}}$ can be devised by first ignoring the binary restriction on the uncertain parameters, and then using any (exact or approximate) method that is designed to address problems with continuous uncertainties. However, recent studies [51, 43] have shown that such approximations result in unnecessarily conservative (i.e., suboptimal) solutions even when the first- and second-stage feasible regions \mathcal{X} and \mathcal{Y} are convex.

2.1.1 Overview of Benders decomposition algorithm

Algorithm 1 outlines a basic Benders decomposition scheme for solving \mathcal{P} when the second-stage decisions are continuous ($\mathcal{Y} = \mathbb{R}_+^{n_2}$) and the relatively complete and sufficiently expensive recourse assumptions are satisfied. The basic idea is to compute lower bounds LB in line 3 by iteratively refining an inner approximation of $\sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$. Indeed, the lower bounding problem (1) can be derived by noting that strong LP duality implies that we have:

$$\begin{aligned}
& \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) \\
&= \sup_{\boldsymbol{\xi} \in \Xi} \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \inf_{\mathbf{y} \in \mathcal{Y}} \left\{ \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : \mathbf{W}\mathbf{y} \geq \mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}\mathbf{x} \right\} \\
&= \sup_{\boldsymbol{\xi} \in \Xi} \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \sup_{\boldsymbol{\mu} \in \mathbb{R}_+^m} \left\{ \boldsymbol{\mu}^\top (\mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}\mathbf{x}) : \mathbf{W}^\top \boldsymbol{\mu} \leq \mathbf{d}(\boldsymbol{\xi}) \right\} \\
&= \sup_{\boldsymbol{\xi}, \boldsymbol{\mu}} \left\{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \boldsymbol{\mu}^\top (\mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}\mathbf{x}) : \begin{array}{l} \boldsymbol{\xi} \in \Xi, \boldsymbol{\mu} \in \mathbb{R}_+^m \\ \mathbf{W}^\top \boldsymbol{\mu} \leq \mathbf{d}(\boldsymbol{\xi}) \end{array} \right\}, \tag{WC \mathcal{D} }
\end{aligned}$$

where the inner maximization over $\boldsymbol{\mu}$ is feasible for all $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\xi} \in \Xi$ under the relatively complete and sufficiently expensive recourse assumptions. Therefore, Algorithm 1 implicitly enumerates the extreme points of the feasible region of the worst-case problem $\mathcal{WC}_{\mathcal{D}}$ (the subscript \mathcal{D} stands for “duality”) in the set \mathcal{O} . In most implementations, the worst-case parameter realization $\hat{\boldsymbol{\xi}}$ and associated dual values $\hat{\boldsymbol{\mu}}$ are not computed individually as indicated in line 4; rather, they are simultaneously calculated by solving problem $(\mathcal{WC}_{\mathcal{D}})$ directly. The algorithm is guaranteed to terminate after finitely many iterations, since the feasible region of $\mathcal{WC}_{\mathcal{D}}$ has a finite number of extreme points. Finally, we note that the lower bounding problem (1) can be readily reformulated by introducing an epigraphical variable and solved using an MILP solver.

Algorithm 1 Basic Benders decomposition scheme to solve \mathcal{P} when $\mathcal{Y} = \mathbb{R}_+^{n_2}$ and the relatively complete recourse assumption is satisfied

Input: Problem \mathcal{P} , tolerance $\epsilon > 0$

Output: Optimal solution \mathbf{x}^* of \mathcal{P}

- 1: Initialize $LB = -\infty$, $UB = +\infty$, $\mathbf{x}^* = \emptyset$, $\mathcal{O} = \emptyset$.
- 2: **repeat**
- 3: Set LB and $\hat{\mathbf{x}}$ as the optimal objective value and solution of problem (1), respectively:

$$\inf_{\mathbf{x} \in \mathcal{X}} \left\{ \sup_{(\boldsymbol{\xi}, \boldsymbol{\mu}) \in \mathcal{O}} \left\{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \boldsymbol{\mu}^\top (\mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}\mathbf{x}) \right\} \right\} \quad (1)$$

- 4: Compute $\hat{\boldsymbol{\xi}} \in \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\hat{\mathbf{x}}, \boldsymbol{\xi})$
 - 5: Update $\mathcal{O} \leftarrow \mathcal{O} \cup \{(\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\mu}})\}$, where $\hat{\boldsymbol{\mu}} \in \arg \max_{\boldsymbol{\mu} \in \mathbb{R}_+^{n_2}} \left\{ (\mathbf{h}(\hat{\boldsymbol{\xi}}) - \mathbf{T}\hat{\mathbf{x}})^\top \boldsymbol{\mu} : \mathbf{W}^\top \boldsymbol{\mu} \leq \mathbf{d}(\hat{\boldsymbol{\xi}}) \right\}$
 - 6: **if** $UB > \mathcal{Q}(\hat{\mathbf{x}}, \hat{\boldsymbol{\xi}})$ **then** update $UB \leftarrow \mathcal{Q}(\hat{\mathbf{x}}, \hat{\boldsymbol{\xi}})$ and $\mathbf{x}^* \leftarrow \hat{\mathbf{x}}$. **end if**
 - 7: **until** $UB - LB \leq \epsilon$
-

2.1.2 Overview of column-and-constraint generation algorithm

Algorithm 2 describes a basic column-and-constraint generation scheme for solving \mathcal{P} when the relatively complete recourse assumption is satisfied. The key difference with the Benders algorithm is that instead of inner approximating $\sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$ using dual values, the algorithm obtains lower bounds by adding second-stage variables $\mathbf{y}^{(\boldsymbol{\xi})} \in \mathcal{Y}$ and constraints $\mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y}^{(\boldsymbol{\xi})} \geq \mathbf{h}(\boldsymbol{\xi})$ for every worst-case parameter realization $\boldsymbol{\xi}$ recorded in the set \mathcal{R} (see line 3). Notably, this algorithm is agnostic to the presence of mixed-integer recourse decisions, as long as the worst-case parameter realizations $\hat{\boldsymbol{\xi}}$ can be efficiently computed in line 4. In particular, if this is done in finitely many iterations, then Algorithm 2 also terminates finitely since Ξ is a finite set.

Algorithm 2 Basic column-and-constraint generation scheme to solve \mathcal{P} when the relatively complete recourse assumption is satisfied

Input: Problem \mathcal{P} , tolerance $\epsilon > 0$

Output: Optimal solution \mathbf{x}^* of \mathcal{P}

- 1: Initialize $LB = -\infty$, $UB = +\infty$, $\mathbf{x}^* = \emptyset$, $\mathcal{R} = \emptyset$.
- 2: **repeat**
- 3: Set LB and $\hat{\mathbf{x}}$ as the optimal objective value and (projected) solution of problem (2):

$$\inf_{\mathbf{x}, \boldsymbol{\eta}, \mathbf{y}} \left\{ \begin{array}{l} \mathbf{x} \in \mathcal{X}, \quad \boldsymbol{\eta} \in \mathbb{R}, \quad \mathbf{y}^{(\boldsymbol{\xi})} \in \mathcal{Y}, \quad \boldsymbol{\xi} \in \mathcal{R}, \\ \boldsymbol{\eta} : \boldsymbol{\eta} \geq \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y}^{(\boldsymbol{\xi})}, \quad \boldsymbol{\xi} \in \mathcal{R}, \\ \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y}^{(\boldsymbol{\xi})} \geq \mathbf{h}(\boldsymbol{\xi}), \quad \boldsymbol{\xi} \in \mathcal{R}. \end{array} \right\} \quad (2)$$

- 4: Compute $\hat{\boldsymbol{\xi}} \in \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\hat{\mathbf{x}}, \boldsymbol{\xi})$
 - 5: Update $\mathcal{R} \leftarrow \mathcal{R} \cup \{\hat{\boldsymbol{\xi}}\}$.
 - 6: **if** $UB > \mathcal{Q}(\hat{\mathbf{x}}, \hat{\boldsymbol{\xi}})$ **then** update $UB \leftarrow \mathcal{Q}(\hat{\mathbf{x}}, \hat{\boldsymbol{\xi}})$ and $\mathbf{x}^* \leftarrow \hat{\mathbf{x}}$. **end if**
 - 7: **until** $UB - LB \leq \epsilon$
-

When the second-stage decisions are continuous ($\mathcal{Y} = \mathbb{R}_+^{n_2}$), the worst-case parameter realizations in line 4 can be computed by solving the duality-based formulation ($\mathcal{WC}_{\mathcal{D}}$) similar to the Benders scheme. However, the original paper on column-and-constraint generation [58] proposes an alternative formulation using the Karush-Kuhn-Tucker (KKT) conditions of the second-stage problem:

$$\begin{aligned}
& \sup_{\xi \in \Xi} \mathcal{Q}(\mathbf{x}, \xi) \\
&= \sup_{\xi \in \Xi} \mathbf{c}(\xi)^\top \mathbf{x} + \inf_{\mathbf{y} \in \mathcal{Y}} \left\{ \mathbf{d}(\xi)^\top \mathbf{y} : \mathbf{W}\mathbf{y} \geq \mathbf{h}(\xi) - \mathbf{T}\mathbf{x} \right\} \\
&= \sup_{\xi, \mathbf{y}, \boldsymbol{\mu}} \left\{ \begin{array}{l} \xi \in \Xi, \mathbf{y} \in \mathcal{Y}, \boldsymbol{\mu} \in \mathbb{R}_+^m \\ \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\xi) \\ \mathbf{c}(\xi)^\top \mathbf{x} + \mathbf{d}(\xi)^\top \mathbf{y} : \mathbf{W}^\top \boldsymbol{\mu} \leq \mathbf{d}_0 \\ \boldsymbol{\mu} \circ [\mathbf{W}\mathbf{y} + \mathbf{T}\mathbf{x} - \mathbf{h}(\xi)] = \mathbf{0} \\ \mathbf{y} \circ [\mathbf{W}^\top \boldsymbol{\mu} - \mathbf{d}_0] = \mathbf{0} \end{array} \right\}. \tag{WC}_{\mathcal{K}}
\end{aligned}$$

Here, we use $\mathbf{u} \circ \mathbf{v} = (u_1 v_1, \dots, u_n v_n)$ to denote the Hadamard product between two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, and the subscript \mathcal{K} in ($\mathcal{WC}_{\mathcal{K}}$) stands for ‘‘KKT’’. When the second-stage decisions are mixed-integer, [57] have proposed a so-called inner-level column-and-constraint generation algorithm to compute worst-case parameter realizations. However, this algorithm also requires the solution of a problem similar to ($\mathcal{WC}_{\mathcal{D}}$) or ($\mathcal{WC}_{\mathcal{K}}$) for fixed values of the discrete second-stage decisions. We elaborate on this further in Section 4.

2.1.3 Solution of the worst-case parameter realization problems

Both the Benders decomposition and column-and-constraint generation algorithms require the solution of either the duality-based ($\mathcal{WC}_{\mathcal{D}}$) or KKT-based ($\mathcal{WC}_{\mathcal{K}}$) optimization problems to compute worst-case parameter realizations (see line 4 of Algorithms 1 and 2). These are mixed-integer bilinear problems and in practice, they are reformulated as MILP problems using one of two approaches.

In the first approach, they are reformulated using additional variables and indicator constraints. For example, suppose that $\mathbf{h}(\xi) = \mathbf{h}_0 + \mathbf{H}\xi$. Then, by introducing $\boldsymbol{\Omega} \in \mathbb{R}^{m \times n_p}$, the bilinear term appearing in the objective function of ($\mathcal{WC}_{\mathcal{D}}$) can be reformulated as follows:

$$\boldsymbol{\mu}^\top \mathbf{h}(\xi) = \boldsymbol{\mu}^\top \mathbf{h}_0 + \text{tr}(\mathbf{H}^\top \boldsymbol{\Omega})$$

where $\text{tr}(\mathbf{A})$ denotes the trace of a square matrix \mathbf{A} . Additional indicator constraints enforcing the definition $\boldsymbol{\Omega} = \boldsymbol{\mu}\xi^\top$ can be modeled as follows:

$$\begin{aligned}
\xi_j = 0 &\implies \Omega_{ij} = 0, \quad (i, j) \in [m] \times [n_p], \\
\xi_j = 1 &\implies \Omega_{ij} = \mu_i, \quad (i, j) \in [m] \times [n_p].
\end{aligned}$$

The $\mathbf{d}(\xi)^\top \mathbf{y}$ term in the objective function of formulation ($\mathcal{WC}_{\mathcal{K}}$) can be reformulated similarly by adding $n_2 \cdot n_p$ new variables and constraints. The complementary slackness condition $\boldsymbol{\mu} \circ [\mathbf{W}\mathbf{y} + \mathbf{T}\mathbf{x} - \mathbf{h}(\xi)] = \mathbf{0}$ in ($\mathcal{WC}_{\mathcal{K}}$) can be reformulated by introducing $\boldsymbol{\alpha} \in \{0, 1\}^m$ and the indicator constraints:

$$\begin{aligned}
\alpha_i = 0 &\implies \mu_i = 0, & i \in [m], \\
\alpha_i = 1 &\implies \mathbf{e}_i^\top [\mathbf{W}\mathbf{y} + \mathbf{T}\mathbf{x} - \mathbf{h}(\xi)] = 0, & i \in [m].
\end{aligned}$$

The condition $\mathbf{y} \circ [\mathbf{W}^\top \boldsymbol{\mu} - \mathbf{d}(\boldsymbol{\xi})] = \mathbf{0}$ can be reformulated similarly by adding n_2 additional binary variables.

In the second approach, the mixed-integer bilinear problems are reformulated as MILP problems by using the same additional variables as in the indicator approach but the bilinear terms are linearized using big-M constants. For example, we can model $\boldsymbol{\Omega} = \boldsymbol{\mu} \boldsymbol{\xi}^\top$ in (\mathcal{WC}_D) as follows:

$$\max\{\mu_i - M(1 - \xi_j), 0\} \leq \Omega_{ij} \leq \min\{M\xi_j, \mu_i\}, \quad (i, j) \in [m] \times [n_p],$$

where M is a valid upper bound on μ_i (which may not always be available or computable). Similarly, the complementary slackness condition $\boldsymbol{\mu} \circ [\mathbf{W}\mathbf{y} + \mathbf{T}\mathbf{x} - \mathbf{h}(\boldsymbol{\xi})] = \mathbf{0}$ in (\mathcal{WC}_K) can be linearized by adding $2m$ new constraints:

$$\boldsymbol{\mu} \leq M\boldsymbol{\alpha}, \quad \mathbf{W}\mathbf{y} + \mathbf{T}\mathbf{x} - \mathbf{h}(\boldsymbol{\xi}) \leq M(\mathbf{e} - \boldsymbol{\alpha}),$$

and $\mathbf{y} \circ [\mathbf{W}^\top \boldsymbol{\mu} - \mathbf{d}(\boldsymbol{\xi})] = \mathbf{0}$ can be linearized similarly with $2n_2$ constraints.

2.2 Contributions

We present a new alternative method to compute worst-case parameter realizations that can be integrated in either the Benders or column-and-constraint generation algorithms. The method uses a Lagrangian dual reformulation that moves all uncertainties from the constraints to the objective function using a single (i.e., scalar-valued) and finite Lagrange multiplier. The optimal value of this multiplier can be generically and efficiently computed for instances of both \mathcal{P} and \mathcal{P}_I . Notably, our method does not suffer from the computational issues which plague the numerical solution of the aforementioned indicator or linearized MILP reformulations that are used in existing implementations. Specifically, we highlight the following points.

1. The proposed method uses a Lagrangian dual formulation that adds fewer variables and constraints compared to existing methods; see Table 1 for a comparison when the second-stage decisions are continuous and the relatively complete recourse assumption is satisfied. We note, however, that the method can also address the absence of relatively complete recourse, presence of mixed-integer recourse, and uncertain objective functions.

Table 1: Formulation sizes of methods that compute worst-case parameter realizations when $\mathcal{Y} = \mathbb{R}_+^{n_2}$ and the relatively complete recourse assumption is satisfied (excluding decision variables for $\boldsymbol{\xi}$, constraints in Ξ , and variable bounds).

Method	Binary variables	Continuous variables	Constraints
\mathcal{WC}_K (linearized)	$m + n_2$	$m + n_2 + n_2 n_p$	$3m + 3n_2 + 3n_2 n_p$
\mathcal{WC}_K (indicator)	$m + n_2$	$m + n_2 + n_2 n_p$	$3m + 3n_2 + 2n_2 n_p$
\mathcal{WC}_D (linearized)	—	$m + m n_p$	$3m n_p + n_2$
\mathcal{WC}_D (indicator)	—	$m + m n_p$	$2m n_p + n_2$
Proposed (problem \mathcal{P})	—	$m + n_p$	$m + n_2$
Proposed (problem \mathcal{P}_I)	—	m	n_2

2. The optimal Lagrange multiplier can be generically and efficiently computed for any instance of \mathcal{P} and \mathcal{P}_I . This is in contrast to the linearized approach where decision-independent big-M upper bounds on the dual variables $\boldsymbol{\mu}$ are typically unknown. Indeed, the use of arbitrary big-M values here can be problematic: on the one hand, small values can restrict the true feasible

region of $\{\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) : \boldsymbol{\xi} \in \Xi\}$ and hence relax the two-stage problem, leading to suboptimal decisions; on the other hand, large values can cause numerical issues. The indicator approach circumvents these difficulties, but its solution can require higher computational effort.

3. Since our method moves all uncertainty to the objective function, it can potentially enable existing methods [36, 3] that address uncertainty only in the objective to solve also problems with uncertainty-affected constraints.
4. We provide evidence of the computational benefits of our proposed method over existing methods via numerical experiments on network design, facility location, and staff rostering problems that have been previously studied in the literature. We provide open source access to our code.

The rest of this paper is organized as follows. Section 3 presents the Lagrangian dual formulation and the main analytical results that will lay the foundation for Section 4, where we describe our proposed algorithms for computing worst-case parameter realizations and how they can be integrated in the Benders or column-and-constraint generation algorithms. Section 5 discusses numerical experiments and the associated findings. Finally, Section 6 offers concluding remarks and directions for future work.

