Uncertainty reduction in robust optimization

Ayşe N. Arslan[a], Michael Poss[b]

[a] Centre Inria de l'Universite de Bordeaux, F-33405 Talence, France
Univ. Bordeaux, CNRS, INRIA, Bordeaux INP, IMB, UMR 5251, F-33400 Talence, France
[b] LIRMM, University of Montpellier, CNRS, France

Abstract

Uncertainty reduction has recently been introduced in the robust optimization literature as a relevant special case of decision-dependent uncertainty. Herein, we identify two relevant situations in which the problem is polynomially solvable. We further provide insights into possible MILP reformulations and the strength of their continuous relaxations.

Keywords: combinatorial optimization, robust optimization, NP-hardness, reformulation

1. Introduction

In this paper, we are interested in robust optimization problems of the form:

$$\begin{aligned} \min_{y\in Y} \quad & f^\top y && \text{(Static-Robust)}\\ \text{s.t.} \quad & H(\xi)y \le g && \forall \xi \in \Xi \end{aligned}$$

where the set $Y \subseteq \mathbb{R}^n$ defines the deterministic structure of solutions y and may incorporate integrality restrictions, $\Xi \subseteq \mathbb{R}^q$ is a polytope, $H(\xi)$ for $\xi \in \Xi$, f and g are real matrices and real vectors of conforming dimensions, respectively. We assume that, all uncertain parameters are affine functions of $\xi \in \Xi$. Using well-known reformulation techniques (Static-Robust) encompasses the cases where f and g may depend affinely on ξ.

(Static-Robust) is typically well-solved by using classical reformulation techniques based on linear programming duality [2]. However, it does not model applications in which it is possible for the decision-maker to take some proactive actions to reduce uncertainty. As such, Nohadani and Sharma [12] introduced decision-dependent polyhedral uncertainty sets that model uncertainty reduction, defined as follows:

$$\Xi(x) = \left\{\xi \in \mathbb{R}^q_+ \;\middle|\; D\xi \le d, \xi \le v + w \circ (e - x)\right\}, \qquad (1)$$

where $v, w \in \mathbb{R}^q_+$, $x \in X \subseteq \{0,1\}^q$ is a binary vector, and e is the vector of all ones.

In (1), x is a decision variable that controls the upper bounds of uncertain parameters. When $x_i = 0$, the uncertain variable ξ_i can be as large as $v_i + w_i$, whereas when $x_i = 1$ its value reduces to v_i. We write the decision-dependent robust uncertainty reduction problem as:

$$\begin{aligned} \min_{x\in X\subseteq\{0,1\}^q, y\in Y} \quad & c^\top x + f^\top y && \text{(UR-Robust)}\\ \text{s.t.} \quad & Ax + H(\xi)y \le g && \forall \xi \in \Xi(x), \end{aligned}$$

where $c \in \mathbb{R}^q_+$ denotes the cost of reducing the uncertainty. We also dedicate a particular interest to the min-max combinatorial variant of the above robust problem with binary optimization variables y and only objective uncertainty. We write

$$\min_{x\in X\subseteq\{0,1\}^q, y\in Y\subseteq\{0,1\}^n} \max_{\xi\in\Xi(x)} c^\top x + (f+\xi)^\top y, \qquad \text{(UR-Min-Max)}$$

where $q = n$. Clearly, problem (UR-Min-Max) is at least as hard as optimizing a linear function over X or Y, so the problem is NP-hard in general.

The first mention of decision-dependent uncertainty sets in the robust optimization literature dates back to [17] where the authors use its expressive power to better model the application at hand, specifically, a software partitioning problem involving multiple instantiations. The notion has also been used by [14, 15], who show how the use of decision-dependent budgets can reduce the conservatism of the so-called budgeted uncertainty set [3], sometimes at no extra computational cost. In yet another context, [7] rely on decision-dependent uncertainty sets to model K-adaptable policies, wherein variables x allow to partition set Ξ optimally. The authors of [12] introduce the uncertainty reduction model (UR-Robust), for which they propose different formulations as well as detailed numerical experiments that illustrate the possible impact of uncertainty reduction. They additionally consider MILP reformulations and a hardness proof for robust optimization problems with a more general decision-dependent uncertainty set structure. More recently, [18] has extended the scope of decision-dependent uncertainty sets to two-stage robust optimization problems, proposing different decomposition algorithms.

In this paper, we first concentrate on (UR-Min-Max) and study its theoretical complexity. In Section 2, we identify two cases where (UR-Min-Max) is essentially as hard as its deterministic counterpart

$$\min_{y\in Y\subseteq\{0,1\}^n} \tilde{f}^\top y, \qquad \text{(Combinatorial)}$$

under the simplifying assumption that D has a single row. We also briefly discuss how the results extend to more general un-

Email addresses: `ayse-nur.arslan@inria.fr` (Ayşe N. Arslan), `michael.poss@lirmm.fr` (Michael Poss)

certainty sets as well as problems with multiple robust constraints. Then, in Section 3, we consider the more general model (UR-Robust) for which we generalise a previous MILP formulation proposed by [12] and develop a new reformulation in the case where $D \geq 0$ and show that its linear relaxation can be arbitrarily better than the former for certain problems. Finally, we numerically illustrate our theoretical result in Section 4 on the shortest path instances described by [12] and compare it to the reformulations proposed therein. We also show that the linear programming relaxations of all formulations coincide for these specific instances. We close with some conclusions in Section 5.

2. Polynomial-time special cases

In this section, we focus on (UR-Min-Max) (so $q = n$) and assume, for ease of exposition, that, $v_j = 0$ for $j \in [n]$, *i.e.*, the parameter ξ_j for $j \in [n]$ can be completely reduced to 0 by setting variable $x_j = 1$ and that the uncertainty set has a single constraint with D a row vector and d a scalar. In this setting, we show that the problem is polynomially solvable if one of the following holds:

1. $X = \{0,1\}^n$ and (Combinatorial) is polynomially solvable for any vector $\tilde{f}$.
2. $D \geq 0$, and the formulations for X and Y satisfy a specific integrality property.

