OJMO

Open Journal of Mathematical Optimization

Jérémy Omer, Michael Poss & Maxime Rougier

Combinatorial Robust Optimization with Decision-Dependent Information Discovery and Polyhedral Uncertainty

Volume 5 (2024), article no. 5 (25 pages)

https://doi.org/10.5802/ojmo.33

Article submitted on December 23, 2023, accepted on July 19, 2024.

Open Journal of Mathematical Optimization is a member of the
Centre Mersenne for Open Scientific Publishing
http://www.centre-mersenne.org/
e-ISSN: 2777-5860

Combinatorial Robust Optimization with Decision-Dependent Information Discovery and Polyhedral Uncertainty

Jérémy Omer
Univ Rennes, INSA Rennes, CNRS, IRMAR - UMR 6625, 35000 Rennes, France
jeremy.omer@insa-rennes.fr

Michael Poss
LIRMM, University of Montpellier, CNRS, France
poss@lirmm.fr

Maxime Rougier
Univ Rennes, INSA Rennes, CNRS, IRMAR - UMR 6625, 35000 Rennes, France
maximerougier01@gmail.com

Abstract

Given a nominal combinatorial optimization problem, we consider a robust two-stages variant with polyhedral cost uncertainty, called Decision-Dependent Information Discovery (DDID). In the first stage, DDID selects a subset of uncertain cost coefficients to be observed, and in the second-stage, DDID selects a solution to the nominal problem, where the remaining cost coefficients are still uncertain. Given a compact linear programming formulation for the nominal problem, we provide a mixed-integer linear programming (MILP) formulation for DDID. The MILP is compact if the number of constraints describing the uncertainty polytope other than lower and upper bounds is constant. The proof of this result involves the generalization to any polyhedral uncertainty set of a classical result, showing that solving a robust combinatorial optimization problem with cost uncertainty amounts to solving several times the nominal counterpart. We extend this formulation to more general nominal problems through column generation and constraint generation algorithms. We illustrate our reformulations and algorithms numerically on the selection problem, the orienteering problem, and the spanning tree problem.

Digital Object Identifier 10.5802/ojmo.33

Keywords robust combinatorial optimization, compact formulations, column generation, cutting plane.

1 Introduction

Decision-dependent information discovery (DDID) tackles optimization problems under uncertainty where the decision maker has the possibility to investigate the value of some of the uncertain parameters, thereby reducing the total amount of uncertainty. The model has innumerous applications in urban planning, project management, resource allocations, scheduling, among many others. The first DDID models where motivated by applications in offshore oilfield exploitation [22] and production planning [23]. Subsequent examples have been considered in the literature and Vayanos et al. [45] detail applications in a R&D project portfolio optimization problem, where a company must choose how to prioritize the projects in its pipeline [13, 39]. Vayanos et al. [45] also describe a preference elicitation with real-valued recommendations where one can investigate how much users like any particular item. They further apply the latter model to improve the US kidney allocation system. Even more recently, Paradiso et al. [30] consider a routing problem, which they apply to collecting medicine crates at the Alrijne hospital.

We consider in this paper a model similar to that studied in [30] and address robust DDID where only the costs are uncertain. We further assume that the underlying nominal optimization problem is a combinatorial optimization problem, thus involving only 0/1 decision variables. Specifically, we define the following feasibility and uncertainty sets.

- $\mathcal{W} = \left\{ w \in \{0,1\}^n \;\middle|\; Gw \leq g \right\}$ is the set characterizing the possible information discovery;
- $\mathcal{Y} = \left\{ y \in \mathbb{Z}^n \;\middle|\; By \geq b, 0 \leq y \leq 1 \right\}$ is the feasible set of a given combinatorial optimization problem;
- $\mathcal{P} = \left\{ y \in \mathbb{R}^n \;\middle|\; By \geq b, 0 \leq y \leq 1 \right\}$ is the relaxed polytope of $\mathcal{Y}$;
- $\Xi = \left\{ \xi \in \mathbb{R}^n \;\middle|\; A\xi \leq r, 0 \leq \xi \leq d \right\}$ is an uncertainty polytope.