3 Lagrangian Dual Formulation

Throughout the paper, we make the following assumptions about \mathcal{P} and $\mathcal{P}_{\mathcal{I}}$.

(A1) Problem \mathcal{P} satisfies $\inf \{\mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : \mathbf{y} \in \mathcal{Y}\} > -\infty$ for all $\boldsymbol{\xi} \in \Xi$.

(A2) Problem $\mathcal{P}_{\mathcal{I}}$ satisfies $\inf \{\mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : \mathbf{y} \in \mathcal{Y}, \mathbf{g}(\mathbf{x}, \mathbf{y}) \geq \mathbf{0}\} > -\infty$ for all $\boldsymbol{\xi} \in \Xi$ and $\mathbf{x} \in \mathcal{X}$.

These assumptions essentially imply that both \mathcal{P} and $\mathcal{P}_{\mathcal{I}}$ have sufficiently expensive recourse, and they are always satisfied whenever \mathcal{Y} is compact or when the objective function is bounded from below by some nominal value. They mostly serve to simplify exposition and to eliminate pathological cases that don't typically arise in applications. We emphasize that we do not assume relatively complete recourse for either problem.

3.1 Main results

Our central idea relies on constructing certain Lagrangian and penalty functions of the second-stage problems. For problem \mathcal{P} , we define the Lagrangian function $\mathcal{L} : \mathcal{X} \times \{0, 1\}^{n_p} \times \mathbb{R}_+ \rightarrow \mathbb{R} \cup \{+\infty\}$ and a corresponding penalty function $\phi : [0, 1]^{n_p} \times \{0, 1\}^{n_p} \rightarrow \mathbb{R}$ as follows:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) = \left[\begin{array}{l} \text{minimize}_{\mathbf{y} \in \mathcal{Y}, \mathbf{z} \in \mathbb{R}_+^{n_p}} \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} + \lambda \phi(\mathbf{z}, \boldsymbol{\xi}) \\ \text{subject to } \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\mathbf{z}), \mathbf{z} \leq \mathbf{e}. \end{array} \right] \quad (3)$$

$$\phi(\mathbf{z}, \boldsymbol{\xi}) = \mathbf{e}^\top \mathbf{z} + \mathbf{e}^\top \boldsymbol{\xi} - 2\mathbf{z}^\top \boldsymbol{\xi}. \quad (4)$$

Similarly, for problem $\mathcal{P}_{\mathcal{I}}$, we define the Lagrangian $\mathcal{L}_{\mathcal{I}} : \mathcal{X} \times \{0, 1\}^{n_p} \times \mathbb{R}_+ \rightarrow \mathbb{R} \cup \{+\infty\}$ and its penalty function $\phi_{\mathcal{I}} : \mathcal{X} \times \mathcal{Y} \times \{0, 1\}^{n_p} \rightarrow \mathbb{R}$ as follows:

$$\mathcal{L}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}, \lambda) = \left[\begin{array}{l} \text{minimize}_{\mathbf{y} \in \mathcal{Y}} \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} + \lambda \phi_{\mathcal{I}}(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) \\ \text{subject to } \mathbf{g}(\mathbf{x}, \mathbf{y}) \geq \mathbf{0}. \end{array} \right] \quad (5)$$

$$\phi_{\mathcal{I}}(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) = \sum_{j \in [n_p]} \sum_{i \in \mathcal{I}_j^1} \xi_j g_i(\mathbf{x}, \mathbf{y}) + \sum_{j \in [n_p]} \sum_{i \in \mathcal{I}_j^0} (1 - \xi_j) g_i(\mathbf{x}, \mathbf{y}). \quad (6)$$

Observe that the uncertain parameters $\boldsymbol{\xi}$ appear only in the objective function of \mathcal{L} and $\mathcal{L}_{\mathcal{I}}$ via the penalty functions ϕ and $\phi_{\mathcal{I}}$, respectively. In fact, the latter satisfy some desirable properties that explain why \mathcal{L} and $\mathcal{L}_{\mathcal{I}}$ constitute Lagrangian relaxations of the second-stage problems, \mathcal{Q} and $\mathcal{Q}_{\mathcal{I}}$, respectively.

Lemma 1 (Properties of the penalty functions). *For any $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\xi} \in \Xi$, the penalty functions ϕ and $\phi_{\mathcal{I}}$ satisfy the following properties.*

1. $\phi(\mathbf{z}, \boldsymbol{\xi}) \geq 0$ for all $\mathbf{z} \in [0, 1]^{n_p}$ and $\phi(\mathbf{z}, \boldsymbol{\xi}) = 0$ if and only if $\mathbf{z} = \boldsymbol{\xi}$.
2. If (\mathbf{y}, \mathbf{z}) is feasible in problem (3) with $\phi(\mathbf{z}, \boldsymbol{\xi}) = 0$, then \mathbf{y} is also feasible in problem $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$.
3. If \mathbf{y} is feasible in problem (5), then $\phi_{\mathcal{I}}(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) \geq 0$ and $\phi_{\mathcal{I}}(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) = 0$ if and only if \mathbf{y} is feasible in problem $\mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi})$.

Proof. The first statement follows from the following observation:

$$\phi(\mathbf{z}, \boldsymbol{\xi}) = \mathbf{e}^{\top} \mathbf{z} + \mathbf{e}^{\top} \boldsymbol{\xi} - 2\mathbf{z}^{\top} \boldsymbol{\xi} \geq \mathbf{z}^{\top} \mathbf{z} + \boldsymbol{\xi}^{\top} \boldsymbol{\xi} - 2\mathbf{z}^{\top} \boldsymbol{\xi} = \|\mathbf{z} - \boldsymbol{\xi}\|_2^2 \geq 0,$$

where the inequality follows from $\mathbf{z}, \boldsymbol{\xi} \in [0, 1]^{n_p}$.

The second statement is a consequence of the first, because under the stated conditions, we have $\mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\mathbf{z})$ as well as $\mathbf{z} = \boldsymbol{\xi}$, which taken together imply that $\mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\boldsymbol{\xi})$; that is, \mathbf{y} is feasible in problem $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$.

Finally, the third statement follows directly from the observation that each summand in the expression defining $\phi_{\mathcal{I}}$ is non-negative by construction; therefore, it is equal to 0 if and only if for each $j \in [n_p]$ and $i \in \mathcal{I}_j^1$: either $\xi_j = 0$ or $g_i(\mathbf{x}, \mathbf{y}) = 0$, which is equivalent to the first implication in problem $\mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi})$; and for each $j \in [n_p]$ and $i \in \mathcal{I}_j^0$: either $(1 - \xi_j) = 0$ or $g_i(\mathbf{x}, \mathbf{y}) = 0$, which is equivalent to the second implication in problem $\mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi})$. \square

Lemma 2 (Weak duality). *For any $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\xi} \in \Xi$, we have:*

$$\begin{aligned} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) &\geq \sup_{\lambda \in \mathbb{R}_+} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \\ \mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}) &\geq \sup_{\lambda \in \mathbb{R}_+} \mathcal{L}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \end{aligned}$$

Proof. One can readily verify that if \mathbf{y} is feasible in $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$, then $(\mathbf{y}, \boldsymbol{\xi})$ is feasible in $\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda)$ for any $\lambda \in \mathbb{R}_+$. Moreover, it has the same objective value since $\phi(\boldsymbol{\xi}, \boldsymbol{\xi}) = 0$. Therefore, $\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda)$ is a relaxation of $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$ for all $\lambda \in \mathbb{R}_+$, which proves the first inequality. To prove the second inequality, note that if \mathbf{y} is feasible in $\mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi})$, then it is clearly also feasible in $\mathcal{L}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}, \lambda)$ with the same objective value due to the third part of Lemma 1. \square

Our first main result establishes that the Lagrangian functions \mathcal{L} and $\mathcal{L}_{\mathcal{I}}$ satisfy a strong duality property, even though the second-stage feasible region \mathcal{Y} may be a non-convex MILP representable set.

Theorem 1 (Strong duality). *For any $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\xi} \in \Xi$, we have:*

$$\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) = \sup_{\lambda \in \mathbb{R}_+} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \quad (7)$$

$$\mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}) = \sup_{\lambda \in \mathbb{R}_+} \mathcal{L}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \quad (8)$$

Proof. Denote the finite (possibly empty) set of extreme points of the feasible region of problem (3) by $\mathcal{V} = \text{ext} \{(\mathbf{y}, \mathbf{z}) \in \mathcal{Y} \times [0, 1]^{n_p} : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\mathbf{z})\}$. Under assumption **(A1)**, problem (3) is bounded from below and therefore, it must attain its optimal value at an extreme point:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) = \inf \left\{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} + \lambda \phi(\mathbf{z}, \boldsymbol{\xi}) : (\mathbf{y}, \mathbf{z}) \in \mathcal{V} \right\}, \quad \forall \lambda \in \mathbb{R}_+. \quad (9)$$

First, suppose that $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) = +\infty$. If $\mathcal{V} = \emptyset$, then $\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) = +\infty$ as well and we are done; therefore, assume $\mathcal{V} \neq \emptyset$. Observe that we must have $\phi(\mathbf{z}, \boldsymbol{\xi}) > 0$ for all $(\mathbf{y}, \mathbf{z}) \in \mathcal{V}$; indeed, if $\phi(\mathbf{z}, \boldsymbol{\xi}) = 0$, then the second statement of Lemma 1 implies that \mathbf{y} is feasible in problem $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$, contradicting $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) = +\infty$. Therefore, $\hat{\phi} := \min\{\phi(\mathbf{z}, \boldsymbol{\xi}) : (\mathbf{y}, \mathbf{z}) \in \mathcal{V}\} > 0$ and equation (9) implies:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \geq \min \left\{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : (\mathbf{y}, \mathbf{z}) \in \mathcal{V} \right\} + \lambda \hat{\phi} \quad \forall \lambda \in \mathbb{R}_+. \quad (10)$$

The right-hand side of the above inequality goes to $+\infty$ as $\lambda \rightarrow +\infty$ and therefore, equation (7) is proved in the case when $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) = +\infty$.

Now, suppose that $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) < +\infty$. Then, Lemma 2 implies $\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) < +\infty$ for all $\lambda \in \mathbb{R}_+$. Therefore, equation (9) implies that $\mathcal{V} \neq \emptyset$ and we can define $\hat{\mathcal{V}}(\lambda) = \arg \min_{(\mathbf{y}, \mathbf{z}) \in \mathcal{V}} \{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} + \lambda \phi(\mathbf{z}, \boldsymbol{\xi}) \}$ to be the set of extreme points that achieve the minimum on the right-hand side of (9).

We now claim that there exists some finite $\bar{\lambda} > 0$ such that for all $\lambda \geq \bar{\lambda}$, we have $\phi(\mathbf{z}, \boldsymbol{\xi}) = 0$ for all $(\mathbf{y}, \mathbf{z}) \in \hat{\mathcal{V}}(\lambda)$. Suppose this was not the case; that is, for all $\lambda \in \mathbb{R}_+$, there exists some $(\hat{\mathbf{y}}, \hat{\mathbf{z}}) \in \hat{\mathcal{V}}(\lambda)$ such that $\phi(\hat{\mathbf{z}}, \boldsymbol{\xi}) > 0$. Then, equation (9) implies that

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \geq \min \left\{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : (\mathbf{y}, \mathbf{z}) \in \mathcal{V} \right\} + \lambda \phi(\hat{\mathbf{z}}, \boldsymbol{\xi}) \quad \forall \lambda \in \mathbb{R}_+.$$

The right-hand side of this inequality goes to $+\infty$ as $\lambda \rightarrow +\infty$ and therefore, it implies that $\sup_{\lambda \in \mathbb{R}_+} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) = +\infty$. But this contradicts $\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) < +\infty$. Therefore, we have established that for all $\lambda \geq \bar{\lambda}$, we have $\phi(\mathbf{z}, \boldsymbol{\xi}) = 0$ for all $(\mathbf{y}, \mathbf{z}) \in \hat{\mathcal{V}}(\lambda)$. The second part of Lemma 1 then implies that \mathbf{y} is feasible in problem $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$ for all $(\mathbf{y}, \mathbf{z}) \in \hat{\mathcal{V}}(\lambda)$. Taken together with equation (9), these statements imply that:

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) &= \min \left\{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} + \lambda \phi(\mathbf{z}, \boldsymbol{\xi}) : (\mathbf{y}, \mathbf{z}) \in \mathcal{V} \right\} \quad \forall \lambda \geq \bar{\lambda} \\ &= \min \left\{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : (\mathbf{y}, \mathbf{z}) \in \hat{\mathcal{V}}(\lambda) \right\} \quad \forall \lambda \geq \bar{\lambda} \\ &\geq \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}). \end{aligned}$$

Together with Lemma 2, this proves (7) when $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) < +\infty$ as well.

Equation (8) can be proved using exactly the same arguments, by replacing \mathcal{V} with $\mathcal{V}_{\mathcal{I}}$ and ϕ with $\phi_{\mathcal{I}}$. We omit details in the interest of brevity. \square

Our second main result establishes an even stronger duality property than Theorem 1. Whereas the latter shows that the maximization over λ can be moved outside the inner minimization over the second-stage decisions, the following theorem shows that it can even be moved outside the minimization over the first-stage decisions, without sacrificing optimality. This will turn out to be crucial for ensuring the validity of the lower bounds and overall correctness of the algorithms proposed in Section 4.

Theorem 2 (Strong duality for the two-stage problem).

$$\inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) = \sup_{\lambda \in \mathbb{R}_+} \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \quad (11)$$

$$\inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}) = \sup_{\lambda \in \mathbb{R}_+} \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \quad (12)$$

Proof. Suppose first that \mathcal{P} is infeasible, so that the left-hand side of (11) is equal to $+\infty$. In other words, for all $\mathbf{x} \in \mathcal{X}$, there exists some $\boldsymbol{\xi} \in \Xi$ such that $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) = +\infty$. Using the same argument preceding equation (10) in the proof of Theorem 1, we can define $\mathcal{V}(\mathbf{x}) := \text{ext} \{(\mathbf{y}, \mathbf{z}) \in \mathcal{Y} \times [0, 1]^{n_p} : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\mathbf{z})\}$ and $\hat{\phi}(\mathbf{x}, \boldsymbol{\xi}) := \inf\{\phi(\mathbf{z}, \boldsymbol{\xi}) : (\mathbf{y}, \mathbf{z}) \in \mathcal{V}(\mathbf{x})\} > 0$. Unlike in the proof of Theorem 1, we make the dependence on \mathbf{x} and $\boldsymbol{\xi}$ explicit. Observe that $\tilde{\phi} := \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \hat{\phi}(\mathbf{x}, \boldsymbol{\xi})$ is strictly positive since for all $\boldsymbol{\xi} \in \Xi$, $\hat{\phi}(\mathbf{x}, \boldsymbol{\xi}) > 0$ and finite if $\mathcal{V}(\mathbf{x}) \neq \emptyset$ or $\hat{\phi}(\mathbf{x}, \boldsymbol{\xi}) = +\infty$ otherwise. Now taking sup over $\boldsymbol{\xi}$ and then inf over \mathbf{x} in equation (10), we have for all $\lambda \in \mathbb{R}_+$ that:

$$\inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \geq \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \inf \left\{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : (\mathbf{y}, \mathbf{z}) \in \mathcal{V}(\mathbf{x}) \right\} + \lambda \tilde{\phi}.$$

The first term in the right-hand side of the above inequality is equal to $+\infty$ if $\mathcal{V}(\mathbf{x}) = \emptyset$ for all $\mathbf{x} \in \mathcal{X}$; otherwise, it is finite, since \mathcal{X} and Ξ are compact. In either case, the right-hand side goes to $+\infty$ as $\lambda \rightarrow \infty$.

Suppose now that \mathcal{P} is feasible, so that the left-hand side of (11) is finite. It suffices to take the minimization over decisions $\mathbf{x} \in \mathcal{X}$ for which $\sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) < +\infty$. In this case, Lemma 2 implies that $\sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) < +\infty$ for all $\lambda \geq 0$, whereas the proof of Theorem 1 establishes that there exists some finite $\bar{\lambda}(\mathbf{x}, \boldsymbol{\xi}) > 0$ (we again make the dependence on \mathbf{x} and $\boldsymbol{\xi}$ explicit here) such that for all feasible $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\xi} \in \Xi$, the function $\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \cdot)$ is maximized by any $\lambda \geq \bar{\lambda}(\mathbf{x}, \boldsymbol{\xi})$. By defining $\tilde{\lambda} := \max\{\bar{\lambda}(\mathbf{x}, \boldsymbol{\xi}) : \mathbf{x} \in \mathcal{X}, \boldsymbol{\xi} \in \Xi\}$ which is finite since \mathcal{X} and Ξ are compact, we then note that $\tilde{\lambda}$ maximizes the function $\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \cdot)$ for all \mathbf{x} and $\boldsymbol{\xi}$. In other words, we have:

$$\inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \sup_{\lambda \in \mathbb{R}_+} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) = \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \tilde{\lambda}) \leq \sup_{\lambda \in \mathbb{R}_+} \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda),$$

where the inequality follows by treating the middle expression as a function that is evaluated at $\lambda = \tilde{\lambda}$ and the right-most expression as maximizing this function over all $\lambda \in \mathbb{R}_+$. The max-min inequality, however, implies that:

$$\inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \sup_{\lambda \in \mathbb{R}_+} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) = \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\lambda \in \mathbb{R}_+} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \geq \sup_{\lambda \in \mathbb{R}_+} \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda).$$

Combining the previous two inequalities with Theorem 1, we obtain:

$$\inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) = \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \sup_{\lambda \in \mathbb{R}_+} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) = \sup_{\lambda \in \mathbb{R}_+} \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda),$$

which proves equation (11).