The results of this section can be extended to uncertainty sets with multiple constraints [13, 16]. They can similarly be extended to multiple constraints affected by uncertainty [1]. However, the complexity of the resulting algorithms will be exponential in the number of constraints of the uncertainty set as well as the number of constraints affected by uncertainty. This is in line with the NP-hardness of the robust counterpart of polynomially solvable combinatorial optimization problems for arbitrary uncertainty polytopes, e.g. [4].

We start by reformulating the inner maximization problem using linear programming duality to obtain:

$$\min_{\substack{x\in X\subseteq\{0,1\}^n, y\in Y\subseteq\{0,1\}^n,\\ \theta\geq 0, \pi\in\mathbb{R}^n_+}} c^\top x + f^\top y + d\theta + \sum_{j\in[n]} w_j(1-x_j)\pi_j \quad \text{(UR-Min)}$$
$$\text{s.t.} \quad D_j\theta + \pi_j \geq y_j \qquad \forall j \in [n].$$

Furthermore, we introduce the set $\Theta = \{0\} \cup \left\{1/D_j \;\middle|\; j \in [n] : D_j > 0\right\}$, which contains at most $n+1$ elements.

Proposition 1. *For each $\theta \in \Theta$, consider the optimization problem*

$$z(\theta) = \min_{x\in X\subseteq\{0,1\}^n, y\in Y\subseteq\{0,1\}^n} c^\top x + f^\top y + \sum_{j\in[n]} w_j(1-x_j)\left([1-D_j\theta]^+ y_j + [-D_j\theta]^+(1-y_j)\right). \quad (2)$$

and denote its optimal solution by $(x(\theta), y(\theta))$. Solving problem (UR-Min) *is equivalent to solving* (2) *for each $\theta \in \Theta$, and returning $(x(\theta), y(\theta))$ that minimizes $d\theta + z(\theta)$.*

Proof. Starting from (UR-Min), let $\theta \geq 0$ be fixed. Then, the expression of π_j for $j \in [n]$ simplifies to $[y_j - D_j\theta]^+$. We next plug this expression into the objective function of (UR-Min) to obtain:

$$\min_{\substack{x\in X\subseteq\{0,1\}^n, y\in Y\subseteq\{0,1\}^n,\\ \theta\geq 0}} c^\top x + f^\top y + d\theta + \sum_{j\in[n]} w_j(1-x_j)[y_j - D_j\theta]^+ \quad (3)$$

which can be solved by searching over possible values of θ. Further, since y_j is binary, it can be taken out of $[\cdot]^+$, giving rise to

$$\min_{\substack{x\in X\subseteq\{0,1\}^n, y\in Y\subseteq\{0,1\}^n,\\ \theta\geq 0}} c^\top x + f^\top y + d\theta + \sum_{j\in[n]} w_j(1-x_j)\left([1-D_j\theta]^+ y_j + [-D_j\theta]^+(1-y_j)\right). \quad (4)$$

We remark that for fixed x and y, because $w_j \geq 0$ for $j \in [n]$, the objective function of the above problem is a positive-weighted sum of piecewise affine convex functions of θ and is therefore piecewise affine convex in θ. Its minimum can be obtained as one of the breakpoints of the piecewise affine functions which are obtained either at $\theta = 0$ when $D_j \leq 0$ or at $\theta = \frac{1}{D_j}$ when $D_j > 0$. As such, (UR-Min) can be solved as a series of problems, each time with a fixed value of θ.

□

We next provide two relevant special cases in which the bilinear reformulation (2), and therefore problem (UR-Min), can be solved in polynomial time. We first consider the case where X is unconstrained.

Corollary 1. *If $X = \{0,1\}^n$, an optimal solution of* (UR-Min) *can be obtained by solving at most $n+1$ deterministic problems of the same form as* (Combinatorial).

Proof. Let us rewrite (2) a little differently in order to better reveal its combinatorial structure:

$$\sum_{j\in[n]} w_j[-D_j\theta]^+ \quad (5)$$
$$+ \min_{y\in Y\subseteq\{0,1\}^n} \left\{ \sum_{j\in[n]} \left(f_j + w_j[1-D_j\theta]^+ - w_j[-D_j\theta]^+\right) y_j \right.$$
$$\left. + \min_{x\in\{0,1\}^n} c^\top x - \sum_{j\in[n]} w_j x_j \left([1-D_j\theta]^+ y_j + [-D_j\theta]^+(1-y_j)\right) \right\}.$$

Assume, now, that the vector y is fixed. Then the optimization problem over the vector x decomposes over its elements, the problem over x_j reading

$$\min_{x_j\in\{0,1\}} \left[c_j - w_j\left([1-D_j\theta]^+ y_j + [-D_j\theta]^+(1-y_j)\right)\right] x_j. \quad (6)$$

When $y_j = 1$, we obtain the optimal value of this problem as $\left[c_j - w_j[1-D_j\theta]^+\right]^-$, otherwise we obtain it as $\left[c_j - w_j[-D_j\theta]^+\right]^-$. As such, its optimal value can be written linearly in y as

$$= \left[c_j - w_j[1-D_j\theta]^+\right]^- y_j + \left[c_j - w_j[-D_j\theta]^+\right]^- (1-y_j). \quad (7)$$

Now integrating into the outer optimization problem over y, we obtain:

$$\sum_{j\in[n]} w_j[-D_j\theta]^+ + \left[c_j - w_j[-D_j\theta]^+\right]^- \tag{8}$$
$$+ \min_{y\in Y\subseteq\{0,1\}^n} \sum_{j\in[n]} \Big(f_j + w_j[1-D_j\theta]^+ - w_j[-D_j\theta]^+$$
$$+ \left[c_j - w_j[1-D_j\theta]^+\right]^- - \left[c_j - w_j[-D_j\theta]^+\right]^-\Big) y_j.$$

This problem is in the same form as (Combinatorial) which, together with Proposition 1, completes the proof. □

Next, we consider the case where X is a subset of $\{0,1\}^n$. Specifically, let us introduce the formulations $X = \{x \in \mathbb{Z}^n \mid A^x x \le b^x\}$ and $Y = \{y \in \mathbb{Z}^n \mid A^y y \le b^y\}$, and introduce further the matrix

$$A' = \begin{pmatrix} A^x & 0 \\ 0 & A^y \\ Id & -Id \end{pmatrix}.$$

Our next result requires that A' be TU, which implies that A^x and A^y need be TU matrices as well.

Corollary 2. *If $D \ge 0$ and A' is totally unimodular, an optimal solution of* (UR-Min) *can be obtained by solving at most $n+1$ linear programs with constraint matrix A'.*

Proof. Assuming $D_j \ge 0$, (2) can be reformulated as

$$\min_{x\in X, y\in Y} c^\top x + \tilde{f}(\theta)^\top y - \sum_{j\in[n]} w_j[1-D_j\theta]^+ x_j y_j, \tag{9}$$

where $\tilde{f}_j(\theta) = f_j + w_j[1-D_j\theta]^+$. Observe that because $c_j \ge 0$, in any optimal solution to (9), $x_j = 1$ implies $y_j = 1$, so that any optimal solution satisfies $x_j \le y_j$. Thus, the optimal solutions to (9) coincide with the optimal solutions to

$$\min_{\substack{x\in X, y\in Y\\ x\le y}} c^\top x + \tilde{f}(\theta)^\top y - \sum_{j\in[n]} w_j[1-D_j\theta]^+ x_j y_j. \tag{10}$$

Then, notice that for any (x,y) that satisfies $x \le y$, $x_j y_j = x_j$ for each $j \in [n]$. Thus, introducing $\tilde{c}_j(\theta) = c_j - w_j[1-D_j\theta]^+$ for each $j \in [n]$, (10) is, under the stated assumptions, equal to the following integer linear program:

$$\begin{aligned} \min \quad & \tilde{c}(\theta)^\top x + \tilde{f}(\theta)^\top y \\ \text{s.t.} \quad & A^x x \le b^x \\ & A^y y \le b^y \\ & x \le y \\ & (x,y) \in \mathbb{Z}^{2n}. \end{aligned}$$

The constraint matrix of the above integer liner program is

$$\begin{pmatrix} A^x & 0 \\ 0 & A^y \\ Id & -Id \end{pmatrix},$$

and the result follows again from Proposition 1. □

Observe that whenever the number of constraints of A' is polynomially bounded by n, Corollary 2 implies that (UR-Min) is polynomially solvable. While verifying whether matrix A' is TU is not an easy task in general, we prove below that this is the case when Y describes the selection problem and X restricts the number of uncertain parameters that can be reduced to k. We mention that the selection problem is often considered in the combinatorial robust optimization literature [6].

Proposition 2. *Let $X = \left\{x \in \mathbb{Z}^n \;\middle|\; \sum_{j\in[n]} x_j \le k\right\}$ and $Y = \left\{y \in \mathbb{Z}^n \;\middle|\; \sum_{j\in[n]} y_j \ge p\right\}$ for given integers $k, p \in [n]$. Then, the resulting matrix*

$$A' = \begin{pmatrix} 1\dots1 & 0\dots0 \\ 0\dots0 & 1\dots1 \\ Id & -Id \end{pmatrix}$$

is TU.

Proof. We prove the result using Ghouila-Houri's theorem. Let C be any subset of columns of A'. The objective is to partition C into S and T such that the condition

$$\sum_{j\in S} A'_{ij} - \sum_{j\in T} A'_{ij} \in \{-1,0,1\} \tag{11}$$

is satisfied for each row i of A'. We introduce next the subsets $C^x = C \cap \{1,\dots,n\}$ and $C^y = C \cap \{n+1,\dots,2n\}$ as well as $\tilde{C}^x = \{i \in C^x \mid i+n \in C^y\}$ and $\tilde{C}^y = \{i \in C^y \mid i-n \in C^x\}$, and let $\tilde{C} = \tilde{C}^x \cup \tilde{C}^y$.

We first concentrate on the elements of $\tilde{C}$ and remark that, for these columns, condition (11) written for the third set of constraints immediately implies that columns $i \in \tilde{C}^x$ and $i+n \in \tilde{C}^y$ should either both be in S or both be in T. Further, in order to satisfy (11) for the first two rows, we must have that $|S\cap C^x| - |T \cap C^x| \in \{0,1,-1\}$ (analogously for $|S\cap C^y|$ and $|T\cap C^y|$). Thus, a partition of $\tilde{C}$ respecting both conditions puts any subset of $\lceil \tilde{C}^x/2\rceil$ indices in $\tilde{C}^x$ into S, and the remaining $\lfloor \tilde{C}^x/2\rfloor$ indices in $\tilde{C}^x$ into T, then proceeds to putting $i+n$ in S for $i \in S$, and $i+n$ in T for $i \in T$, respectively, covering hence $\tilde{C}^y$.