As in the proof of Theorem 1, equation (8) can be proved using the exact same arguments, and we again omit details for the sake of brevity. \square

3.2 Optimal Lagrange multipliers

Theorems 1 and 2 reduce the computation of a worst-case parameter realization from optimizing over the second-stage value function to optimizing over the Lagrangian dual function. We now characterize the structure of an optimal Lagrange multiplier for \mathcal{P} and $\mathcal{P}_{\mathcal{I}}$. Before that, however, we elucidate a useful property of the worst-case Lagrangian functions.

Lemma 3 (Properties of the worst-case Lagrangian functions). *For any $\mathbf{x} \in \mathcal{X}$, the functions $\sup\{\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \cdot) : \boldsymbol{\xi} \in \Xi\}$ and $\sup\{\mathcal{L}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}, \cdot) : \boldsymbol{\xi} \in \Xi\}$ are piecewise linear, continuous, and monotonically non-decreasing in $\lambda \in \mathbb{R}_+$.*

Proof. Observe that (i) the objective function of $\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda)$ is affine in λ ; (ii) its optimal solution is always attained at any of the finite number of extreme points of its feasible region in problem (3); and therefore, (iii) the function $\sup\{\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \cdot) : \boldsymbol{\xi} \in \Xi\}$ is the supremum over a finite set ($\boldsymbol{\xi} \in \Xi$) of the infimum of a finite number of affine functions. This shows that it is piecewise linear and continuous.

Its non-decreasing nature follows directly from the non-negativity of the slopes of the linear pieces; that is, of the penalty functions ϕ and $\phi_{\mathcal{I}}$ that was established in Lemma 1. Specifically, consider any $\lambda_1, \lambda_2 \in \mathbb{R}$ such that $\lambda_1 \leq \lambda_2$. Then, for any $\mathbf{y} \in \mathcal{Y}$, $\mathbf{z} \in [0, 1]^{n_p}$ and $\boldsymbol{\xi} \in \Xi$, we have $\mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} + \lambda_1 \phi(\mathbf{z}, \boldsymbol{\xi}) \leq \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} + \lambda_2 \phi(\mathbf{z}, \boldsymbol{\xi})$. Therefore, $\sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda_1) \leq \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda_2)$ implying that it is non-decreasing in λ . The argument for $\sup\{\mathcal{L}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}, \cdot) : \boldsymbol{\xi} \in \Xi\}$ is similar. \square

A necessary condition for the optimality of a Lagrange multiplier directly falls out of Lemma 3, which we present next.

Theorem 3 (Necessary conditions for optimality of multipliers). *1. For any $\mathbf{x} \in \mathcal{X}$ such that $Q(\mathbf{x}, \boldsymbol{\xi})$ is finite for all $\boldsymbol{\xi} \in \Xi$, we have that*

$$\bar{\lambda} \in \arg \max_{\lambda \in \mathbb{R}_+} \left\{ \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \right\}$$

only if there exists $\bar{\boldsymbol{\xi}} \in \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \bar{\lambda})$ and an optimal solution $(\bar{\mathbf{y}}, \bar{\mathbf{z}})$ of $\mathcal{L}(\mathbf{x}, \bar{\boldsymbol{\xi}}, \bar{\lambda})$ such that $\phi(\bar{\mathbf{z}}, \bar{\boldsymbol{\xi}}) = 0$.

2. For any $\mathbf{x} \in \mathcal{X}$ such that $Q_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi})$ is finite for all $\boldsymbol{\xi} \in \Xi$, we have that

$$\bar{\lambda} \in \arg \max_{\lambda \in \mathbb{R}_+} \left\{ \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \right\}$$

only if there exists $\bar{\boldsymbol{\xi}} \in \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{L}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}, \bar{\lambda})$ and an optimal solution $\bar{\mathbf{y}}$ of $\mathcal{L}_{\mathcal{I}}(\mathbf{x}, \bar{\boldsymbol{\xi}}, \bar{\lambda})$ such that $\phi_{\mathcal{I}}(\mathbf{x}, \bar{\mathbf{y}}, \bar{\boldsymbol{\xi}}) = 0$.

Proof. Suppose that the stated conditions hold and $\bar{\lambda}$ is an optimal multiplier. Theorem 1 implies that $\sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \bar{\lambda})$ must be finite. Therefore, denote $\hat{\mathcal{U}} := \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \bar{\lambda})$ and for each $\boldsymbol{\xi} \in \hat{\mathcal{U}}$, denote the set of minimizers of problem $\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \bar{\lambda})$ as $\hat{\mathcal{V}}(\boldsymbol{\xi})$. Suppose for the sake of contradiction, that for all $\boldsymbol{\xi} \in \hat{\mathcal{U}}$ and for all $(\mathbf{y}, \mathbf{z}) \in \hat{\mathcal{V}}(\boldsymbol{\xi})$ we have that $\phi(\mathbf{z}, \boldsymbol{\xi}) > 0$. Define $\hat{\phi} = \min\{\phi(\mathbf{z}, \boldsymbol{\xi}) : (\mathbf{y}, \mathbf{z}) \in \hat{\mathcal{V}}(\boldsymbol{\xi}), \boldsymbol{\xi} \in \hat{\mathcal{U}}\} > 0$. Then, by construction:

$$\sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \bar{\lambda}) = \max_{\boldsymbol{\xi} \in \hat{\mathcal{U}}} \min_{(\mathbf{y}, \mathbf{z}) \in \hat{\mathcal{V}}(\boldsymbol{\xi})} \{\mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} + \bar{\lambda} \phi(\mathbf{z}, \boldsymbol{\xi})\}$$

Continuity of the function on the left-hand side (with respect to λ) implies that for some sufficiently small $\epsilon > 0$, $\max_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \bar{\lambda} + \epsilon) \geq \max_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \bar{\lambda}) + \epsilon \hat{\phi}$, contradicting the optimality of $\bar{\lambda}$. The proof of the second statement is similar and omitted for the sake of brevity. \square

Although Theorem 3 is only a necessary condition for optimality, we shall show in the next section that we can still use it in conjunction with Theorem 2 to design an algorithm for computing the optimal multiplier. For now, we close this section by providing a sufficient condition under which we can compute a closed-form expression for the optimal Lagrange multiplier in problem \mathcal{P} .

Theorem 4 (Closed-form expression for optimal Lagrange multiplier). *Suppose that for every $\mathbf{x} \in \mathcal{X}$, either there exists $\boldsymbol{\xi} \in \Xi$ such that $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) = +\infty$ or $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) < +\infty$ for all $\boldsymbol{\xi} \in \{0, 1\}^{n_p}$. Then, for any feasible first-stage decision $\mathbf{x} \in \mathcal{X}$; that is, for which $\sup \{\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) : \boldsymbol{\xi} \in \Xi\} < +\infty$, we have that*

$$u(\mathbf{x}) - \ell(\mathbf{x}) \in \arg \max_{\lambda \in \mathbb{R}_+} \left\{ \max_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \right\},$$

where $u(\mathbf{x})$ is any finite upper bound on $\sup \{\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) : \boldsymbol{\xi} \in \Xi\}$ and $\ell(\mathbf{x})$ is any finite lower bound on $\inf \{\mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : \boldsymbol{\xi} \in \Xi, \mathbf{y} \in \mathcal{Y}\}$.

Proof. Suppose that $\mathbf{x} \in \mathcal{X}$ is any feasible first-stage decision in \mathcal{P} . Then, we claim that

$$\text{Proj}_{\mathbf{z}} (\{(\mathbf{y}, \mathbf{z}) \in \mathcal{Y} \times [0, 1]^{n_p} : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\mathbf{z})\}) = [0, 1]^{n_p}.$$

The ‘ \subseteq ’ inclusion is obvious. To show the reverse ‘ \supseteq ’ inclusion, we note that the condition stated in the Theorem implies that since \mathbf{x} is feasible and has a finite second-stage value $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) < +\infty$ for all $\boldsymbol{\xi} \in \Xi$, it also has a finite second-stage value for all $\boldsymbol{\xi} \in \{0, 1\}^{n_p}$. In other words, the set $\{\mathbf{y} \in \mathcal{Y} : \mathbf{W}\mathbf{y} \geq \mathbf{h}(\mathbf{z}) - \mathbf{T}\mathbf{x}\}$ is non-empty for any setting of $\mathbf{z} = \boldsymbol{\xi} \in \{0, 1\}^{n_p}$. This proves the claim.

Therefore, problem (3) always has an optimal solution $(\hat{\mathbf{y}}, \hat{\mathbf{z}})$ such that $\hat{\mathbf{z}} \in \{0, 1\}^{n_p}$, for any $\boldsymbol{\xi} \in \Xi$ and $\lambda \in \mathbb{R}_+$ (since its objective is linear in \mathbf{z}). In this case, the definition of $\phi(\hat{\mathbf{z}}, \boldsymbol{\xi})$ is equivalent to that of the Hamming distance between two binary vectors and therefore, we have $\phi(\hat{\mathbf{z}}, \boldsymbol{\xi}) \in \{0, 1, 2, \dots, n_p\}$.

We are now ready to prove the statement of the theorem. In particular, we aim to show that $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) = \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda)$ is satisfied for all $\lambda \geq u(\mathbf{x}) - \ell(\mathbf{x})$ and all $\boldsymbol{\xi} \in \Xi$. Note that along with Theorem 1, this will prove the first part of this theorem. Suppose for the sake of contradiction that $\mathcal{Q}(\mathbf{x}, \hat{\boldsymbol{\xi}}) > \mathcal{L}(\mathbf{x}, \hat{\boldsymbol{\xi}}, \hat{\lambda})$ for some $\hat{\lambda} \geq u(\mathbf{x}) - \ell(\mathbf{x})$ and some $\hat{\boldsymbol{\xi}} \in \Xi$. Let $(\hat{\mathbf{y}}, \hat{\mathbf{z}})$ be an optimal solution of $\mathcal{L}(\mathbf{x}, \hat{\boldsymbol{\xi}}, \hat{\lambda})$ (i.e., problem (3)). Then, the second part of Lemma 1 implies that we must have $\phi(\hat{\mathbf{z}}, \hat{\boldsymbol{\xi}}) > 0$ (otherwise, $\hat{\mathbf{y}}$ would be feasible in $\mathcal{Q}(\mathbf{x}, \hat{\boldsymbol{\xi}})$ with the same objective value $\mathcal{L}(\mathbf{x}, \hat{\boldsymbol{\xi}}, \hat{\lambda})$ which contradicts the hypothesis). But since we have established in the previous paragraph that $\phi(\hat{\mathbf{z}}, \boldsymbol{\xi}) \in \{0, 1, 2, \dots, n_p\}$, this implies that we must have $\phi(\hat{\mathbf{z}}, \boldsymbol{\xi}) \geq 1$. Along with the definition of $\ell(\mathbf{x})$, this further implies the following (where we note that a finite value for $\ell(\mathbf{x})$ is guaranteed under assumption **(A1)**):

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \hat{\boldsymbol{\xi}}, \hat{\lambda}) &= \mathbf{c}(\hat{\boldsymbol{\xi}})^\top \mathbf{x} + \mathbf{d}(\hat{\boldsymbol{\xi}})^\top \hat{\mathbf{y}} + \hat{\lambda} \phi(\hat{\mathbf{z}}, \hat{\boldsymbol{\xi}}) \\ &\geq \inf_{\boldsymbol{\xi} \in \Xi, \mathbf{y} \in \mathcal{Y}} \{\mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y}\} + \hat{\lambda} \geq \ell(\mathbf{x}) + \hat{\lambda}. \end{aligned}$$

But from the hypothesis, and from the definition of $u(\mathbf{x})$, we have:

$$\mathcal{L}(\mathbf{x}, \hat{\boldsymbol{\xi}}, \hat{\lambda}) < \mathcal{Q}(\mathbf{x}, \hat{\boldsymbol{\xi}}) \leq \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) \leq u(\mathbf{x}).$$

The last two inequalities together imply that $\hat{\lambda} < u(\mathbf{x}) - \ell(\mathbf{x})$ which is the desired contradiction. \square

Theorem 4 provides a closed-form expression for the optimal Lagrange multiplier of $\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda)$ in terms of any known upper bound $u(\mathbf{x})$ on the worst-case objective value of \mathbf{x} and any known lower bound $\ell(\mathbf{x})$ on the best-case objective value of \mathbf{x} . The next theorem provides some sufficient conditions under which these quantities can be efficiently computed.

Theorem 5 (Closed-form expressions for $\ell(\mathbf{x})$ and $u(\mathbf{x})$). *Suppose that the conditions of Theorem 4 hold and that $\mathbf{x} \in \mathcal{X}$ is any feasible first-stage decision in \mathcal{P} . Then, the quantities $\ell(\mathbf{x})$ and $u(\mathbf{x})$ admit closed-form expressions under any of the following conditions.*

1. If $\mathcal{Y} \subseteq [y_1^\ell, y_1^u] \times [y_2, \bar{y}_2] \times \dots \times [y_{n_2}^\ell, y_{n_2}^u]$ is compact, or more generally, if for all $\boldsymbol{\xi} \in \Xi$, an optimal solution $\hat{\mathbf{y}}$ of the problem $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$ is known to satisfy $\hat{\mathbf{y}} \in [\mathbf{y}^\ell, \mathbf{y}^u]$ for some $\mathbf{y}^\ell, \mathbf{y}^u \in \mathbb{R}^{n_2}$, then

$$\begin{aligned} \ell(\mathbf{x}) &= \inf_{\boldsymbol{\xi}, \mathbf{y}} \{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : \boldsymbol{\xi} \in \{0, 1\}^{n_p}, y_i \in \{y_i^\ell, y_i^u\}, i \in [n_2] \}, \\ u(\mathbf{x}) &= \sup_{\boldsymbol{\xi}, \mathbf{y}} \{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : \boldsymbol{\xi} \in \{0, 1\}^{n_p}, y_i \in \{y_i^\ell, y_i^u\}, i \in [n_2] \}. \end{aligned}$$

2. If the objective function $\mathbf{d}(\boldsymbol{\xi}) = \mathbf{d}_0 + \mathbf{D}\boldsymbol{\xi}$ satisfies $\mathbf{D}^\top \mathbf{y} \geq \mathbf{0}$ for all $\{\mathbf{y} \in \mathcal{Y} : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\boldsymbol{\xi})\}$ and all $\boldsymbol{\xi} \in \Xi$, then

$$\ell(\mathbf{x}) = \inf_{\boldsymbol{\xi} \in [0, 1]^{n_p}} \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \inf_{\mathbf{y} \in \mathcal{Y}} \mathbf{d}_0^\top \mathbf{y},$$

If also $\mathbf{h}(\boldsymbol{\xi}) = \mathbf{h}_0 + \mathbf{H}\boldsymbol{\xi}$ satisfies $\mathbf{H}^\top \mathbf{y} \geq \mathbf{0}$ for all $\{\mathbf{y} \in \mathcal{Y} : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\boldsymbol{\xi})\}$ and all $\boldsymbol{\xi} \in \Xi$, then

$$u(\mathbf{x}) = \sup_{\boldsymbol{\xi} \in \{0, 1\}^{n_p}} \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\mathbf{e})^\top \mathbf{y}^0,$$

for any arbitrary $\mathbf{y}^0 \in \{\mathbf{y} \in \mathcal{Y} : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\mathbf{e})\}$.

3. If \mathbf{x} is static robust feasible; that is, the set $\mathcal{Y}(\mathbf{x}) := \{\mathbf{y} \in \mathcal{Y} : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \sup_{\boldsymbol{\xi} \in \Xi} \mathbf{h}(\boldsymbol{\xi})\} \neq \emptyset$, then

$$u(\mathbf{x}) = \sup_{\boldsymbol{\xi} \in \{0, 1\}^{n_p}} \{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y}^0 \},$$

for any arbitrary $\mathbf{y}^0 \in \mathcal{Y}(\mathbf{x})$.

Proof. The first statement follows by observing first the following:

$$\begin{aligned} \inf_{\boldsymbol{\xi} \in \Xi} \inf_{\mathbf{y} \in \mathcal{Y}} \{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} \} &\geq \inf_{\boldsymbol{\xi} \in [0, 1]^{n_p}} \inf_{\mathbf{y} \in [y^\ell, y^u]} \{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} \}, \\ \text{and } \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) &= \sup_{\boldsymbol{\xi} \in \Xi} \inf_{\mathbf{y} \in \mathcal{Y}} \{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\boldsymbol{\xi}) \} \\ &\leq \sup_{\boldsymbol{\xi} \in \{0, 1\}^{n_p}} \inf_{\mathbf{y} \in \mathcal{Y}} \{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\boldsymbol{\xi}) \} \\ &\leq \sup_{\boldsymbol{\xi} \in \{0, 1\}^{n_p}} \sup_{\mathbf{y} \in [y^\ell, y^u]} \{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} \}, \end{aligned}$$

and then by noting that the optimal value of a disjoint bilinear program over polytopes is attained at extreme points of the individual polytopes.

The expression for $\ell(\mathbf{x})$ in the second statement follows by noting that because $\boldsymbol{\xi} \geq \mathbf{0}$, we have $\mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} = \mathbf{d}_0^\top \mathbf{y} + \boldsymbol{\xi}^\top \mathbf{D}^\top \mathbf{y} \geq \mathbf{d}_0^\top \mathbf{y}$ for all $\mathbf{y} \in \mathcal{Y}$. Similarly, the expression for $u(\mathbf{x})$ can be obtained by noting that because $\boldsymbol{\xi} \leq \mathbf{e}$, we have $\mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} = \mathbf{d}_0^\top \mathbf{y} + \boldsymbol{\xi}^\top \mathbf{D}^\top \mathbf{y} \leq \mathbf{d}_0^\top \mathbf{y} + \mathbf{e}^\top \mathbf{D}^\top \mathbf{y} = \mathbf{d}(\mathbf{e})^\top \mathbf{y}$ for all $\mathbf{y} \in \mathcal{Y}$. Moreover, any $\mathbf{y} \in \mathcal{Y}$ that is feasible in problem $\mathcal{Q}(\mathbf{x}, \mathbf{e})$ is also feasible in $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$ for any $\boldsymbol{\xi} \in \Xi$, since $\mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\mathbf{e}) \geq \mathbf{h}(\boldsymbol{\xi})$. Therefore, we have:

$$\begin{aligned} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) &= \sup_{\boldsymbol{\xi} \in \Xi} \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \sup_{\boldsymbol{\xi} \in \Xi} \inf_{\mathbf{y} \in \mathcal{Y}} \{ \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\boldsymbol{\xi}) \} \\ &= \sup_{\boldsymbol{\xi} \in \Xi} \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \inf_{\mathbf{y} \in \mathcal{Y}} \{ \mathbf{d}(\mathbf{e})^\top \mathbf{y} : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\mathbf{e}) \} \end{aligned}$$

$$\leq \sup_{\boldsymbol{\xi} \in \{0,1\}^{n_p}} \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\mathbf{e})^\top \mathbf{y}^0.$$

As per the condition of Theorem 4, since \mathbf{x} satisfies $\sup \{Q(\mathbf{x}, \boldsymbol{\xi}) : \boldsymbol{\xi} \in \Xi\} < +\infty$, we also must have that the feasible region of problem $Q(\mathbf{x}, \mathbf{e})$ is non-empty; that is, \mathbf{y}^0 is guaranteed to exist.

The last statement follows from the max-min inequality:

$$\begin{aligned} \sup_{\boldsymbol{\xi} \in \Xi} Q(\mathbf{x}, \boldsymbol{\xi}) &\leq \inf_{\mathbf{y} \in \mathcal{Y}} \left\{ \sup_{\boldsymbol{\xi} \in \Xi} \left\{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y} \right\} : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \max_{\boldsymbol{\xi} \in \Xi} \mathbf{h}(\boldsymbol{\xi}) \right\} \\ &\leq \sup_{\boldsymbol{\xi} \in \{0,1\}^{n_p}} \left\{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}(\boldsymbol{\xi})^\top \mathbf{y}^0 \right\}, \end{aligned}$$

where $\mathbf{y}^0 \in \mathcal{Y}(\mathbf{x})$. □

We close with some remarks about Theorems 4 and 5. First, the condition under which Theorem 4 is established is neither weaker nor stronger than relatively complete recourse. It is equivalent to stating that if a first-stage decision is feasible for all $\boldsymbol{\xi} \in \Xi$, then it is also feasible for all $\boldsymbol{\xi} \in \{0, 1\}^{n_p}$. This condition is satisfied in many applications, including those where slack variables are explicitly added to ensure problem feasibility. Second, all of the conditions stated in Theorem 5 are easily verifiable and largely problem-independent. Nevertheless, although the closed-form expressions in Theorem 5 remain valid under its stated conditions, it is important to choose small values of $u(\mathbf{x})$ and large values of $\ell(\mathbf{x})$ as this will lead to smaller values of the optimal Lagrange multiplier. This will be important for numerical reasons that we shall elaborate in the next section.

4 Computation of Worst-Case Parameter Realizations

We now use the analytical results from the previous section to design practical methods for computing worst-case parameter realizations that can be integrated in the Benders and column-and-constraint generation algorithms even when the relatively complete recourse assumption may not be satisfied. We also discuss extensions to mixed-integer second-stage decisions.

Theorems 3 and 4 characterize the optimal Lagrange multiplier only in the context of feasible first-stage decisions. To reliably identify a potentially infeasible first-stage decision $\mathbf{x} \in \mathcal{X}$ for which the second-stage problem becomes infeasible for some $\boldsymbol{\xi} \in \Xi$, we introduce the constraint violation functions $\mathcal{S} : \mathcal{X} \times \Xi \rightarrow \mathbb{R}$ and $\mathcal{S}_{\mathcal{I}} : \mathcal{X} \times \Xi \rightarrow \mathbb{R}$ which minimize constraint infeasibility in the problems \mathcal{L} and $\mathcal{L}_{\mathcal{I}}$, respectively.

$$\mathcal{S}(\mathbf{x}, \boldsymbol{\xi}) = \left[\begin{array}{l} \underset{\mathbf{y}, \mathbf{z}, \boldsymbol{\sigma}}{\text{minimize}} \quad \mathbf{e}^\top \boldsymbol{\sigma} + \phi(\mathbf{z}, \boldsymbol{\xi}) \\ \text{subject to} \quad \mathbf{y} \in \mathcal{Y}, \quad \mathbf{z} \in [0, 1]^{n_p}, \quad \boldsymbol{\sigma} \in \mathbb{R}_+^m \\ \quad \quad \quad \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} + \boldsymbol{\sigma} \geq \mathbf{h}(\mathbf{z}). \end{array} \right] \quad (13)$$

$$\mathcal{S}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}) = \left[\begin{array}{l} \underset{\mathbf{y}, \boldsymbol{\sigma}}{\text{minimize}} \quad \mathbf{e}^\top \boldsymbol{\sigma} + \phi_{\mathcal{I}}(\mathbf{x}, \mathbf{y}, \boldsymbol{\xi}) \\ \text{subject to} \quad \mathbf{y} \in \mathcal{Y}, \quad \boldsymbol{\sigma} \in \mathbb{R}_+^m \\ \quad \quad \quad \mathbf{g}(\mathbf{x}, \mathbf{y}) + \boldsymbol{\sigma} \geq \mathbf{0}. \end{array} \right] \quad (14)$$

We immediately note the following properties of these functions.

Lemma 4 (Properties of the constraint violation functions). *For any $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\xi} \in \Xi$, the functions \mathcal{S} and $\mathcal{S}_{\mathcal{I}}$ satisfy the following properties.*

1. $\mathcal{S}(\mathbf{x}, \boldsymbol{\xi}) \geq 0$ and $\mathcal{S}(\mathbf{x}, \boldsymbol{\xi}) = 0$ if and only if $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) < +\infty$.
2. $\mathcal{S}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}) \geq 0$ and $\mathcal{S}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}) = 0$ if and only if $\mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}) < +\infty$.

Proof. First, since $\mathcal{Y} \neq \emptyset$, both minimization problems (13) and (14) are always feasible. Also, since the penalty functions ϕ and $\phi_{\mathcal{I}}$ are always non-negative, the objective functions of both problems are non-negative by construction. Finally, note that if $\mathcal{S}(\mathbf{x}, \boldsymbol{\xi}) = 0$, then any optimal solution $(\mathbf{y}, \mathbf{z}, \boldsymbol{\sigma})$ satisfies $\boldsymbol{\sigma} = \mathbf{0}$, implying that (\mathbf{y}, \mathbf{z}) is feasible in the second-stage problem $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$. Similarly, for any feasible solution of the latter problem, we can construct a feasible solution in $\mathcal{S}(\mathbf{x}, \boldsymbol{\xi})$ with objective value 0, by setting $\boldsymbol{\sigma} = \mathbf{0}$. Similar arguments can be made for $\mathcal{S}_{\mathcal{I}}$. \square

Using the constraint violation functions, we can compute worst-case parameter realizations even when the relatively complete recourse assumption may not be satisfied. Algorithm 3 describes the computations for \mathcal{P} when the sufficient conditions of Theorem 5 are satisfied, whereas Algorithms 4 and 5 describe the computations for general instances of \mathcal{P} and $\mathcal{P}_{\mathcal{I}}$, respectively. All algorithms first compute the worst-case constraint violations. If this is equal to 0 (with actual implementations using a small non-zero tolerance), then Lemma 4 ensures that $\mathbf{x} \in \mathcal{X}$ is a feasible first-stage decision. The algorithms then proceed to compute worst-case parameter realizations over the Lagrangian dual functions. We note that the computation of the worst-case constraint violations can be skipped if the problem is known to satisfy the relatively complete recourse assumption.

Algorithm 3 applies Theorem 4 to compute a closed-form expression for the optimal Lagrange multiplier when \mathbf{x} is feasible. Indeed, in this case, the conditions of Theorem 4 are satisfied and $\hat{\lambda}$ in line 3 is an optimal multiplier. The conditions in Theorem 5 and the corresponding closed-form expressions that are used for computing the optimal multiplier can all be verified and computed efficiently, and even analytically in several applications. We note, however, that whenever possible, problem-specific knowledge that can yield smaller values of $u(\mathbf{x})$ (i.e., tighter upper bounds on the worst-case objective value of \mathbf{x}) and larger values of $\ell(\mathbf{x})$ (i.e., tighter lower bounds on the best-case objective value of \mathbf{x}) should be used, since it will lead to smaller values of $\hat{\lambda}$ and a computationally less expensive maximization (over $\boldsymbol{\xi} \in \Xi$) of the Lagrangian function evaluated at $\hat{\lambda}$ in line 4. Note that if \mathbf{x} is infeasible, then any positive value of $\hat{\lambda}$ is an optimal multiplier and line 1 arbitrarily sets $\hat{\lambda} = 1$.

Algorithm 3 Computation of worst-case parameter realization for \mathcal{P} when the conditions in Theorem 5 are satisfied

Input: $\mathbf{x} \in \mathcal{X}$

Output: $\hat{\boldsymbol{\xi}} \in \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$ and $\hat{\lambda} \in \arg \max_{\lambda \in \mathbb{R}_+} \left\{ \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \right\}$

1: Compute $\hat{\boldsymbol{\xi}} \in \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{S}(\mathbf{x}, \boldsymbol{\xi})$ and set $\hat{\lambda} = 1$

2: **if** $\mathcal{S}(\mathbf{x}, \hat{\boldsymbol{\xi}}) = 0$ **then**

3: Compute $\ell(\mathbf{x})$ and $u(\mathbf{x})$ as per Theorem 4 and set $\hat{\lambda} = u(\mathbf{x}) - \ell(\mathbf{x})$

4: Compute $\hat{\boldsymbol{\xi}} \in \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \hat{\lambda})$

5: **end if**

Algorithms 4 and 5 exploit Lemma 3 and Theorem 3 to compute a multiplier for \mathcal{P} and $\mathcal{P}_{\mathcal{I}}$, respectively, when $\mathbf{x} \in \mathcal{X}$ is a feasible first-stage decision. Since the worst-case Lagrangian functions are monotonically non-decreasing in λ , the algorithms verify if the necessary conditions in Theorem 3 are satisfied for exponentially increasing trial values of $\hat{\lambda}$. Indeed, the factor 2 appearing in line 5 is arbitrary, and any value greater than 1 will also work. We highlight that if $\hat{\lambda}$ is the

smallest possible optimal multiplier, then the algorithm terminates after $\mathcal{O}(\log(\max\{\bar{\lambda}, \lambda^0\}/\lambda^0))$ trial evaluations of $\hat{\lambda}$. Note that computing $\hat{\mathbf{y}}, \hat{\mathbf{z}}$ in line 7 requires solving a deterministic linear program if the second-stage decisions are continuous, or a deterministic mixed-integer program otherwise. Since Theorem 3 is only a necessary condition, it is possible that the final multiplier $\hat{\lambda}$ and parameter realization $\hat{\boldsymbol{\xi}}$ are not optimal and hence, $\mathcal{L}(\mathbf{x}, \hat{\boldsymbol{\xi}}, \hat{\lambda}) < \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$. We shall discuss a practical workaround in the next subsection.

Algorithm 4 Computation of worst-case parameter realization for general \mathcal{P}

Input: $\mathbf{x} \in \mathcal{X}$, $\lambda^0 > 0$

Output: Either $\hat{\boldsymbol{\xi}} \in \Xi : \mathcal{Q}(\mathbf{x}, \hat{\boldsymbol{\xi}}) = +\infty$, $\hat{\lambda} = \lambda^0$ or $\hat{\boldsymbol{\xi}}, \hat{\lambda}$ satisfying conditions of Theorem 3

- 1: Compute $\hat{\boldsymbol{\xi}} \in \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{S}(\mathbf{x}, \boldsymbol{\xi})$ and set $\hat{\lambda} = \lambda^0$
 - 2: **if** $\mathcal{S}(\mathbf{x}, \hat{\boldsymbol{\xi}}) = 0$ **then**
 - 3: Set $\hat{\lambda} = \lambda^0/2$
 - 4: **repeat**
 - 5: Update $\hat{\lambda} \leftarrow 2\hat{\lambda}$
 - 6: Compute $\hat{\boldsymbol{\xi}} \in \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \hat{\lambda})$
 - 7: Compute $(\hat{\mathbf{y}}, \hat{\mathbf{z}}) \in \arg \min_{(\mathbf{y}, \mathbf{z}) \in \mathcal{Y} \times [0,1]^{n_p}} \left\{ \mathbf{d}(\hat{\boldsymbol{\xi}})^\top \mathbf{y} + \hat{\lambda} \phi(\mathbf{z}, \hat{\boldsymbol{\xi}}) : \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} \geq \mathbf{h}(\mathbf{z}) \right\}$
 - 8: **until** $\phi(\hat{\mathbf{z}}, \hat{\boldsymbol{\xi}}) = 0$
 - 9: **end if**
-

Algorithm 5 Computation of worst-case parameter realization for general $\mathcal{P}_{\mathcal{I}}$

Input: $\mathbf{x} \in \mathcal{X}$, $\lambda^0 > 0$

Output: Either $\hat{\boldsymbol{\xi}} \in \Xi : \mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \hat{\boldsymbol{\xi}}) = +\infty$, $\hat{\lambda} = \lambda^0$ or $\hat{\boldsymbol{\xi}}, \hat{\lambda}$ satisfying conditions of Theorem 3

- 1: Compute $\hat{\boldsymbol{\xi}} \in \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{S}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi})$ and set $\hat{\lambda} = \lambda^0$
 - 2: **if** $\mathcal{S}_{\mathcal{I}}(\mathbf{x}, \hat{\boldsymbol{\xi}}) = 0$ **then**
 - 3: Set $\hat{\lambda} = \lambda^0/2$
 - 4: **repeat**
 - 5: Update $\hat{\lambda} \leftarrow 2\hat{\lambda}$
 - 6: Compute $\hat{\boldsymbol{\xi}} \in \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{L}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}, \hat{\lambda})$
 - 7: Compute $\hat{\mathbf{y}} \in \arg \min_{\mathbf{y} \in \mathcal{Y}} \left\{ \mathbf{d}(\hat{\boldsymbol{\xi}})^\top \mathbf{y} + \hat{\lambda} \phi_{\mathcal{I}}(\mathbf{x}, \mathbf{y}, \hat{\boldsymbol{\xi}}) : \mathbf{g}(\mathbf{x}, \mathbf{y}) \geq \mathbf{0} \right\}$
 - 8: **until** $\phi_{\mathcal{I}}(\mathbf{x}, \hat{\mathbf{y}}, \hat{\boldsymbol{\xi}}) = 0$
 - 9: **end if**
-

Algorithms 4 and 5 appear quite similar to simply computing the maximum of the second-stage value functions over the uncertain parameters. However, their main advantage is that unlike the latter, which require the solution of mixed-integer bilinear problems (refer to Section 2.1.3), all intermediate optimization problems in lines 1, 4 of Algorithm 3, and in lines 1, 6 of Algorithms 4 and 5 are relatively easier to solve. To illustrate this, when the second-stage decisions are continuous ($\mathcal{Y} = \mathbb{R}_+^{n_2}$), it can be readily verified using LP duality that these optimization problems can be

reformulated as follows. Here, we suppose that $\mathbf{h}(z) = \mathbf{h}_0 + \mathbf{H}z$ and $\mathbf{g}(x, \mathbf{y}) = \mathbf{T}x + \mathbf{W}\mathbf{y} - \mathbf{h}_0$.

$$\begin{aligned} \sup_{\xi \in \Xi} \mathcal{S}(x, \xi) &= \left[\begin{array}{l} \text{maximize}_{\xi \in \Xi, \mu \in \mathbb{R}_+^m, \beta \in \mathbb{R}_+^{n_p}} (\mathbf{h}_0 - \mathbf{T}x)^\top \mu + \mathbf{e}^\top (\xi - \beta) \\ \text{subject to} \quad \mathbf{W}^\top \mu \leq \mathbf{0}, \quad \mu \leq \mathbf{e} \\ \quad \quad \quad 2\xi - \mathbf{H}^\top \mu - \beta \geq \mathbf{e}. \end{array} \right] & (\mathcal{WC}_S) \\ \sup_{\xi \in \Xi} \mathcal{L}(x, \xi, \lambda) &= \left[\begin{array}{l} \text{maximize}_{\xi \in \Xi, \mu \in \mathbb{R}_+^m, \beta \in \mathbb{R}_+^{n_p}} \mathbf{c}(\xi)^\top x + (\mathbf{h}_0 - \mathbf{T}x)^\top \mu + \mathbf{e}^\top (\lambda \xi - \beta) \\ \text{subject to} \quad \mathbf{W}^\top \mu \leq \mathbf{d}(\xi) \\ \quad \quad \quad 2\lambda \xi - \mathbf{H}^\top \mu - \beta \geq \lambda \mathbf{e}. \end{array} \right] & (\mathcal{WC}_L) \\ \sup_{\xi \in \Xi} \mathcal{S}_I(x, \xi) &= \left[\begin{array}{l} \text{maximize}_{\xi \in \Xi, \mu \in \mathbb{R}_+^m} (\mathbf{h}_0 - \mathbf{T}x)^\top (\mu - \delta(\mathbf{I}, \xi)) \\ \text{subject to} \quad \mathbf{W}^\top \mu \leq \delta(\mathbf{W}, \xi), \quad \mu \leq \mathbf{e}. \end{array} \right] & (\mathcal{WC}_{S_I}) \\ \sup_{\xi \in \Xi} \mathcal{L}_I(x, \xi, \lambda) &= \left[\begin{array}{l} \text{maximize}_{\xi \in \Xi, \mu \in \mathbb{R}_+^m} \mathbf{c}(\xi)^\top x + (\mathbf{h}_0 - \mathbf{T}x)^\top (\mu - \lambda \delta(\mathbf{I}, \xi)) \\ \text{subject to} \quad \mathbf{W}^\top \mu \leq \mathbf{d}(\xi) + \lambda \delta(\mathbf{W}, \xi). \end{array} \right] & (\mathcal{WC}_{L_I}) \\ \delta(\mathbf{A}, \xi) &= \sum_{j \in [n_p]} \sum_{i \in \mathcal{I}_j^1} \xi_j \mathbf{A}^\top \mathbf{e}_i + \sum_{j \in [n_p]} \sum_{i \in \mathcal{I}_j^0} (1 - \xi_j) \mathbf{A}^\top \mathbf{e}_i, \quad \mathbf{A} \in \mathbb{R}^{m \times n}, \quad n \in \mathbb{Z}_+. \end{aligned}$$

We note that all of the above problems (\mathcal{WC}_S) , (\mathcal{WC}_L) , (\mathcal{WC}_{S_I}) and (\mathcal{WC}_{L_I}) are already in MILP form. This is in contrast to (\mathcal{WC}_D) and (\mathcal{WC}_K) , which are bilinear and need further reformulation before they can be cast as MILP problems. On the one hand, this results in smaller optimization problems to be solved; see Table 1. On the other hand, it does not require any arbitrary big-M upper bounds on dual variables that can lead to suboptimal decisions. We note however that, as mentioned in the description of Algorithm 4, it is crucial to choose tight upper bounds $u(\mathbf{x})$ on the worst-case objective value of \mathbf{x} and tight lower bounds $\ell(\mathbf{x})$ on the best-case objective value of \mathbf{x} when computing the optimal multiplier in \mathcal{P} . This will impact numerical performance when solving (\mathcal{WC}_L) .