We now extend the constructed partition to $C' = C \setminus \tilde{C}$. Notice that for C', the condition (11) written for the third group of constraints of A' are always satisfied, because the corresponding rows of A' restricted to the columns of C' contain only one element different from zero. Therefore, elements of C' can be added gradually into S and T so as to satisfy $|S\cap C^x| - |T\cap C^x| \in \{0,1,-1\}$ (analogously for $|S\cap C^y|$ and $|T\cap C^y|$). □

The results of this section extend to the general case where $v \ne 0$ expressed by (1). To do so, it suffices to introduce the uncertain parameters ξ^1_j and ξ^2_j for ξ_j, $j \in [n]$ to obtain:

$$\Xi^{\text{lifted}}(x) = \left\{ \xi, \xi^1, \xi^2 \in \mathbb{R}^n_+ \;\middle|\; \begin{array}{l} D\xi \le d \\ \xi = \xi^1 + \xi^2 \\ \xi^1 \le v \\ \xi^2 \le w \circ (e - x) \end{array} \right\}$$

which recovers the set $\Xi(x)$ by projecting out the variables ξ_1, ξ_2. Further, by projecting out the variables ξ, we obtain:

$$\bar{\Xi}(x) = \left\{\xi^1, \xi^2 \in \mathbb{R}^n_+ \;\middle|\; D(\xi^1 + \xi^2) \le d, \xi^1 \le v, \xi^2 \le w \circ (e - x)\right\}.$$

We remark that, in the set $\bar{\Xi}(x)$, reducible uncertain parameters ξ^2_j for $j \in [n]$, are completely reduced to 0 when $x_j = 1$ in line with our initial assumption. The preceding developments then can be repeated in the same manner, the only difference being the addition of the term $v_j[1 - D_j\theta]^+ - v_j[-D_j\theta]^+$ to the coefficient of each variable y_j.

Remark 1. *The dependence of the uncertainty set on the decision variables is here motivated by uncertainty reduction. However, it has been suggested in the literature to let, additionally, the right-hand-side vector d depend on y, motivated, for instance, by the probabilistic guarantees of the budgeted uncertainty set [3, 14, 15]. In particular, combining Theorem 3 from [15] with Theorem 1 still leads to solving $n + 1$ deterministic problems of the form* (Combinatorial) *whenever d depends affinely on y.*

3. Reformulations

In this section, we focus on reformulations of (UR-Robust) in the case where $D \ge 0$. We consider a single robust constraint written as:

$$a_i^\top x + h_i(\xi)^\top y \le g_i \qquad \forall \xi \in \Xi(x), \tag{12}$$

where a_i and $h_i(\xi)$ for $\xi \in \Xi$ are the i^{th} row of matrices A and $H(\xi)$ for $\xi \in \Xi$, respectively. Since $h_i(\xi)$ is an affine function of ξ, we can express it as $h_i(\xi) = \bar{h}_i + \bar{H}_i \xi$ where $\bar{h}_i$ and $\bar{H}_i$ are of conforming dimensions.

Constraint (12) is a semi-infinite constraint that is commonly treated in robust optimization using classical reformulation techniques based on linear programming duality. To do so, we write it equivalently as:

$$\max_{\xi \in \Xi(x)} \xi^\top \bar{H}_i^\top y \le g_i - a_i^\top x - \bar{h}_i^\top y \tag{13}$$

integrating the definition of $h_i(\xi)$ and gathering the constant terms (in ξ) on the right-hand-side of the constraint. Then, using classical linear programming duality arguments, we obtain the deterministic equivalent expression of constraint (12) as the system of constraints:

$$\sigma^\top d + \pi^\top (v + w \circ (e - x)) \le g_i - a_i^\top x - \bar{h}_i^\top y \tag{14}$$

$$D^\top \sigma + \pi \ge \bar{H}_i^\top y \tag{15}$$

$$\pi, \sigma \ge 0, \tag{16}$$

where σ and π are dual variables corresponding, respectively, to constraints $D\xi \le d$ and $\xi \le v + w \circ (e - x)$. Although deterministic, this reformulation is nonlinear due to the presence of the term $\pi^\top (v + w \circ (e - x))$ which involves the multiplication between variables π and x. Since variables x are assumed to be binary these terms can be linearized using the big-M technique where the big-M should be tailored based on upper bounds of dual variables π. The linear relaxation of such a formulation can be quite weak. Nevertheless, Nohadani and Sharma [12] proposed such a formulation for the general case of decision-dependent uncertainty sets where they additionally discussed conditions under which the upper bound constraints can be removed (which they call the modified big-M formulation).

The authors additionally proposed a formulation for (UR-Robust), which they called the $\bar{\Pi}$ formulation, where the decision-dependence of (1) can be transferred to the objective function of the problem through a big-M coefficient. Their result was stated in the case where $\bar{H}_i = I$ for $i \in [m]$. We generalize it here to any $\bar{H}_i \in R^q$.

Proposition 3. *We have that*

$$\max_{\xi \in \Xi(x)} \xi^\top \bar{H}_i^\top y = \max_{\xi^1, \xi^2 \in \bar{\Xi}(0)} y^\top \bar{H}_i \xi^1 + (\bar{H}_i^\top y - \bar{\Pi} x)^\top \xi^2 \tag{17}$$

where $\bar{\Pi} = \mathrm{diag}(\pi^{\max})$ is a diagonal matrix with π^{max} a vector of component-wise upper bounds on dual variables π in (14)-(16)*.*

In (17), $\bar{\Pi}$ acts as a big-M coefficient so that when $x_j = 1$ the corresponding uncertain parameter ξ^2_j is equal to zero at optimality while the uncertainty set now becomes independent of decisions x. Nohadani and Sharma [12] further prove that when $\bar{H}_i = I$ for $i \in [m]$ and $D, y \ge 0$ this upper bound can be estimated using the upper bounds on variables y. We next generalize this result to any $\bar{H}_i \in R^q$ further removing the assumption that $y \ge 0$:

Proposition 4. *If $D \ge 0$ then $\pi^{\max}_j$ for $j \in [q]$ can be set to:*

$$\max \left\{0, \max_{y \in Y} \quad (\bar{H}_i^\top y)^\top e_j\right\}. \tag{18}$$

The proofs of Proposition 3 and 4 proceed very similarly to those presented in [12]. Proposition 3 provides a way to obtain a linear formulation for (UR-Robust), and Proposition 4 provides a way to estimate the big-M coefficients necessary for this formulation based on the knowledge of the primal problem. However, in order to calculate these coefficients one might need to solve n, potentially difficult, optimization problems.