4.1 Integration in Benders decomposition algorithm

Algorithm 6 presents a Benders decomposition scheme to solve \mathcal{P}_I when the second-stage decisions are continuous ($\mathcal{Y} = \mathbb{R}_+^{n_2}$). Note that relatively complete recourse is not assumed. The algorithm iteratively refines inner approximations of $\sup_{\lambda \geq 0} \sup_{\xi \in \Xi} \mathcal{L}_I(\mathbf{x}, \xi, \lambda)$ and $\sup_{\xi \in \Xi} \mathcal{S}_I(\mathbf{x}, \xi, \lambda)$. Indeed, the former is reflected in the objective function of the lower bounding problem (15), where the set \mathcal{O} enumerates the extreme points of the feasible region of problem (\mathcal{WC}_{L_I}) . Similarly, note that we have:

$$\inf_{\mathbf{x} \in \mathcal{X}} \sup_{\xi \in \Xi} \mathcal{L}_I(\mathbf{x}, \xi, \lambda) = \inf_{\mathbf{x} \in \mathcal{X}} \left\{ \max_{\xi \in \Xi} \mathcal{L}_I(\mathbf{x}, \xi, \lambda) : \mathcal{S}_I(\mathbf{x}, \xi) \leq 0 \quad \forall \xi \in \Xi \right\}.$$

In other words, the constraints of the lower bounding problem (15), written for all points in \mathcal{F} , enumerate the extreme points of the feasible region of problem (\mathcal{WC}_{S_I}) . In Benders terminology, the sets \mathcal{O} and \mathcal{F} corresponding to feasibility and optimality cuts, respectively.

It is crucial to note that the algorithm implicitly exploits the strong duality result in Theorem 2. Indeed, since we have:

$$\inf_{\mathbf{x} \in \mathcal{X}} \sup_{\xi \in \Xi} \mathcal{Q}_I(\mathbf{x}, \xi, \lambda) \geq \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\xi \in \Xi} \mathcal{L}_I(\mathbf{x}, \xi, \lambda), \quad \lambda \geq 0,$$

Algorithm 6 Benders decomposition scheme along with Algorithm 5 to solve $\mathcal{P}_{\mathcal{I}}$ when $\mathcal{Y} = \mathbb{R}_+^{n_2}$ and $g(\mathbf{x}, \mathbf{y}) = \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} - \mathbf{h}_0$

Input: Problem $\mathcal{P}_{\mathcal{I}}$, tolerance $\epsilon > 0$

Output: \mathbf{x}^* (an optimal solution of $\mathcal{P}_{\mathcal{I}}$ if it is feasible or \emptyset otherwise)

1: Initialize $LB = -\infty$, $UB = +\infty$, $\mathbf{x}^* = \emptyset$, $\mathcal{O} = \emptyset$, $\mathcal{F} = \emptyset$, $\lambda = 1$.

2: **repeat**

3: Set LB as the optimal objective value of the problem:

$$\begin{aligned} & \underset{\mathbf{x} \in \mathcal{X}}{\text{minimize}} \quad \sup_{(\boldsymbol{\xi}, \boldsymbol{\mu}) \in \mathcal{O}} \left\{ \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + (\boldsymbol{\mu} - \lambda \boldsymbol{\delta}(\mathbf{I}, \boldsymbol{\xi}))^\top (\mathbf{h}_0 - \mathbf{T}\mathbf{x}) \right\} \\ & \text{subject to} \quad (\boldsymbol{\mu} - \boldsymbol{\delta}(\mathbf{I}, \boldsymbol{\xi}))^\top (\mathbf{h}_0 - \mathbf{T}\mathbf{x}) \leq 0, \quad (\boldsymbol{\xi}, \boldsymbol{\mu}) \in \mathcal{F}. \end{aligned} \quad (15)$$

4: **if** $LB < +\infty$ **then**

5: Set $\hat{\mathbf{x}}$ as an optimal solution of problem (15)

6: Compute $\hat{\boldsymbol{\xi}}$, $\hat{\lambda}$ by using Algorithm 5 (with inputs $\hat{\mathbf{x}}$, λ) and executing its lines 1 and 6 by computing optimal solutions $(\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\mu}})$ of problems $\mathcal{WC}_{\mathcal{S}_{\mathcal{I}}}$ (at $\mathbf{x} = \hat{\mathbf{x}}$) and $\mathcal{WC}_{\mathcal{L}_{\mathcal{I}}}$ (at $\mathbf{x} = \hat{\mathbf{x}}$ and $\lambda = \hat{\lambda}$), respectively.

7: **if** $\mathcal{S}_{\mathcal{I}}(\hat{\mathbf{x}}, \hat{\boldsymbol{\xi}}) > 0$ **then**

8: Update $\mathcal{F} \leftarrow \mathcal{F} \cup \{(\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\mu}})\}$

9: **else**

10: Update $\mathcal{O} \leftarrow \mathcal{O} \cup \{(\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\mu}})\}$.

11: Update $\lambda \leftarrow \hat{\lambda}$

12: **if** $UB > \mathcal{L}_{\mathcal{I}}(\hat{\mathbf{x}}, \hat{\boldsymbol{\xi}}, \hat{\lambda})$ **then** update $UB \leftarrow \mathcal{L}_{\mathcal{I}}(\hat{\mathbf{x}}, \hat{\boldsymbol{\xi}}, \hat{\lambda})$ and $\mathbf{x}^* \leftarrow \hat{\mathbf{x}}$. **end if**

13: **end if**

14: **else**

15: Set $\hat{\mathbf{x}} = \emptyset$

16: **end if**

17: **until** $UB - LB \leq \epsilon$ or $LB = +\infty$

problem (15) provides a rigorous lower bound on the optimal value of $\mathcal{P}_{\mathcal{I}}$, even though $(\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\mu}})$ may correspond to extreme points of $(\mathcal{WC}_{\mathcal{L}_{\mathcal{I}}})$ for potentially different values of λ . The algorithm terminates finitely since Algorithm 5 also terminates in a finite number of iterations, only a finite number of trial values of λ are enumerated, and because the feasible regions of $(\mathcal{WC}_{\mathcal{L}_{\mathcal{I}}})$ and $(\mathcal{WC}_{\mathcal{S}_{\mathcal{I}}})$ have a finite number of extreme points for any $\lambda \geq 0$. Note that if the problem is known to satisfy relatively complete recourse, then $\mathcal{F} = \emptyset$ is guaranteed in all iterations of the algorithm.

Since Algorithm 5 computes a Lagrange multiplier that only satisfies the necessary optimality conditions in Theorem 3, it is possible that Algorithm 6 outputs a solution whose worst-case objective value is estimated to be less than the optimal value of $\mathcal{P}_{\mathcal{I}}$. We can verify if this happens to be the case at the end of algorithm by solving a mixed-integer bilinear variant of $(\mathcal{WC}_{\mathcal{L}_{\mathcal{I}}})$ where $\lambda \geq 0$ is also a decision variable. Here, the bilinear products $\lambda \boldsymbol{\xi}$ can be reformulated by adding variables $\boldsymbol{\rho} \in \mathbb{R}_+^{n_p}$ and indicator constraints:

$$\begin{aligned} \xi_j = 0 & \implies \rho_j = 0, \quad j \in [n_p], \\ \xi_j = 1 & \implies \rho_j = \lambda, \quad j \in [n_p]. \end{aligned}$$

If the optimal value of this problem is larger than predicted (UB), then we can use its optimal solution λ to initialize another run of the algorithm with an appropriately updated UB . In doing

so, we can retain the feasibility and optimality sets, \mathcal{F} and \mathcal{O} , and do not have to re-initialize them to be empty sets. We emphasize that if Algorithm 3 outputs the true optimal multiplier λ , then the bilinear problem has to be solved only at most once (because of the strong duality result in Theorem 2).

Finally, we note that the corresponding Benders scheme for \mathcal{P} is similar with the difference that it enumerates the extreme points of $(\mathcal{WC}_{\mathcal{L}})$ and $(\mathcal{WC}_{\mathcal{S}})$ instead. Also, the worst-case parameter realizations can be computed using Algorithm 3 whenever the conditions therein are satisfied.

4.2 Integration in column-and-constraint generation algorithm

Algorithm 7 presents a column-and-constraint generation scheme for solving problem $\mathcal{P}_{\mathcal{I}}$ without assuming relatively complete recourse. It is quite similar to Algorithm 2 for solving problem \mathcal{P} . The main difference is in the formulation of the lower bounding problem (16), where we note the absence of indicator constraints, since all constraints are unconditionally known for each parameter realization recorded in the set \mathcal{R} . We also note that old values of the Lagrange multipliers λ are used as inputs to Algorithm 5 so that λ is non-decreasing across algorithmic iterations. The algorithm terminates finitely since Algorithm 5 also terminates in a finite number of iterations and Ξ is a finite set. The corresponding column-and-constraint generation scheme for \mathcal{P} is similar with the difference that the worst-case parameter realizations are computed using Algorithm 4, or using Algorithm 3 whenever applicable.

Algorithm 7 Column-and-constraint generation scheme with Algorithm 5 to solve $\mathcal{P}_{\mathcal{I}}$

Input: Problem $\mathcal{P}_{\mathcal{I}}$, tolerance $\epsilon > 0$

Output: \mathbf{x}^* (an optimal solution of $\mathcal{P}_{\mathcal{I}}$ if it is feasible or \emptyset otherwise)

1: Initialize $LB = -\infty$, $UB = +\infty$, $\mathbf{x}^* = \emptyset$, $\mathcal{R} = \emptyset$, $\lambda = 1$.

2: **repeat**

3: Set LB as the optimal objective value of the problem:

$$\inf_{\mathbf{x}, \eta, \mathbf{y}} \left\{ \begin{array}{l} \mathbf{x} \in \mathcal{X}, \quad \eta \in \mathbb{R}, \quad \mathbf{y}^{(\xi)} \in \mathcal{Y}, \quad \xi \in \mathcal{R}, \\ \eta \geq \mathbf{c}(\xi)^\top \mathbf{x} + \mathbf{d}(\xi)^\top \mathbf{y}^{(\xi)}, \quad \xi \in \mathcal{R}, \\ \eta : \mathbf{g}(\mathbf{x}, \mathbf{y}^{(\xi)}) \geq \mathbf{0}, \quad \xi \in \mathcal{R}, \\ g_i(\mathbf{x}, \mathbf{y}^{(\xi)}) = 0, \quad i \in \mathcal{I}_j^0, \quad j \in [n_p] : \xi_j = 0, \quad \xi \in \mathcal{R}, \\ g_i(\mathbf{x}, \mathbf{y}^{(\xi)}) = 0, \quad i \in \mathcal{I}_j^1, \quad j \in [n_p] : \xi_j = 1, \quad \xi \in \mathcal{R}. \end{array} \right. \quad (16)$$

4: **if** $LB < +\infty$ **then**

5: Set $\hat{\mathbf{x}}$ as an optimal (projected) solution of problem (16)

6: Compute $\hat{\xi}, \hat{\lambda}$ by using Algorithm 5 (with inputs $\hat{\mathbf{x}}, \lambda$)

7: Update $\mathcal{R} \leftarrow \mathcal{R} \cup \{\hat{\xi}\}$ and $\lambda \leftarrow \hat{\lambda}$

8: **if** $UB > \mathcal{L}_{\mathcal{I}}(\hat{\mathbf{x}}, \hat{\xi}, \hat{\lambda})$ **then** update $UB \leftarrow \mathcal{L}_{\mathcal{I}}(\hat{\mathbf{x}}, \hat{\xi}, \hat{\lambda})$ and $\mathbf{x}^* \leftarrow \hat{\mathbf{x}}$. **end if**

9: **else**

10: Set $\hat{\mathbf{x}} = \emptyset$

11: **end if**

12: **until** $UB - LB \leq \epsilon$ or $LB = +\infty$

We re-emphasize a few key points from the previous subsection that apply here as well. First, Algorithm 7 rigorously identifies if a solution (or the problem instance itself) is infeasible. Second,

it computes rigorous lower bounds LB on the optimal value of $\mathcal{P}_{\mathcal{I}}$. Finally, it is possible that the scheme terminates at a solution whose worst-case objective value is estimated to be less than the optimal value of $\mathcal{P}_{\mathcal{I}}$, since Algorithm 5 computes a Lagrange multiplier that only satisfies the necessary optimality conditions in Theorem 3. At the end of the algorithm, we can verify if this happens to be the case by solving the true problem $\mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi})$. If it happens to be larger than UB , we can keep increasing λ (at an exponential rate) in Algorithm 5 until we observe an increase in the value of $\mathcal{L}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}, \lambda)$ and the necessary optimality conditions are satisfied. We can then continue the column-and-constraint generation Algorithm 7 after updating UB but retaining LB and \mathcal{R} without re-initializing them. Note that if Algorithm 3 outputs the true optimal multiplier λ , then the true problem $\mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi})$ has to be solved only at most once (because of the strong duality result in Theorem 2). Indeed, this was the case in all our experiments, which makes us suspect that the necessary conditions in Theorem 3 are almost sufficient (although one can construct counter-examples).

When the second-stage decisions are continuous ($\mathcal{Y} = \mathbb{R}_+^{n_2}$), the intermediate optimization problems in lines 1 and 6 of Algorithm 5 can be computed by solving problems $\mathcal{WC}_{\mathcal{S}_{\mathcal{I}}}$ and $\mathcal{WC}_{\mathcal{L}_{\mathcal{I}}}$, similar to the Benders scheme. When the second-stage decisions are mixed-integer, [57] have proposed an inner-level column-and-constraint generation algorithm under the assumption of extended relatively complete recourse, which we shall define shortly. In the following, we briefly describe how our method can be integrated in this inner-level algorithm and without requiring this assumption. We only present it for the general formulation \mathcal{P} assuming that the conditions of Theorem 5 are satisfied so that Algorithm 3 is applicable. The extension toward instances of \mathcal{P} and $\mathcal{P}_{\mathcal{I}}$ without these conditions is straightforward but requires additional notation, and so we omit it in the interest of brevity.

Consider the following instance of problem \mathcal{P} with both continuous (\mathbf{y}_c) and discrete (\mathbf{y}_d) second-stage decisions.

$$\begin{aligned} & \inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}), \\ \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) = & \left[\begin{array}{l} \text{minimize}_{\mathbf{y} \in \mathcal{Y}} \quad \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}_c(\boldsymbol{\xi})^\top \mathbf{y}_c + \mathbf{d}_d(\boldsymbol{\xi})^\top \mathbf{y}_d \\ \text{subject to} \quad \mathbf{T}\mathbf{x} + \mathbf{W}_c \mathbf{y}_c + \mathbf{W}_d \mathbf{y}_d \geq \mathbf{h}(\boldsymbol{\xi}) \\ \quad \quad \quad \mathbf{y} = (\mathbf{y}_c, \mathbf{y}_d) \in \mathbb{R}^{n_{c2}} \times \mathcal{Y}_d \end{array} \right], \end{aligned} \quad (17)$$

where $\mathcal{Y} = \mathbb{R}^{n_{c2}} \times \mathcal{Y}_d$ and $\mathcal{Y}_d \subseteq \mathbb{Z}^{n_{d2}}$ is a compact MILP representable set.

Before we present the algorithm for computing worst-case parameter realizations, we note that the Lagrangian and constraint violation functions are exactly the same as in the case of problem \mathcal{P} that we presented in Section 3. For example, the former is defined as follows:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) = \left[\begin{array}{l} \text{minimize}_{\mathbf{y} \in \mathcal{Y}, \mathbf{z} \in \mathbb{R}_+^{n_p}} \quad \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}_c(\boldsymbol{\xi})^\top \mathbf{y}_c + \mathbf{d}_d(\boldsymbol{\xi})^\top \mathbf{y}_d + \lambda \phi(\mathbf{z}, \boldsymbol{\xi}) \\ \text{subject to} \quad \mathbf{T}\mathbf{x} + \mathbf{W}_c \mathbf{y}_c + \mathbf{W}_d \mathbf{y}_d \geq \mathbf{h}(\mathbf{z}), \quad \mathbf{z} \leq \mathbf{e}. \end{array} \right]$$

with $\mathcal{S}(\mathbf{x}, \boldsymbol{\xi})$ defined similarly. The key observation in [57] is to note that the worst-case values of the Lagrangian and constraint violation functions (over $\boldsymbol{\xi} \in \Xi$) can be reformulated as semi-infinite programs. For example, we can reformulate $\sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda)$ as follows:

$$\sup_{\boldsymbol{\xi} \in \Xi, \eta \in \mathbb{R}} \left\{ \eta : \eta \leq \left[\begin{array}{l} \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}_d(\boldsymbol{\xi})^\top \mathbf{y}_d \\ + \inf_{(\mathbf{y}_c, \mathbf{z}) \in \Pi(\mathbf{x}, \mathbf{y}_d)} \left\{ \mathbf{d}_c(\boldsymbol{\xi})^\top \mathbf{y}_c + \lambda \phi(\mathbf{z}, \boldsymbol{\xi}) \right\} \end{array} \right], \mathbf{y}_d \in \mathcal{Y}_d \right\}$$

$$\begin{aligned}
&= \sup_{\xi \in \Xi, \eta \in \mathbb{R}} \left\{ \eta : \eta \leq \left[\begin{array}{l} \mathbf{c}(\xi)^\top \mathbf{x} + \mathbf{d}_d(\xi)^\top \mathbf{y}_d \\ + \sup_{(\mu, \beta) \in \Delta(\xi, \lambda, 1)} \left\{ [\mathbf{h}_0 - \mathbf{T}\mathbf{x} - \mathbf{W}_d \mathbf{y}_d]^\top \boldsymbol{\mu} + \mathbf{e}^\top (\lambda \xi - \boldsymbol{\beta}) \right\} \end{array} \right], \mathbf{y}_d \in \mathcal{Y}_d \right\} \\
\Pi(\mathbf{x}, \mathbf{y}_d) &= \{(\mathbf{y}_c, \mathbf{z}) \in \mathbb{R}_+^{nc_2} \times [0, 1]^{n_p} : \mathbf{W}_c \mathbf{y}_c - \mathbf{H}\mathbf{z} \geq \mathbf{h}_0 - \mathbf{T}\mathbf{x} - \mathbf{W}_d \mathbf{y}_d\} \\
\Delta(\xi, \lambda, \kappa) &= \{(\boldsymbol{\mu}, \boldsymbol{\beta}) \in \mathbb{R}_+^m \times \mathbb{R}_+^{n_p} : \mathbf{W}^\top \boldsymbol{\mu} \leq \kappa \mathbf{d}_c(\xi), 2\lambda \xi - \mathbf{H}^\top \boldsymbol{\mu} - \boldsymbol{\beta} \geq \lambda \mathbf{e}\}
\end{aligned}$$

and $\sup_{\xi \in \Xi} \mathcal{S}(\mathbf{x}, \xi)$ can be reformulated similarly.

Algorithm 8 presents the computation of the worst-case parameter realization in the presence of mixed-integer second-stage decisions \mathbf{y}_d in problem (17) assuming that it satisfies the conditions in Theorem 5. We note that it does not rely on the extended relatively complete recourse assumption, which requires the second-stage problem to be feasible for any fixed values of the first-stage decisions $\mathbf{x} \in \mathcal{X}$, uncertain parameters $\xi \in \Xi$ and discrete-valued second-stage decisions $\mathbf{y}_d \in \mathcal{Y}_d$. The key idea of the algorithm is to solve the above semi-infinite reformulation by enumerating \mathcal{Y}_d in the set \mathcal{D} . Indeed, it adds a full set of dual variables $(\boldsymbol{\mu}^{(\mathbf{y}_d)}, \boldsymbol{\beta}^{(\mathbf{y}_d)})$ and associated constraints for each identified $\mathbf{y}_d \in \mathcal{D}$ and thus obtains upper bounds on $\sup_{\xi \in \Xi} \mathcal{S}(\mathbf{x}, \xi)$ and $\sup_{\xi \in \Xi} \mathcal{L}(\mathbf{x}, \xi, \lambda)$ in lines 3 and 12, respectively. Note that the optimization problems in these lines can be solved with any MILP solver. The algorithm terminates in a finite number of iterations because \mathcal{Y}_d is a finite set.

5 Numerical Experiments

The primary goal of our numerical experiments is to measure the computational benefits of using the Lagrangian formulation over traditional methods in computing worst-case parameter realizations, whereas the secondary goal is to assess its impacts on the run time of the overall exact method. To that end, we consider three different applications: network design with uncertain edge failures, facility location with random facility disruptions, and staff rostering under demand uncertainty. Collectively, they represent instances of both \mathcal{P} and $\mathcal{P}_{\mathcal{I}}$, feature the presence and absence of relatively complete recourse as well as continuous and mixed-integer second-stage decisions. Notably, all of these applications have been previously studied in the literature and problem-specific column-and-constraint generation methods have been developed for their solution.

The experiments were implemented in Julia using JuMP [20] as the modeling language and Gurobi 9.1 [29] (with default options) as the LP and MILP solver. All runs were conducted on an Intel Xeon 2.3 GHz computer, with a limit of eight cores per run. Our code is available at <https://github.com/AnirudhSubramanyam/RobustOptLagrangianDual.jl>.

5.1 Network design with uncertain edge failures

We first consider the network design problem that was studied in [42]. Consider a network with nodes V and edges E . Each node $i \in V$ has degree D_i and it is associated with a (positive or negative) demand $b_i \in \mathbb{R}$ for a single commodity. The origin and destination nodes of each edge $e \in E$ are denoted by $o(e) \in V$ and $d(e) \in V$, respectively, whereas its pre-installed capacity is denoted by $p_e \in \mathbb{R}_+$. Each edge may fail randomly and the uncertain parameter $\xi_e \in \{0, 1\}$ indicates whether the edge e has failed. The goal is to decide (before any failures occur) the amount of additional capacity $x_e \in \mathbb{R}_+$ that must be installed for each edge $e \in E$, so that all nodal demands can be satisfied using feasible forward and reverse edge flows, denoted $y_e^f \in \mathbb{R}_+$ and $y_e^b \in \mathbb{R}_+$, respectively, under any postulated realization of the edge failures $\xi \in \Xi$. This problem

Algorithm 8 Computation of worst-case parameter realization for (17) using the inner-level column-and-constraint generation scheme and Algorithm 3

Input: $\mathbf{x} \in \mathcal{X}$

Output: $\hat{\boldsymbol{\xi}} \in \arg \max_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi})$ and $\hat{\lambda} \in \arg \max_{\lambda \in \mathbb{R}_+} \left\{ \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\mathbf{x}, \boldsymbol{\xi}, \lambda) \right\}$

1: Initialize $\hat{\boldsymbol{\xi}} \in \Xi$ (arbitrary), $LB = -\infty$, $UB = +\infty$, $\mathcal{D} = \emptyset$, $\hat{\lambda} = 1$

2: **repeat**

3: Set UB and $\hat{\boldsymbol{\xi}}$ as the optimal objective value and (projected) solution of the problem:

$$\inf_{\mathbf{x}, \eta, \mathbf{y}} \left\{ \eta : \begin{array}{l} \boldsymbol{\xi} \in \Xi, \quad \eta \in \mathbb{R}, \quad (\boldsymbol{\mu}^{(\mathbf{y}_d)}, \boldsymbol{\beta}^{(\mathbf{y}_d)}) \in \Delta(\boldsymbol{\xi}, 1, 0), \quad \boldsymbol{\mu}^{(\mathbf{y}_d)} \leq \mathbf{e}, \quad \mathbf{y}_d \in \mathcal{D} \\ \eta \leq (\mathbf{h}_0 - \mathbf{T}\mathbf{x} - \mathbf{W}_d \mathbf{y}_d)^\top \boldsymbol{\mu}^{(\mathbf{y}_d)} + \mathbf{e}^\top (\boldsymbol{\xi} - \boldsymbol{\beta}^{(\mathbf{y}_d)}), \quad \mathbf{y}_d \in \mathcal{D} \end{array} \right\}$$

4: Set $(\hat{\mathbf{y}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\sigma}}) \in \arg \min_{(\mathbf{y}, \mathbf{z}, \boldsymbol{\sigma}) \in \mathcal{Y} \times [0, 1]^{n_p} \times \mathbb{R}_+^m} \{ \mathbf{e}^\top \boldsymbol{\sigma} + \phi(\mathbf{z}, \hat{\boldsymbol{\xi}}) : \mathbf{T}\mathbf{x} + \mathbf{W}_c \mathbf{y}_c + \mathbf{W}_d \mathbf{y}_d + \boldsymbol{\sigma} \geq \mathbf{h}(\mathbf{z}) \}$

5: Update $\mathcal{D} \leftarrow \mathcal{D} \cup \{ \hat{\mathbf{y}}_d \}$.

6: Update $LB = \mathbf{e}^\top \hat{\boldsymbol{\sigma}} + \phi(\hat{\mathbf{z}}, \hat{\boldsymbol{\xi}})$

7: **until** $LB > 0$ or $UB = 0$

8: **if** $UB = 0$ **then**

9: Compute $\ell(\mathbf{x})$ and $u(\mathbf{x})$ as per Theorem 4

10: Set $\hat{\lambda} = u(\mathbf{x}) - \ell(\mathbf{x})$ and $LB = -\infty$

11: **repeat**

12: Set UB and $\tilde{\boldsymbol{\xi}}$ as the optimal objective value and (projected) solution of the problem:

$$\inf_{\mathbf{x}, \eta, \mathbf{y}} \left\{ \eta : \begin{array}{l} \boldsymbol{\xi} \in \Xi, \quad \eta \in \mathbb{R}, \quad (\boldsymbol{\mu}^{(\mathbf{y}_d)}, \boldsymbol{\beta}^{(\mathbf{y}_d)}) \in \Delta(\boldsymbol{\xi}, \hat{\lambda}, 1), \quad \mathbf{y}_d \in \mathcal{D} \\ \eta \leq \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} + \mathbf{d}_d(\boldsymbol{\xi})^\top \mathbf{y}_d + (\mathbf{h}_0 - \mathbf{T}\mathbf{x} - \mathbf{W}_d \mathbf{y}_d)^\top \boldsymbol{\mu}^{(\mathbf{y}_d)} + \mathbf{e}^\top (\hat{\lambda} \boldsymbol{\xi} - \boldsymbol{\beta}^{(\mathbf{y}_d)}), \quad \mathbf{y}_d \in \mathcal{D} \end{array} \right\}$$

13: Compute $\hat{\mathbf{y}} \in \arg \min_{\mathbf{y} \in \mathcal{Y}} \{ \mathbf{d}_c(\tilde{\boldsymbol{\xi}})^\top \mathbf{y}_c + \mathbf{d}_d(\tilde{\boldsymbol{\xi}})^\top \mathbf{y}_d : \mathbf{T}\mathbf{x} + \mathbf{W}_c \mathbf{y}_c + \mathbf{W}_d \mathbf{y}_d \geq \mathbf{h}(\tilde{\boldsymbol{\xi}}) \}$

14: Update $\mathcal{D} \leftarrow \mathcal{D} \cup \{ \hat{\mathbf{y}}_d \}$.

15: **if** $LB < \mathbf{d}_c(\tilde{\boldsymbol{\xi}})^\top \hat{\mathbf{y}}_c + \mathbf{d}_d(\tilde{\boldsymbol{\xi}})^\top \hat{\mathbf{y}}_d$ **then** $LB \leftarrow \mathbf{d}_c(\tilde{\boldsymbol{\xi}})^\top \hat{\mathbf{y}}_c + \mathbf{d}_d(\tilde{\boldsymbol{\xi}})^\top \hat{\mathbf{y}}_d$, $\hat{\boldsymbol{\xi}} \leftarrow \tilde{\boldsymbol{\xi}}$ **end if**

16: **until** $UB - LB \leq \epsilon$

17: **end if**

can be formulated as the following instance of $\mathcal{P}_{\mathcal{I}}$.

$$\inf_{\mathbf{x} \in \mathbb{R}_+^{|E|}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}),$$

$$\mathcal{Q}_{\mathcal{I}}(\mathbf{x}, \boldsymbol{\xi}) = \left[\begin{array}{l} \text{minimize } (\mathbf{c}^x)^\top \mathbf{x} + (\mathbf{c}^y)^\top (\mathbf{y}^f + \mathbf{y}^b) \\ \mathbf{y}^f, \mathbf{y}^b \in \mathbb{R}_+^{|E|} \\ \text{subject to } \sum_{e \in E: d(e)=i} (y_e^f - y_e^b) + \sum_{e \in E: o(e)=i} (y_e^b - y_e^f) = b_i, \quad i \in V \\ \mathbf{y}^f \leq \mathbf{x} + \mathbf{p}, \quad \mathbf{y}^b \leq \mathbf{x} + \mathbf{p}, \\ \xi_e = 1 \implies y_e^f = 0, \quad e \in E \\ \xi_e = 1 \implies y_e^b = 0, \quad e \in E \end{array} \right],$$

where for each edge $e \in E$, $c_e^x \in \mathbb{R}$ and $c_e^y \in \mathbb{R}$ represent the per-unit capacity installation and transportation costs, respectively. The uncertainty set is parameterized by a budget of uncertainty $k \geq 0$, and defined as follows:

$$\Xi = \left\{ \boldsymbol{\xi} \in \{0, 1\}^{|E|} : \mathbf{e}^\top \boldsymbol{\xi} \leq k, \sum_{e \in E: d(e)=i} \xi_e + \sum_{e \in E: o(e)=i} \xi_e \leq D_i - 1, i \in V_T \right\},$$

where $V_T \subseteq V$ denotes the subset of terminal nodes; that is, those with non-zero demands. Thus, the uncertainty set models the simultaneous failure of up to k edges at any moment. We refer readers to [42] for the motivation behind the choice of this uncertainty set, the larger context of the network design problem, as well as additional details about the problem formulation.

In [42], the authors use a big-M representation of the indicator constraints as follows: $y_e^f, y_e^b \leq (x_e + p_e)(1 - \xi_e)$ for each $e \in E$. Nevertheless, we retain the indicators since our method can exploit that structure. Also, we note that this problem does not satisfy the relatively complete recourse assumption.

We conduct experiments on the same set of instances as [42] with the caveat that we do not preprocess the original network superstructure as described in Appendix B of that paper. Also, we set $c_e^x/c_e^y = 1$ for all $e \in E$, whereas [42] study a range of different values for this ratio (including the one that we consider). Apart from these differences, we benchmark our method against the one proposed in [42], which is a variant of the original column-and-constraint generation algorithm proposed in [58]. Specifically, [42] modify the original algorithm to account for the lack of relatively complete recourse and they use the duality-based mixed-integer bilinear formulation ($\mathcal{WC}_{\mathcal{D}}$) for computing worst-case parameter realizations; the latter is further reformulated using additional variables and indicator constraints (see Section 2.1.3) and solved using an MILP solver. We benchmark (our implementation) of this method against Algorithm 7, which uses Algorithm 5 for computing worst-case parameter realizations. In both cases, we set a total time limit of 1 hour.

Table 2 summarizes and compares the computational performance of the existing algorithm in [42] against our proposed method for three representative test instances, since all other instances are solved in less than two minutes on average. In this table, the columns # It., t_{tot} and $t_{\text{wc}}^{\text{avg}}$ report the total number of iterations of the column-and-constraint generation algorithm, the total run time of the algorithm, and the average time to compute worst-case parameter realizations, respectively. The last column $t_{\text{wc}}^{\text{speedup}}$ reports the average speedup with respect to the time for computing worst-case realizations and it is simply the ratio of the corresponding $t_{\text{wc}}^{\text{avg}}$ values. We note that in all instances, the largest reported k value corresponds to the smallest uncertainty set for which that instance becomes infeasible, and the corresponding column t_{tot} reports the time to prove infeasibility. Figure 1 graphically shows the same quantities but averages them across all instances for each k .

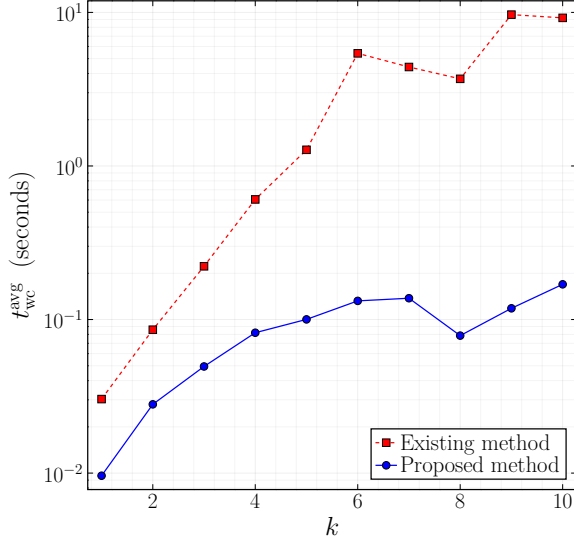
Table 2 and Figure 1 show that the proposed method can offer an average speedup of more than a factor of 30 for computing worst-case parameter realizations. Since the majority of the total run time is spent for this purpose, this directly translates to a speedup in the total run time of the algorithm as well. In particular, we observe that the proposed method can also compute optimal solutions (within 2 minutes) for cases where the existing method can not find a feasible solution in the time limit of 1 hour. Finally, we can observe from Table 2 that that the two methods do not necessarily execute the same number of column-and-constraint generation iterations. The main reason for this difference is that whenever a candidate first-stage decision \boldsymbol{x} is infeasible, the parameter realization that is optimal in the worst-case constraint violation formulation ($\mathcal{WC}_{\mathcal{S}_T}$) is not necessarily the same as that identified by the method of [42]. In fact, even when \boldsymbol{x} is feasible, the

Table 2: Computational comparison of the algorithm in [42] and the proposed method for the network design problem.