We next propose an alternative formulation that allows us to better capitalize on the knowledge of the primal formulation. To this end, we remark that in $\bar{\Xi}(x)$ each uncertain parameter ξ^2_j is bounded by $w_j \ge 0$ when $x_j = 0$, and by 0 when $x_j = 1$. Since ξ^2_j is also lower-bounded by 0, this implies that $\xi^2_j = 0$ when $x_j = 1$, *i.e.*, the effect of ξ^2_j is completely eliminated from the constraints. The following result closely follows from this observation.

Theorem 1. *If $D \ge 0$, constraint* (12) *can be equivalently expressed as:*

$$a_i^\top x + h_i(\xi^1 + (e - x) \circ \xi^2)^\top y \le g_i \qquad \forall \xi^1, \xi^2 \in \bar{\Xi}(0). \tag{19}$$

Proof. The proof consists in showing that (19) has at least one worst case realization ξ^1, ξ^2 such that $\xi^2_j = 0$ whenever $x_j = 1$ for $j \in [q]$. To this end, consider the optimization problem

$$\max_{\xi^1, \xi^2 \in \bar{\Xi}(0)} \quad h_i(\xi^1 + (e - x) \circ \xi^2)^\top y \tag{20}$$

and let (ξ^1, ξ^2) be an optimal solution such that there exists $k \in [q]$ with $\xi^2_k = \epsilon > 0$ and $x_k = 1$. Construct now the solution $(\bar{\xi}^1, \bar{\xi}^2)$ such that $\bar{\xi}^1 = \xi^1$, $\bar{\xi}^2_j = \xi^2_j$ for $j \in [q]\backslash\{k\}$ and $\bar{\xi}^2_k = \xi^2_k - \epsilon$. Clearly, $0 \leq \bar{\xi}^1 \leq v$ and $0 \leq \bar{\xi}^2 \leq w$. Further,

$$D(\bar{\xi}^1 + \bar{\xi}^2) = D(\xi^1 + \xi^2 - \epsilon e_k) \leq d - \epsilon D e_k \leq d$$

where the last inequality holds since we assume that $D \geq 0$. The feasible solution $(\bar{\xi}^1, \bar{\xi}^2)$ has additionally the same objective value as the solution (ξ^1, ξ^2) since the objective coefficient of variable ξ^2_k is equal to zero when $x_k = 1$ which concludes the proof. □

As a result of Theorem 1, the uncertainty set becomes free of variables x whereas the effect of uncertainty reduction is accounted for through the term $(e - x) \circ \xi_2$. We remark that the condition $D \geq 0$ is necessary for Theorem 1 to hold since otherwise the value of ξ^2_j can be increased in order to increase the value of another uncertain parameter even when $x_j = 1$ which results in (12) and (19) no longer being equivalent.

In order to proceed with the derivation of our reformulation, we first write constraints (19) equivalently as:

$$\max_{\xi^1, \xi^2 \in \bar{\Xi}(0)} \quad (\xi^1 + (e - x) \circ \xi^2)^\top \bar{H}_i^\top y \leq g_i - a_i^\top x - \bar{h}_i^\top y. \quad (21)$$

Then, using linear programming duality, we obtain the system of inequalities:

$$\pi^\top d + q^\top v + r^\top w \leq g_i - a_i^\top x - \bar{h}_i^\top y \quad (22)$$
$$D^\top \pi + q \geq \bar{H}_i^\top y \quad (23)$$
$$D^\top \pi + r \geq \bar{H}_i^\top y - \bar{H}_i^\top (y \circ x) \quad (24)$$
$$\pi, q, r \geq 0, \quad (25)$$

which can then be linearized using the big-M technique in order to eliminate the bilinear terms $y \circ x$ where the big-M should be tailored based on lower and upper bounds of variables y which can be deduced from the knowledge of the problem. This formulation is advantageous compared to the previous one especially when y are binary in which case the big-M coefficient can be set to 1. We next illustrate this point with an example.

Example 1. *Consider the "box" uncertainty set* $\Xi(x) = \left\{\xi \in \mathbb{R}^q_+ \;\middle|\; \xi \leq e - x\right\}$ *for the robust problem*

$$z = \min_{x \in X \subseteq \{0,1\}^q, y \in Y} \max_{\xi \in \Xi(x)} \quad y^\top \bar{H} \xi, \quad (26)$$

where we assume, for simplicity, that $y^\top \bar{H} \geq 0$ *for* $y \in Y$. *On the one hand, the alternative approach proposed in Theorem 1 reformulates* (26) *as* $\min_{x \in X \subseteq \{0,1\}^q, y \in Y} y^\top \bar{H}(e - x)$ *since* $\Xi(0) = [0,1]^q$ *and* $y^\top \bar{H} \geq 0$ *for* $y \in Y$ *by assumption. The linear programming relaxation of the linearization of this formulation is*

$$\begin{aligned} z^{\text{new}}_{LR} = \min_{x \in rel(X), y \in rel(Y)} & \sum_{j \in [n], k \in [q]} \bar{H}_{jk}(y_j - \eta_{jk}) \\ s.t. \quad & \eta_{jk} \leq x_k && \forall j \in [n], \forall k \in [q] \\ & \eta_{jk} \leq y_j && \forall j \in [n], \forall k \in [q] \\ & \eta_{jk} \geq x_k + y_j - 1 && \forall j \in [n], \forall k \in [q] \\ & \eta \geq 0, \end{aligned}$$

where $rel(P)$ *denotes the linear programming relaxation of formulation P. On the other hand, the linear programming relaxation obtained through Proposition 3 is*

$$z^{\bar{\Pi}}_{LR} = \min_{x \in rel(X), y \in rel(Y)} \max_{\xi \in \Xi(0)} \quad (\bar{H}^\top y - \pi^{\max} \circ x)^\top \xi,$$

where $\pi^{\max}_j$ *can be replaced by* $\max_{y \in Y} (\bar{H}^\top y)^\top e_j$ *as a result of Proposition 4 and the assumption that* $y^\top \bar{H} \geq 0$ *for* $y \in Y$.