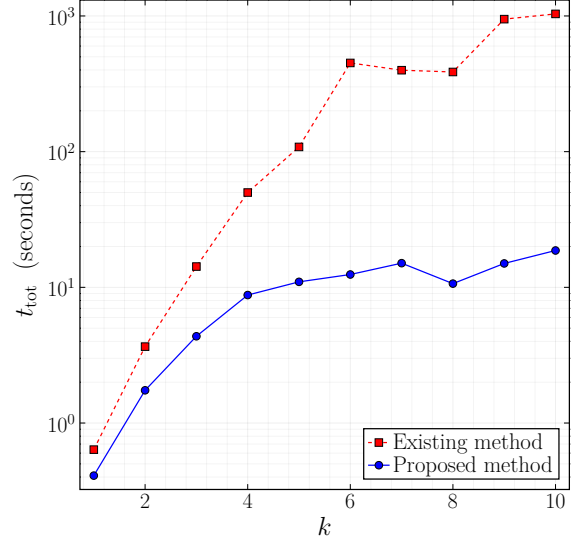
k	Algorithm in [42]			Proposed method			
	# It.	t_{tot} (s)	$t_{\text{wc}}^{\text{avg}}$ (s)	# It.	t_{tot} (s)	$t_{\text{wc}}^{\text{avg}}$ (s)	$t_{\text{wc}}^{\text{speedup}}$
Instance dfn-bwin							
1	10	0.4	0.03	10	0.2	0.01	2.5
2	32	3.0	0.08	33	1.3	0.03	2.8
3	64	23.9	0.33	65	5.3	0.06	5.2
4	73	67.7	0.86	80	10.5	0.11	7.9
5	82	124.6	1.46	86	13.6	0.13	10.8
6	73	565.8	7.33	64	12.3	0.17	42.2
7	67	271.3	3.96	63	9.5	0.13	30.0
8	71	270.1	3.74	69	7.2	0.09	40.9
9	86	1462.0	16.60	88	17.7	0.18	93.4
10	102	1873.9	18.17	87	32.7	0.35	52.2
11	117	TL*	30.49	125	100.4	0.77	39.5
12	106	TL*	33.63	99	119.9	1.17	28.7
13	92	TL*	38.69	85	253.6	2.90	13.3
14	74	TL*	47.98	66	118.4	1.73	27.7
15	83	TL*	42.84	73	58.4	0.78	55.1
16	21	0.8	0.02	13	0.4	0.01	1.6
Instance dfn-gwin							
1	12	1.0	0.03	12	0.8	0.01	3.1
2	35	3.7	0.08	32	1.8	0.03	2.9
3	47	7.8	0.14	60	4.6	0.05	2.9
4	82	62.7	0.70	77	9.6	0.09	8.1
5	81	161.5	1.90	83	12.1	0.11	17.0
6	83	731.7	8.39	80	18.5	0.19	45.3
7	93	826.4	8.59	90	25.7	0.24	36.2
8	118	658.0	5.46	105	12.4	0.09	60.2
9	98	851.6	8.50	102	13.3	0.10	84.3
10	17	1.4	0.03	16	1.1	0.01	2.4
Instance di-yuan							
1	14	0.5	0.03	15	0.3	0.01	4.5
2	37	4.2	0.10	42	2.1	0.03	3.6
3	49	11.0	0.20	51	3.2	0.04	5.3
4	72	19.5	0.25	80	6.3	0.05	4.9
5	80	38.9	0.46	86	7.2	0.05	8.5
6	100	54.3	0.52	103	6.5	0.04	13.7
7	137	97.4	0.68	134	10.1	0.04	15.6
8	120	231.6	1.89	138	12.3	0.05	35.2
9	133	528.9	3.92	124	14.0	0.08	51.3
10	129	1231.7	9.45	117	22.3	0.15	63.7
11	119	2518.6	20.97	112	43.9	0.32	66.5
12	119	10.6	0.06	124	5.7	0.02	2.6
Avg	74.4 [†]	385.3 [†]	8.38	74.0 [†]	10.4 [†]	0.27	30.6

* Reached time limit of 1 hour without providing a feasible solution.

[†] Averaged across instances where both methods terminated within the time limit.



(a) Average time for worst-case realizations



(b) Total run time of algorithm

Figure 1: Comparison of computational times of the existing method in [42] against the proposed method for the network design problem.

two methods may compute different worst-case parameter realizations if the second-stage problem $\mathcal{Q}_I(\mathbf{x}, \boldsymbol{\xi})$ does not happen to have a unique optimal solution.

5.2 Facility location with random facility disruptions

We now consider the facility location problem that was studied in [17]. Let I and J denote sets of customers and facility locations, respectively. Each customer $i \in I$ is associated with some demand $h_i \in \mathbb{R}_+$ and a per-unit penalty cost $p_i \in \mathbb{R}_+$ for failure to satisfy its demand. Each location $j \in J$ is associated with a fixed cost $c_j \in \mathbb{R}_+$ and capacity $C_j \in \mathbb{R}_+$ if a facility is built at that location. Finally, $d_{ij} \in \mathbb{R}_+$ represents the per-unit transportation cost if a facility built at location $j \in J$ satisfies some portion of the demand of customer $i \in I$. Once built, facilities may experience random disruptions and the uncertain parameter $\xi_j \in \{0, 1\}$ indicates whether facility $j \in J$ has been disrupted. The goal is to decide (before any disruptions occur) the subset of locations where facilities must be built, so that the sum of total transportation and penalty costs is minimized in the worst-case disruption scenario. This problem can be formulated as the following instance of \mathcal{P}_I .

$$\inf_{\mathbf{x} \in \{0,1\}^{|J|}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}_I(\mathbf{x}, \boldsymbol{\xi}),$$

$$\mathcal{Q}_I(\mathbf{x}, \boldsymbol{\xi}) = \left[\begin{array}{l} \text{minimize} \quad \mathbf{c}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y} + \mathbf{p}^\top \mathbf{u} \\ \mathbf{y} \in \mathbb{R}_+^{|I| \times |J|}, \mathbf{u} \in \mathbb{R}_+^{|I|} \\ \text{subject to} \quad \sum_{j \in J} y_{ij} + u_i \geq h_i, \quad i \in I \\ \sum_{i \in I} y_{ij} \leq C_j x_j, \quad j \in J \\ \xi_j = 1 \implies \sum_{i \in I} y_{ij} = 0, \quad j \in J \end{array} \right],$$

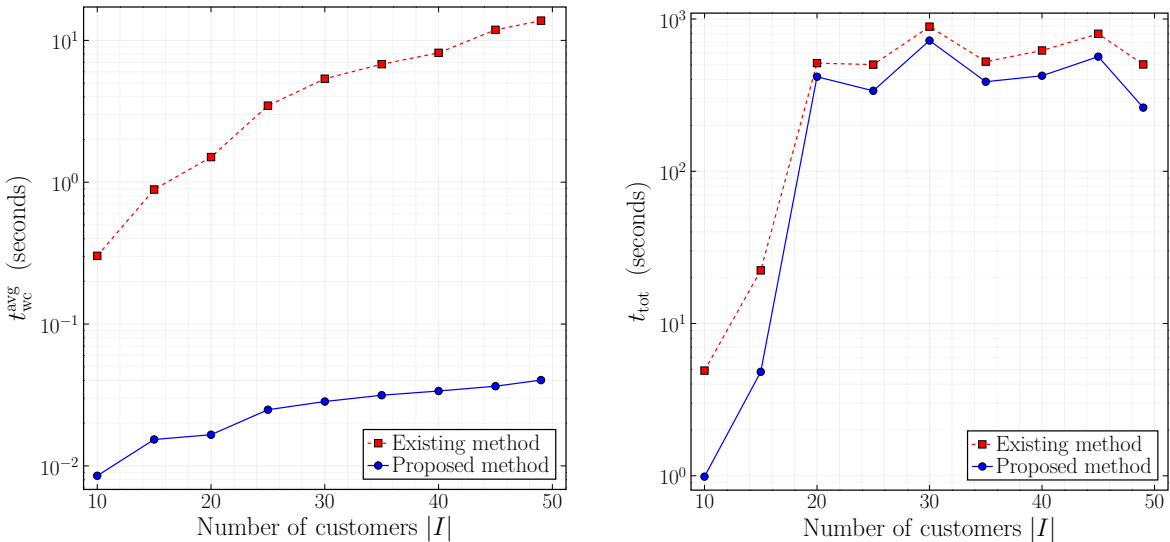
where the uncertainty set is parameterized by a budget of uncertainty $k \geq 0$, and defined as follows:

$$\Xi = \left\{ \boldsymbol{\xi} \in \{0, 1\}^{|J|} : \mathbf{e}^\top \boldsymbol{\xi} \leq k \right\}.$$

Thus the set models the simultaneous disruption of up to k facilities. We note that [16] model the indicator constraints in the second-stage problem as follows: $\sum_{i \in I} y_{ij} \leq C_j x_j (1 - \xi_j)$ for each $j \in J$; but we retain the above structure since our method can exploit it. We refer readers to [17] for additional details about the problem and model formulation.

We conduct experiments on the same set of instances as [17] with the exception that we do not consider uncertainty in customer demands, which that work also addresses. They develop a column-and-constraint generation method for solving the problem. Similar to [58], they propose to use the KKT-based mixed-integer bilinear formulation ($\mathcal{WC}_{\mathcal{K}}$) for computing worst-case parameter realizations; the latter is further reformulated by linearizing the bilinear terms using problem-specific big-M coefficients and solved using an MILP solver. We benchmark (our implementation) of this method against Algorithm 7, which uses Algorithm 5 for computing worst-case parameter realizations. Since the problem satisfies relatively complete recourse by construction, we skip the computation of the constraint violation functions. In all cases, we set a total time limit of 2 hours.

Table 3 summarizes and compares the computational performance of the existing algorithm in [17] against our proposed method, averaged across all test instances with uncertainty set parameter $k = 4$. The columns have the same interpretation as in Table 2. Table 4 summarizes the computational performance across the same test instances as a function of the uncertainty set parameter k , and also includes the average optimality gap, which we define as $100\% \times (UB - LB)/UB$. Finally, Figure 1 graphically shows the same quantities but averages them as a function of the instance size; that is, the number of customers $|I|$.



(a) Average time for worst-case realizations

(b) Total run time of algorithm

Figure 2: Comparison of computational times of the existing method in [17] against the proposed method for the facility location problem.

Tables 3 and 4, and Figure 2a show that the proposed method can offer an average speedup of more than a factor of 240 for computing worst-case parameter realizations. However, unlike the network design application, the majority of the total run time is spent in solving the lower-bounding

Table 3: Computational comparison of the algorithm in [17] and the proposed method for the facility location problem ($k = 4$).

(J , I)	Algorithm in [17]			Proposed method			
	# It.	t_{tot} (s)	$t_{\text{wc}}^{\text{avg}}$ (s)	# It.	t_{tot} (s)	$t_{\text{wc}}^{\text{avg}}$ (s)	$t_{\text{wc}}^{\text{speedup}}$
(10,10)	16	8.4	0.46	16	1.4	0.01	38.9
(10,15)	17	19.1	1.04	17	2.1	0.02	46.4
(10,20)	7	12.2	1.71	7	0.6	0.02	98.4
(10,25)	9	20.9	2.26	9	1.0	0.02	123.6
(10,30)	8	58.0	7.19	8	0.9	0.02	360.2
(10,35)	7	68.5	9.75	7	0.8	0.02	496.6
(10,40)	9	110.3	12.20	9	0.9	0.02	557.2
(10,45)	8	157.1	19.58	8	1.1	0.03	705.2
(10,49)	8	146.9	18.28	8	1.4	0.03	586.3
(15,15)	27	79.1	2.39	27	15.0	0.03	83.4
(15,20)	23	104.0	4.06	23	11.9	0.03	152.0
(15,25)	41	253.7	4.27	41	84.4	0.04	104.8
(15,30)	20	168.8	7.93	20	11.1	0.03	235.2
(15,35)	40	541.4	11.04	40	100.2	0.05	244.6
(15,40)	46	844.8	15.14	46	157.9	0.06	253.8
(15,45)	32	608.8	16.55	32	81.2	0.06	296.9
(15,49)	55	1341.6	19.63	55	268.4	0.07	282.0
(20,20)	130	5510.2	4.33	130	4624.7	0.04	98.3
(20,25)	98	4023.5	13.44	98	2628.2	0.06	216.4
(20,30)	108	6090.0	11.26	108	4963.4	0.06	198.7
(20,35)	105	TL	14.45	108	6030.6	0.07	219.8
(20,40)	100	TL	17.94	100	5454.4	0.07	272.5
(20,45)	93	6531.9	17.52	93	5216.0	0.07	238.8
(20,49)	76	TL	24.57	84	TL	0.08	323.8
(25,25)	91	TL	8.63	95	TL	0.05	170.2
(25,30)	84	TL	12.46	91	TL	0.05	237.4
(25,35)	81	TL	11.98	86	TL	0.05	219.2
(25,40)	77	TL	18.63	85	TL	0.06	310.2
(25,45)	69	TL	27.52	76	TL	0.07	385.2
(25,49)	68	TL	31.06	77	TL	0.07	422.4
(30,30)	59	TL	14.24	61	TL	0.05	267.5
(30,35)	50	TL	17.40	53	TL	0.06	306.7
(30,40)	45	TL	18.43	46	TL	0.06	290.6
(30,45)	43	TL	25.60	46	TL	0.06	446.8
(30,49)	40	TL	25.95	43	TL	0.06	424.9
Avg	38.2 [†]	1271.4 [†]	13.40	38.2 [†]	865.4 [†]	0.05	284.4

[†] Averaged across instances where both methods terminated within the time limit.

Table 4: Computational comparison of the algorithm in [17] and the proposed method for the facility location problem.

k	Algorithm in [17]				Proposed method				
	Opt	Gap (%)	t_{tot}^\dagger (s)	$t_{\text{wc}}^{\text{avg}}$ (s)	Opt	Gap (%)	t_{tot}^\dagger (s)	$t_{\text{wc}}^{\text{avg}}$ (s)	$t_{\text{wc}}^{\text{speedup}}$
1	35/35	0.00	15.7	1.33	35/35	0.00	6.3	0.01	114.6
2	35/35	0.00	296.9	4.55	35/35	0.00	202.6	0.02	207.3
3	28/35	1.08	1073.9	9.33	29/35	0.87	816.8	0.04	244.2
4	21/35	5.47	1271.4	13.40	23/35	5.19	865.4	0.05	284.4
	119/140	1.64	569.0	7.15	122/140	1.51	406.4	0.03	240.6

[†] Averaged across instances where both methods terminated within the time limit.

problems (2) and (16); therefore, the speedup in computing worst-case parameter realizations translates only partially to a speedup in the total run time of the algorithm. This is particularly evident for larger instances as shown in Figure 2b. Nevertheless, the decrease in total run time is an encouraging sign. Interestingly, and in contrast to the network design application, Table 3 shows that whenever both methods terminate in the time limit, they execute the same number of column-and-constraint generation iterations. This is likely because the second-stage problem $\mathcal{Q}_I(\mathbf{x}, \boldsymbol{\xi})$ always has a unique optimal solution; indeed, in this case, the KKT-based ($\mathcal{WC}_\mathcal{K}$) and Lagrangian duality-based ($\mathcal{WC}_{\mathcal{L}_T}$) formulations (the latter corresponding to the optimal Lagrange multiplier) are guaranteed to identify the same worst-case parameter realization in every iteration of the algorithm.

Finally, we also compare the computational performance of the Benders decomposition scheme described in Algorithm 6 with the column-and-generation method. Similar to the latter, the former uses Algorithm 5 for computing worst-case parameter realizations. Therefore, any differences in their performance can be attributed primarily to differences in their lower-bounding master problems.

Table 5 summarizes and compares the two methods across the same set of test instances as those reported in Table 4. We observe that for small values of the uncertainty budget parameter $k \in \{1, 2, 3\}$, the column-and-constraint generation algorithm outperforms the Benders decomposition scheme. However, for $k = 4$, the latter slightly outperforms the former, featuring smaller solution times and optimality gaps. This difference in performance is because the number of worst-case parameter realizations grows exponentially with k . Each of these adds additional binary variables to the column-and-constraint generation master problem, making its solution significantly more difficult with each iteration. In contrast, the Benders decomposition scheme adds only a single constraint to its master problem in every iteration. Although this leads to weaker lower bounds and a larger number of iterations in general (as is also evidenced from the table), this trade-off appears to favor the Benders decomposition scheme for large values of k .