We show next an example in which $z^{\bar{\Pi}}_{LR} = 0 < z^{\text{new}}_{LR} = z$. *Consider* $X = \left\{x \in \{0,1\}^q \;\middle|\; e^\top x = 1\right\}$, $Y = \left\{y \in \{0,1\}^n \;\middle|\; e^\top y = n - 1\right\}$, $\bar{H}_{1k} = M$ *for each* $k \in [q]$, *where* $M > 0$ *is large enough, and* $\bar{H}_{jk} = 1$ *for each* $j > 1$ *and* $k \in [q]$. *We first remark that the definitions of* Y *and* $\bar{H}$ *imply that an optimal solution to both relaxations satisfy* $y_1 = 0$ *and* $y_j = 1$ *for* $j > 1$. *Therefore,* $\eta_{1k} = 0$ *for each* $k \in [q]$ *and* $\eta_{jk} = \min(x_k, y_j) = x_k$ *for each* $j > 1$ *and* $k \in [q]$, *so the problem simplifies to*

$$\begin{aligned} z^{\text{new}}_{LR} = \min_{x \in rel(X)} \sum_{j \in [n]\backslash\{1\}, k \in [q]} (1 - x_k) &= \sum_{j \in [n]\backslash\{1\}, k \in [q]} 1 - \sum_{j \in [n]\backslash\{1\}} 1 \\ &= (n-1)(q-1), \end{aligned}$$

where the second equality holds since $e^\top x = 1$. *This value is also matched by the the integral optimal solution of the problem. On the other hand,* $\max_{y \in Y} (\bar{H}^\top y)^\top e_k \geq M$ *for each* k, *so Proposition 4 implies that* $\pi^{\max}_k = M$ *for each* k *and* $z^{\bar{\Pi}}_{LR} = 0$ *since* $(\bar{H}^\top y - \pi^{\max} \circ x)$ *can be rendered negative by setting* $x_k = \frac{1}{q}$ *for* $k \in [q]$ *for* M *sufficiently large.*

As illustrated by Example 1 the linear relaxation of (22)-(25) can be significantly stronger than that of the formulation proposed by [12].

4. Numerical experiments

In this section, we illustrate the numerical relevance of our complexity result presented in Theorem 1 on the robust shortest path problem. This problem was introduced by [12] in its generic form:

$$\min_{x \in X \subseteq \{0,1\}^{|A|}, y \in Y} \max_{\xi \in \Xi^{\text{SP}}(x)} c^\top x + (\bar{d} + \frac{1}{2}\xi \circ \bar{d})^\top y$$

where set X expresses the constraints imposed on variables x and set Y contains the flow constraints describing the shortest path problem, and $\Xi^{\text{SP}}(x)$ is given as:

$$\Xi^{\text{SP}}(x) = \left\{\xi \in \mathbb{R}^{|A|}_+ \;\middle|\; \sum_{(i,j) \in A} \xi_{ij} \leq \Gamma, \xi_{ij} \leq 1 - \gamma_{ij} x_{ij} \quad \forall (i,j) \in A\right\}.$$

In the following, we assume, without loss of generality, that $\bar{d} > 0$ since otherwise the corresponding components of ξ are always equal to 0 in the inner maximization problem.

We concentrate on the variant where $X = \{0,1\}^{|A|}$. Nohadani and Sharma [12] proposed three different formulations for this problem. Among those, $\bar{\Pi}$ and modified big-M formulations are the most numerically promising based on their results. We repeat these formulations here for completeness:

$$\min_{\substack{x\in X, y\in Y,\\ p,q,r\geq 0}} \quad c^\top x + \bar{d}^\top y + \Gamma p + \sum_{(i,j)\in A} (1-\gamma_{ij})q_{ij} + \gamma_{ij} r_{ij} \qquad (\bar{\Pi})$$

$$\begin{aligned} \text{s.t.} \quad & p + q_{ij} \geq \frac{\bar{d}_{ij} y_{ij}}{2} && \forall (i,j)\in A \\ & p + r_{ij} \geq \frac{\bar{d}_{ij}(y_{ij} - x_{ij})}{2} && \forall (i,j)\in A, \end{aligned}$$

and

$$\min_{\substack{x\in X, y\in Y,\\ p,q,r\geq 0}} \quad c^\top x + \bar{d}^\top y + \Gamma p + \sum_{(i,j)\in A} (1-\gamma_{ij})q_{ij} + r_{ij} \qquad \text{(mod. big-}M\text{)}$$

$$\begin{aligned} \text{s.t.} \quad & p + q_{ij} \geq \frac{\bar{d}_{ij} y_{ij}}{2} && \forall (i,j)\in A \\ & r_{ij} \geq \gamma_{ij} q_{ij} - \frac{\gamma_{ij}\bar{d}_{ij} x_{ij}}{2} && \forall (i,j)\in A. \end{aligned}$$

In the above formulations, big-M and $\bar{\pi}_{ij}$ are omitted as their values have already been set to $\gamma_{ij}\frac{\bar{d}_{ij}}{2}$ and 1, respectively. We also corrected small typos from Table 3 of [12]. For the same problem, the formulation we proposed in Section 3 is given as:

$$\min_{\substack{x\in X, y\in Y,\\ p,q,r,v\geq 0}} \quad c^\top x + \bar{d}^\top y + \Gamma p + \sum_{(i,j)\in A} (1-\gamma_{ij})q_{ij} + \gamma_{ij} r_{ij} \qquad (new)$$

$$\begin{aligned} \text{s.t.} \quad & p + q_{ij} \geq \frac{\bar{d}_{ij} y_{ij}}{2} && \forall (i,j)\in A \\ & p + r_{ij} \geq \frac{\bar{d}_{ij}(y_{ij} - v_{ij})}{2} && \forall (i,j)\in A \\ & v_{ij} \leq x_{ij} && \forall (i,j)\in A \\ & v_{ij} \leq y_{ij} && \forall (i,j)\in A, \end{aligned}$$

from which we have removed the redundant constraints $v_{ij} \geq x_{ij} + y_{ij} - 1$ for $(i,j)\in A$.

Remark 2. *Let $z^{\bar{\Pi}}$, z^M, and z^{new} denote the optimal values of the linear programming relaxations of* $(\bar{\Pi})$, (mod. big-M), *and* (*new*)*, respectively. It holds that $z^{\bar{\Pi}} = z^M = z^{\text{new}}$.*

Proof. First, let us consider the formulation (mod. big-M). Let $r'_{ij} = q_{ij} - \frac{\bar{d}_{ij}x_{ij}}{2}$ and note that $r_{ij} = \gamma_{ij} r'_{ij}$ at optimality. We may therefore replace r_{ij} by $\gamma_{ij} r'_{ij}$ in (mod. big-M) and impose the constraint $r'_{ij} \geq q_{ij} - \frac{\bar{d}_{ij}x_{ij}}{2}$ for $(i,j)\in A$. Further, renaming r'_{ij} as r_{ij}, the second set of constraints becomes:

$$r_{ij} \geq q_{ij} - \frac{\bar{d}_{ij} x_{ij}}{2} \qquad \forall (i,j)\in A,$$

and the objective coefficient of r_{ij} is now γ_{ij} so the objective functions of all formulations coincide. Next, because of the positive cost coefficients of q and r, we can substitute these variables in $(\bar{\Pi})$ by

$$q_{ij}^{\bar{\Pi}} = \left[\frac{\bar{d}_{ij} y_{ij}}{2} - p\right]^+ \text{ and } r_{ij}^{\bar{\Pi}} = \left[\frac{\bar{d}_{ij}(y_{ij} - x_{ij})}{2} - p\right]^+$$

and we obtain similarly for (mod. big-M) that

$$q_{ij}^{M} = \left[\frac{\bar{d}_{ij} y_{ij}}{2} - p\right]^+ \text{ and } r_{ij}^{M} = \left[\left[\frac{\bar{d}_{ij} y_{ij}}{2} - p\right]^+ - \frac{\bar{d}_{ij} x_{ij}}{2}\right]^+.$$

Now, $\frac{\bar{d}_{ij}x_{ij}}{2} \geq 0$ implies that

$$\left[\left[\frac{\bar{d}_{ij} y_{ij}}{2} - p\right]^+ - \frac{\bar{d}_{ij} x_{ij}}{2}\right]^+ = \left[\frac{\bar{d}_{ij} y_{ij}}{2} - p - \frac{\bar{d}_{ij} x_{ij}}{2}\right]^+,$$

so the set of optimal solutions coincide for the linear programming relaxations of both formulations.

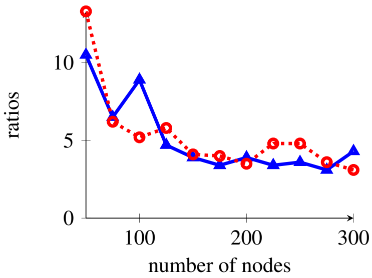
To prove the equivalence with the last formulation, we observe first that in any optimal solution to (*new*), we have $v_{ij} = \min(x_{ij}, y_{ij})$. Next we remark that if $\min(x_{ij}, y_{ij}) = y_{ij}$ the constraint $p + r_{ij} \geq \frac{\bar{d}_{ij}(y_{ij}-v_{ij})}{2}$ becomes redundant in (*new*), as does the constraint $p + r_{ij} \geq \frac{\bar{d}_{ij}(y_{ij}-x_{ij})}{2}$ in $(\bar{\Pi})$ since $y_{ij} - x_{ij} \leq 0$. Otherwise, if $\min(x_{ij}, y_{ij}) = x_{ij}$ the two constraints are equivalent. We therefore conclude that the set of optimal solutions for the linear relaxations of both formulations coincide, proving the result.

□

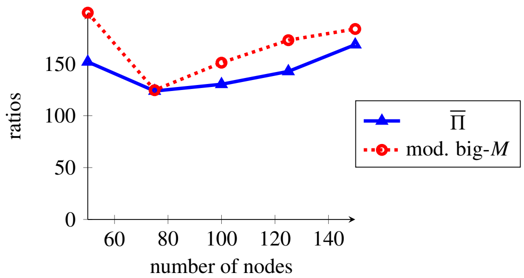
Following Remark 2, we disregard formulation (*new*) of our numerical experiments as it is significantly larger than the other two. We implemented the MILP formulations $(\bar{\Pi})$ and (mod. big-M) in JuMP [10], using the commercial solver CPLEX 20.1 as well as the open source solver HiGHs [8]. Our implementation of Theorem 1 benefited from Graphs.jl [5] as well as RobustShortestPath.jl [9]. The experiments are run using a single thread on a Intel Xeon E312xx (Sandy Bridge).

Instances were randomly generated following the procedure described in [12]. To do so, we created n points in the 100×100 square and connected them to create a complete graph. We then used euclidean distances for $\bar{d}$ and kept only the 40% shortest edges of the resulting complete graph in order to obtain the final graph. We set $\gamma_{ij} = 0.2$ and $c_{ij} = 1$ for each $(i,j)\in A$, and $\Gamma = 2$. For each $n \in \{25, 50, \ldots, 300\}$, we generated 10 instances. While CPLEX could sove all instances within a little less than 2 hours, that was not the case for HiGHS for which some instances with $n = 150$ required between 2 and 3 hours of solution time, so we limited its results to $n \in \{25, 50, \ldots, 150\}$.