5.3 Staff rostering under demand uncertainty

We finally examine the staff rostering problem that was studied in [57]. Consider a service organization that has to assign its regular staff members to shifts to meet unknown service demands, but may also occasionally have to hire part-time staff to meet random demand surges. Let I and J denote the number of regular and part-time staff, respectively; and let T and N denote the number

Table 5: Computational comparison of the Benders decomposition and column-and-constraint generation algorithms for the facility location problem.

k	Benders decomposition				Column-and-constraint generation			
	Opt	# It. [†]	Gap (%)	t_{tot}^{\dagger} (s)	Opt	# It. [†]	Gap (%)	t_{tot}^{\dagger} (s)
1	27/35	356.9	1.80	474.7	35/35	5.7	0.00	1.6
2	28/35	517.2	1.98	671.2	35/35	13.0	0.00	22.6
3	24/35	362.0	2.23	223.7	29/35	25.8	0.87	222.9
4	24/35	507.8	3.57	556.5	23/35	43.9	5.19	1289.5
	103/140	436.8	2.39	488.7	122/140	21.0	1.51	349.8

[†] Averaged across instances where both methods terminated within the time limit.

of shifts (or time periods) in the scheduling horizon and number of work-hours per shift per staff member, respectively. Each shift $t \in [T]$ is associated with an uncertain demand $d_t \in [d_t^0, d_t^0 + \bar{d}_t]$, where $d_t^0, \bar{d}_t \in \mathbb{R}_+$ are the forecasts of the nominal demand and its deviation; and a penalty cost $M_t \in \mathbb{R}_+$ per unit of unmet demand. The set of possible demand realizations is modeled using a cardinality-constrained uncertainty set with parameter $k \geq 0$:

$$\mathcal{D} = \{ \mathbf{d} \in \mathbb{R}_+^T : \exists \boldsymbol{\xi} \in \Xi : \mathbf{d} = \mathbf{d}^0 + \bar{\mathbf{d}} \circ \boldsymbol{\xi} \}, \text{ and } \Xi = \{ \boldsymbol{\xi} \in \{0, 1\}^T : \mathbf{e}^\top \boldsymbol{\xi} \leq k \}.$$

Furthermore, each regular staff member $i \in [I]$ has a fixed wage cost $c_{it} \in \mathbb{R}_+$, and a minimum $l_i \in \mathbb{R}_+$ and maximum $u_i \in \mathbb{R}_+$ number of shifts that they can work over the planning horizon. Similarly, each part-time staff member $j \in [J]$ has a fixed and hourly wage cost of $f_{jt} \in \mathbb{R}_+$ and $h_{jt} \in \mathbb{R}_+$, respectively, and minimum $a_j \in \mathbb{R}_+$ and maximum $b_j \in \mathbb{R}_+$ number of shifts they can work over the planning horizon. The goal is to allocate (before observing any demand) the allocation of regular staff members to shifts, so that the total costs of hiring any additional part-time staff members and penalty costs for unserved demands is minimized under the worst-case demand realization. This problem can be formulated as the following instance of \mathcal{P} .

$$\inf_{\mathbf{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} \mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}),$$

$$\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}) = \left[\begin{array}{l} \text{minimize } \mathbf{c}^\top \mathbf{x} + \mathbf{f}^\top \mathbf{y} + \mathbf{d}^\top \mathbf{u} + \mathbf{M}^\top \mathbf{w} \\ \text{subject to } N \sum_{i \in [I]} x_{it} + \sum_{j \in [J]} v_{jt} + w_t \geq d_t^0 + \bar{d}_t \xi_t, \quad t \in [T] \end{array} \right],$$

where the feasible sets of first- and second-stage decisions are as follows:

$$\mathcal{X} = \left\{ \mathbf{x} \in \{0, 1\}^{I \times T} : \begin{array}{l} x_{it} + x_{i,t+1} + x_{i,t+2} \leq 2, \quad i \in [I], t \in [T-2] \\ l_i \leq \sum_{t \in [T]} x_{it} \leq u_i, \quad i \in [I] \end{array} \right\},$$

$$\mathcal{Y} = \left\{ (\mathbf{y}, \mathbf{v}) \in \{0, 1\}^{J \times T} \times \mathbb{R}_+^{J \times T} : \begin{array}{l} y_{jt} + y_{j,t+1} \leq 1, \quad j \in [J], t \in [T-1] \\ a_j \leq \sum_{t \in [T]} y_{jt} \leq b_j, \quad j \in [J] \\ v_{jt} \leq N y_{jt}, \quad j \in [J], t \in [T] \end{array} \right\}.$$

The first constraints in \mathcal{X} and \mathcal{Y} model the fact that regular and part-time staff members cannot work more than 3 and 2 consecutive shifts, respectively. Note that the second-stage decision y_{jt} is binary and it indicates whether part-time staff member $j \in [J]$ is hired in a particular shift $t \in [T]$ whereas the second-stage decision v_{jt} is continuous and it record the number of hours part-time staff member j is hired to work for in shift t . Therefore, this is an example of a problem featuring mixed-integer second-stage recourse decisions. We refer readers to [57] for additional details about the model formulation.

We use the same experimental setup as [57] and randomly generate 10 instances, each of size $(I, J, T) = (12, 3, 21)$. However, since these are solved relatively fast, we also consider larger instances of twice the size: $(I, J, T) = (24, 6, 42)$. In doing so, we also scale the parameters $\mathbf{l}, \mathbf{u}, \mathbf{a}, \mathbf{b}, \mathbf{d}^0$, and $\bar{\mathbf{d}}$ by a factor of two, but keep all cost parameters unchanged. In [57], the problem is solved using a two-level column-and-constraint generation method, where the upper bounding problem of the inner-level algorithm (see line 12 of Algorithm 8) is solved using a KKT-based mixed-integer bilinear formulation ($\mathcal{WC}_{\mathcal{K}}$) of the second-stage problem (for fixed values of \mathbf{y}). The latter is then further reformulated by linearizing the bilinear terms using problem-specific big-M coefficients and solved using an MILP solver. We benchmark (our implementation) of this method, where the inner-level column-and-constraint generation is solved using Algorithm 8 instead. Since the problem satisfies relatively complete recourse by construction, we skip the computation of the worst-case constraint violations. Also, since the second set of conditions in Theorem 5 are satisfied, we directly compute the optimal Lagrange multiplier using Theorem 4. In all cases, we set a total time limit of 2 hours.

Table 6: Computational comparison of the algorithm in [57] and the proposed method for the staff rostering problem.

(T, k)	Algorithm in [57]				Proposed method				
	Opt	# It.	Gap (%)	t_{tot}^{\dagger} (s)	Opt	# It.	Gap (%)	t_{tot}^{\dagger} (s)	$t_{\text{tot}}^{\text{speedup}, \dagger}$
(12, 3)	10/10	71.3	0.00	9.0	10/10	71.8	0.00	9.3	1.0
(12, 6)	10/10	145.4	0.00	20.5	10/10	147.4	0.00	23.4	0.9
(12, 9)	10/10	119.9	0.00	548.3	10/10	117.5	0.00	300.1	1.8
	30/30	112.2	0.00	192.6	30/30	112.2	0.00	111.0	1.7
(24, 3)	4/10	679.5	0.62	808.8	5/10	679.8	0.51	512.6	1.6
(24, 6)	1/10	556.0	2.17	436.7	2/10	506.0	2.15	183.3	2.4
(24, 9)	1/10	780.0	2.61	1125.5	1/10	771.0	2.69	488.6	2.3
	6/30	675.7	1.80	799.5	8/30	666.0	1.78	453.8	1.8

[†] Averaged across instances where both methods terminated within the time limit.

Table 6 summarizes and compares the computational performance of the algorithm in [57] against our proposed method. The columns # It. and Gap (%) report, respectively, the total number of iterations in the inner-level column-and-constraint generation algorithm (summed across all outer-level iterations) and the optimality gap defined as $100\% \times (UB - LB)/UB$. Thus, the former quantity is equal to the number of times that the upper bounding problem in line 12 of Algorithm 8 is solved. We observe that although the larger class of instances with $T = 24$ shifts is difficult to solve using either method, our proposed algorithm is able to solve more instances and with a smaller optimality gap, with an average speedup (in total run time) of roughly 1.75. Finally,

although not shown in this table, the number of outer-level iterations for $T = 24$ never exceeded 10 for either method and for any value of k . This is because the solution of these instances is limited more by the lower bounding problem (2) in the column-and-constraint generation algorithm, than by the computation of worst-case parameter realizations.

6 Conclusions

This paper presented new Lagrangian dual formulations for two-stage robust optimization problems with binary-valued uncertain data. Their crucial features are the use of appropriately defined penalty functions that move all uncertainties from the constraints to the objective function using a finite scalar-valued Lagrange multiplier. The proposed formulations satisfy strong duality properties even in the presence of discrete first- and second-stage decisions, uncertain objective functions, and in the absence of relatively complete recourse. We showed how they can be used to efficiently calculate worst-case parameter realizations that can be readily integrated in existing high-level exact algorithms. Notably, they circumvent many of the numerical issues that plague existing methods, including smaller MILP formulations and a lack of reliance on decision-independent bounds on dual variables. The proposed methods can exploit problem structure in applications where the binary parameters switch on or off constraints, and they can potentially enable existing algorithms that address uncertainty only in the objective function to solve also problems with uncertainty-affected constraints. Numerical experiments on a diverse set of network design, facility location and staff rostering test instances showed that they can provide computational speedups over existing methods that have been tailored for those specific applications.

We believe that this work can be extended along several directions. First, we need to build extensions of the proposed Lagrangian dual formulations and associated exact methods for problems featuring random recourse or mixed-integer uncertainties (including single-stage problems with mixed-integer decisions). Looking beyond exact algorithms, it would also be interesting to assess whether the proposed ideas can be useful in other contexts as well, including but not limited to approximations schemes for multi-stage robust optimization problems, and worst-case uncertainty quantification for nonlinear models.

References

- [1] Maria Albareda-Sambola, Elena Fernández, and Francisco Saldanha-da Gama. The facility location problem with bernoulli demands. *Omega*, 39(3):335–345, 2011.
- [2] Amir Ardestani-Jaafari and Erick Delage. Linearized robust counterparts of two-stage robust optimization problems with applications in operations management. *INFORMS Journal on Computing*, 2020.
- [3] Ayşe Arslan and Boris Detienne. Decomposition-based approaches for a class of two-stage robust binary optimization problems. *INFORMS Journal on Computing*, 2021.
- [4] Josette Ayoub and Michael Poss. Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Computational Management Science*, 13(2):219–239, 2016.
- [5] Luiz André Barroso, Jimmy Clidaras, and Urs Hölzle. The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Synthesis lectures on computer architecture*, 8(3):1–154, 2013.

- [6] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton university press, 2009.
- [7] Aharon Ben-Tal, Alexander Goryashko, Elana Guslitzer, and Arkadi Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical programming*, 99(2):351–376, 2004.
- [8] Dimitris Bertsimas and Constantine Caramanis. Finite adaptability in multistage linear optimization. *IEEE Transactions on Automatic Control*, 55(12):2751–2766, 2010.
- [9] Dimitris Bertsimas and Iain Dunning. Multistage robust mixed-integer optimization with adaptive partitions. *Operations Research*, 64(4):980–998, 2016.
- [10] Dimitris Bertsimas and Angelos Georghiou. Design of near optimal decision rules in multistage adaptive mixed-integer optimization. *Operations Research*, 63(3):610–627, 2015.
- [11] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.
- [12] Hans-Georg Beyer and Bernhard Sendhoff. Robust optimization—a comprehensive survey. *Computer methods in applied mechanics and engineering*, 196(33-34):3190–3218, 2007.
- [13] Daniel Bienstock and Abhinav Verma. The $N - k$ Problem in Power Grids: New Models, Formulations, and Numerical Experiments. *SIAM Journal on Optimization*, 20(5):2352–2380, 2010.
- [14] Christoph Buchheim and Jannis Kurtz. Min–max–min robust combinatorial optimization. *Mathematical Programming*, 163(1-2):1–23, 2017.
- [15] Constantine Caramanis, Shie Mannor, and Huan Xu. 14 robust optimization in machine learning. *Optimization for machine learning*, page 369, 2012.
- [16] Xin Chen and Yuhan Zhang. Uncertain linear programs: Extended affinely adjustable robust counterparts. *Operations Research*, 57(6):1469–1482, 2009.
- [17] Chun Cheng, Yossiri Adulyasak, and Louis-Martin Rousseau. Robust facility location under demand uncertainty and facility disruptions. *Omega*, 103:102429, 2021.
- [18] Antonio J Conejo and Xuan Wu. Robust optimization in power systems: a tutorial overview. *Optimization and Engineering*, pages 1–23, 2021.
- [19] Jing Dong and Rouba Ibrahim. Managing supply in the on-demand economy: Flexible workers, full-time employees, or both? *Operations Research*, 68(4):1238–1264, 2020.
- [20] Iain Dunning, Joey Huchette, and Miles Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.
- [21] Frank J Fabozzi, Petter N Kolm, Dessislava A Pachamanova, and Sergio M Focardi. *Robust portfolio optimization and management*. John Wiley & Sons, 2007.
- [22] Wei Feng, Yi Zhang, Gang Rong, and Yiping Feng. Finite adaptability in data-driven robust optimization for production scheduling: a case study of the ethylene plant. *Industrial & Engineering Chemistry Research*, 58(16):6505–6518, 2019.

- [23] Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014.
- [24] Francis Garuba, Marc Goerigk, and Peter Jacko. A comparison of discrete and polyhedral uncertainty sets for robust network design. *arXiv preprint arXiv:2003.10507*, 2020.
- [25] Angelos Georghiou, Daniel Kuhn, and Wolfram Wiesemann. The decision rule approach to optimization under uncertainty: methodology and applications. *Computational Management Science*, 16(4):545–576, 2019.
- [26] Fred Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975.
- [27] Marc Goerigk, Jannis Kurtz, and Michael Poss. Min–max–min robustness for combinatorial problems with discrete budgeted uncertainty. *Discrete Applied Mathematics*, 285:707–725, 2020.
- [28] Selcuk Goren and Ihsan Sabuncuoglu. Robustness and stability measures for scheduling: single-machine environment. *IIE Transactions*, 40(1):66–83, 2008.
- [29] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2021.
- [30] Grani A Hanasusanto and Daniel Kuhn. Conic programming reformulations of two-stage distributionally robust linear programs over wasserstein balls. *Operations Research*, 66(3):849–869, 2018.
- [31] Grani A Hanasusanto, Daniel Kuhn, and Wolfram Wiesemann. K-adaptability in two-stage robust binary programming. *Operations Research*, 63(4):877–891, 2015.
- [32] Hossein Hashemi Doulabi, Patrick Jaillet, Gilles Pesant, and Louis-Martin Rousseau. Exploiting the structure of two-stage robust optimization models with exponential scenarios. *INFORMS Journal on Computing*, 33(1):143–162, 2021.
- [33] Rabih A Jabr, Izudin Džafić, and Bikash C Pal. Robust optimization of storage investment on transmission networks. *IEEE Transactions on Power Systems*, 30(1):531–539, 2014.
- [34] Ruiwei Jiang, Minseok Ryu, and Guanglin Xu. Data-driven distributionally robust appointment scheduling over wasserstein balls. *arXiv preprint arXiv:1907.03219*, 2019.
- [35] Ruiwei Jiang, Siqian Shen, and Yiling Zhang. Integer programming approaches for appointment scheduling with random no-shows and service durations. *Operations Research*, 65(6):1638–1656, 2017.
- [36] Nicolas Kämmerling and Jannis Kurtz. Oracle-based algorithms for binary two-stage robust optimization. *Computational Optimization and Applications*, 77(2):539–569, 2020.
- [37] Hervé Kerivin and A Ridha Mahjoub. Design of survivable networks: A survey. *Networks: An International Journal*, 46(1):1–21, 2005.
- [38] Arie MCA Koster, Manuel Kutschka, and Christian Raack. Robust network design: Formulations, valid inequalities, and computations. *Networks*, 61(2):128–149, 2013.
- [39] Panos Kouvelis and Gang Yu. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 2013.

- [40] Hongqiang Harry Liu, Srikanth Kandula, Ratul Mahajan, Ming Zhang, and David Gelernter. Traffic engineering with forward fault correction. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, pages 527–538, 2014.
- [41] Mengshi Lu, Lun Ran, and Zuo-Jun Max Shen. Reliable facility location design under uncertain correlated disruptions. *Manufacturing & Service Operations Management*, 17(4):445–455, 2015.
- [42] Logan R Matthews, Chrysanthos E Gounaris, and Ioannis G Kevrekidis. Designing networks with resiliency to edge failures using two-stage robust optimization. *European Journal of Operational Research*, 279(3):704–720, 2019.
- [43] Areesh Mittal, Can Gokalp, and Grani A Hanasusanto. Robust quadratic programming with mixed-integer uncertainty. *INFORMS Journal on Computing*, 32(2):201–218, 2020.
- [44] Tri-Dung Nguyen and Andrew W Lo. Robust ranking and portfolio optimization. *European Journal of Operational Research*, 221(2):407–416, 2012.
- [45] Krzysztof Postek and Dick den Hertog. Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing*, 28(3):553–574, 2016.
- [46] Soroosh Shafieezadeh-Abadeh, Peyman Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. *arXiv preprint arXiv:1509.09259*, 2015.
- [47] J Cole Smith, Mike Prince, and Joseph Geunes. Modern network interdiction problems and algorithms. In *Handbook of combinatorial optimization*, pages 1949–1987. Springer New York, 2013.
- [48] Anirudh Subramanyam. *Robust optimization of vehicle routing problems under uncertainty*. PhD thesis, Carnegie Mellon University, 2018.
- [49] Anirudh Subramanyam, Chrysanthos E Gounaris, and Wolfram Wiesemann. K-adaptability in two-stage mixed-integer robust optimization. *Mathematical Programming Computation*, 12(2):193–224, 2020.
- [50] Anirudh Subramanyam, Frank Mufalli, José M Laínez-Aguirre, Jose M Pinto, and Chrysanthos E Gounaris. Robust multiperiod vehicle routing under customer order uncertainty. *Operations Research*, 69(1):30–60, 2021.
- [51] Anirudh Subramanyam, Mohamed El Tonbari, and Kibaek Kim. Data-driven two-stage conic optimization with zero-one uncertainties. *arXiv preprint arXiv:2001.04934*, 2020.
- [52] Aurélie Thiele, Tara Terry, and Marina Epelman. Robust linear optimization with recourse. Technical report, Lehigh University and University of Michigan, 2009.
- [53] Zidong Wang, Fuwen Yang, Daniel W. C. Ho, and Xiaohui Liu. Robust h_∞ control for networked systems with random packet losses. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(4):916–924, 2007.
- [54] Guanglin Xu and Samuel Burer. A copositive approach for two-stage adjustable robust optimization with uncertain right-hand sides. *Computational Optimization and Applications*, 70(1):33–59, 2018.

- [55] Lei Xu and Tsan Sheng Adam Ng. A robust mixed-integer linear programming model for mitigating rail transit disruptions under uncertainty. *Transportation Science*, 54(5):1388–1407, 2020.
- [56] Wei Yuan, Jianhui Wang, Feng Qiu, Chen Chen, Chongqing Kang, and Bo Zeng. Robust optimization-based resilient distribution network planning against natural disasters. *IEEE Transactions on Smart Grid*, 7(6):2817–2826, 2016.
- [57] Bo Zeng and Long Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461, 2013.
- [58] Long Zhao and Bo Zeng. An exact algorithm for two-stage robust optimization with mixed integer recourse problems. Technical report, University of South Florida, 2012.
- [59] Jianzhe Zhen, Dick Den Hertog, and Melvyn Sim. Adjustable robust optimization via fourier–motzkin elimination. *Operations Research*, 66(4):1086–1100, 2018.

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).