We report in Figure 1 the geometric averages of the ratios between the solution times of Theorem 1 divided by those of formulations mod. big-M and $(\bar{\Pi})$. The results illustrate that the numerical efficiency of the two MILP formulations are comparable, both being between 3 and nearly 200 times slower than the polynomial algorithms obtained from Theorem 1, depending mostly on the performance of the MIP solver. The results thus offer an interesting practical takeaway on the benefit of using Theorem 1 depending on the available solver. On the one hand, if one is unable to use high performance commercial solvers such as CPLEX and Gurobi, then the polynomial approach from Theorem 1 is more than 2 orders of magnitude faster than the reformulation approaches, with a slightly increasing tendency as the number of nodes increases. On the other hand, the advantage of Theorem 1 is less important



(a) Using CPLEX for the MILPs.



(b) Using HiGHS for the MILPs.

Figure 1: Geometric averages of the solution times obtained using Theorem 1 divided by those of formulations mod. big-M and $(\overline{\Pi})$.

against a high performance solver such as CPLEX. Interestingly, the speed-up versus CPLEX was much more marked in [15], for which the MILP reformulations were orders of magnitude slower than the polynomial-time algorithms (see Remark 1).

5. Conclusions

In this paper, we consider robust optimization problems with uncertainty reduction where the upper bounds on the uncertain parameters are adjusted by the binary decision variables controlled by the decision-maker.

We first focus on the min-max version of this problem where the decisions are described by combinatorial sets. For these problems, we show that they can be solved as a series of deterministic problems whenever the uncertainty reduction decisions are not constrained. Further, we show that the min-max problem amounts to solving a series of linear programs even when the uncertainty reduction decision are constrained provided that a specific matrix is TU. We demonstrate the numerical interest of this result on the robust shortest path problem with uncertainty reduction which was first considered by [12]. Our results indicate that depending on the performance of the MIP solver the approach we propose can be significantly more efficient than reformulation approaches proposed by [12]. We further remark that the shortest path problem is well-suited for reformulation approaches given the rather small formulations available for the problem. Results could be different for problems less suited to MILP formulations, such as the minimum spanning tree for which formulations are typically much larger [11]. Our algorithmic approach could be expected to perform even more favorably in that context. We also propose an alternative MILP formulation for the problems under consideration in the case where $D \geq 0$. We show that these formulations provide a significantly stronger linear relaxation compared to the formulations proposed in the literature in certain cases although for the shortest path problem considered in our numerical study all formulations are shown to have the same relaxation value.

We leave for future work the extension of our reformulations as a series of linear programs for more general problems than the one considered in this paper. This involves the study of the integrality of specific polytopes.

Acknowledgements

This work was supported by Agence Nationale de la Recherche, the French national research agency, under the grant ANR-22-CE48-0018.

References

[1] Eduardo Álvarez-Miranda, Ivana Ljubic, and Paolo Toth. A note on the bertsimas & sim algorithm for robust combinatorial optimization problems. *4OR*, 11(4):349–360, 2013. doi: 10.1007/s10288-013-0231-6. URL `https://doi.org/10.1007/s10288-013-0231-6`.

[2] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999. doi: 10.1016/S0167-6377(99)00016-4.

[3] Dimitris Bertsimas and Melvyn Sim. The Price of Robustness. *Operations Research*, 52(1):35–53, 2004. doi: 10.1287/opre.1030.0065.

[4] Christoph Buchheim and Jannis Kurtz. Robust combinatorial optimization under convex and discrete cost uncertainty. *EURO J. Comput. Optim.*, 6(3):211–238, 2018. doi: 10.1007/S13675-018-0103-0. URL `https://doi.org/10.1007/s13675-018-0103-0`.

[5] James Fairbanks, Mathieu Besançon, Schölly Simon, Júlio Hoffiman, Nick Eubank, and Stefan Karpinski. Juliagraphs/graphs.jl: an optimized graphs package for the julia programming language, 2021. URL `https://github.com/JuliaGraphs/Graphs.jl/`.

[6] Marc Goerigk and Michael Hartisch. *An Introduction to Robust Combinatorial Optimization*, volume 361 of *International Series in Operations Research & Management Science*. Springer, 2024. Forthcoming. `https://link.springer.com/book/9783031612602`.

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

[8] Qi Huangfu and JA Julian Hall. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1):119–142, 2018.

[9] Taehan Lee and Changhyun Kwon. A short note on the robust combinatorial optimization problems with cardinality constrained uncertainty. *4OR*, 12:373–378, 2014.

[10] Miles Lubin, Oscar Dowson, Joaquim Dias Garcia, Joey Huchette, Benoît Legat, and Juan Pablo Vielma. Jump 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*, 2023. doi: 10.1007/s12532-023-00239-3.

[11] Thomas L Magnanti and Laurence A Wolsey. Optimal trees. *Handbooks in operations research and management science*, 7:503–615, 1995.

[12] Omid Nohadani and Kartikey Sharma. Optimization under decision-dependent uncertainty. *SIAM Journal on Optimization*, 28(2):1773–1795, 2018.

[13] Jérémy Omer and Michael Poss. Combinatorial robust optimization with decision-dependent information discovery and polyhedral uncertainty. *hal preprint hal-04097679*, 2023.

[14] Michael Poss. Robust combinatorial optimization with variable budgeted uncertainty. *4OR*, 11(1):75–92, 2013. doi: 10.1007/s10288-012-0217-9. URL `https://doi.org/10.1007/s10288-012-0217-9`.

[15] Michael Poss. Robust combinatorial optimization with variable cost uncertainty. *Eur. J. Oper. Res.*, 237(3):836–845, 2014. doi: 10.1016/j.ejor.2014.02.060. URL `https://doi.org/10.1016/j.ejor.2014.02.060`.

[16] Michael Poss. Robust combinatorial optimization with knapsack uncertainty. *Discrete Optimization*, 27:88–102, 2018.

[17] Simon A Spacey, Wolfram Wiesemann, Daniel Kuhn, and Wayne Luk. Robust software partitioning with multiple instantiation. *INFORMS Journal on Computing*, 24(3):500–515, 2012.

[18] Bo Zeng and Wei Wang. Two-stage robust optimization with decision dependent uncertainty. *arXiv preprint arXiv:2203.16484*, 2022.