Observe that binary restrictions are written implicitly for $\mathcal{Y}$. This will make more natural the writing of the dualization over $\mathcal{P}$, the polyhedral relaxation of $\mathcal{Y}$. The DDID problem we consider is then defined by:

$$z^{\text{DDID}} = \min_{w\in\mathcal{W}} \max_{\bar{\xi}\in\Xi} \min_{y\in\mathcal{Y}} \max_{\xi\in\Xi(w,\bar{\xi})} \sum_{i\in[n]} (c_i + \xi_i) y_i, \tag{DDID}$$

where c is a given cost vector and $\Xi(w,\bar{\xi}) = \left\{\xi \in \Xi \mid w \circ \xi = w \circ \bar{\xi}\right\}$, where $v \circ w = (v_1 w_1, \ldots, v_n w_n)$ for any pair of vectors $v, w \in \mathbb{R}^n$. Observe that constraint $w \circ \xi = w \circ \bar{\xi}$ guarantees that the observed cost values are not modified after setting the solution y to the combinatorial optimization problem.

▶ Remark 1. The lower bounds equal to 0 in the definition of Ξ is without loss of generality. If the polytope were instead defined by $\Xi = \left\{\xi \in \mathbb{R}^n \mid A\xi \le r, \underline{d} \le \xi \le d\right\}$, we could change variables by setting $\xi' := \xi - \underline{d}$ and obtain a new polytope Ξ' that satisfies $\underline{d}' = 0$, together with a new cost vector c' and a new right-hand-side r'.

We define next the outermost objective function of (DDID), namely

$$\Phi(w) = \max_{\bar{\xi}\in\Xi} \min_{y\in\mathcal{Y}} \max_{\xi\in\Xi(w,\bar{\xi})} \sum_{i\in[n]} (c_i + \xi_i) y_i.$$

Folklore complexity results in min-max robust combinatorial optimization [25] imply that computing Φ for general polytopes is $\mathcal{NP}$-hard even when optimizing over $\mathcal{Y}$ is easy.

Observation 1. *Computing Φ is $\mathcal{NP}$-hard if the number of rows of A is part of the input, even if* $\min_{y\in\mathcal{Y}} c^T y$ *can be solved in polynomial time.*

Despite its many applications, solving DDID exactly even with the budget uncertainty polytope [9, 10] has so far remained a formidable challenge.

1.1 Literature Review

We contextualize next how (DDID) fits within the robust optimization landscape. Robust combinatorial optimization introduced in [25] originally considered min-max optimization problems of the form

$$z^{\text{MM}} = \min_{y\in\mathcal{Y}} \max_{\xi\in\Xi} \sum_{i\in[n]} (c_i + \xi_i) y_i \tag{Min-Max}$$

for discrete uncertainty sets Ξ. They proved in particular that (Min-Max) is $\mathcal{NP}$-hard even when Ξ consists of only two points and $\mathcal{Y}$ is the feasibility set of polynomially solvable optimization problems, such as the selection problem or the shortest path problem. In fact, their results apply also to polyhedrons so these problems remain hard even when Ξ is, for instance, the convex hull of two points.

Slightly later than [25], Ben-Tal and Nemirovski [7] considered robust optimization through a different perspective, focusing on convex uncertainty sets and uncertain constraints. They developed the first compact convex reformulations for these problems. While their focus was on convex optimization, applying their reformulations to robust combinatorial optimization problems with polyhedral uncertainty leads to compact mixed-integer linear programming formulations. These can be readily solved numerically using state-of-the-art solvers like CPLEX or Gurobi, despite the theoretical hardness of these problems. An important step forward arose with the introduction of the budget uncertainty set [9, 10] and its extension to more general knapsack constraints [33] (special case of Ξ where A is non-negative). Extending the seminal result of [9], Poss [33] showed that if the number of rows of A is constant, then the min-max robust counterparts of polynomial problems remains polynomial, contrasting with the difficulty proved by [25] for arbitrary sets. These rather theoretical results have been pursued for specific variants of the set [19, 47] and complemented by efficient algorithms that leverage the structure of the set, e.g. for vehicle routing [20, 31], scheduling [43], lot-sizing [1] and inventory routing [11, 37], only to name a few.

After the basic robust models were introduced by [25] and [7], many extensions have been considered in the literature. We briefly mention below two of these extensions that relate to (DDID). On the one hand, robust optimization with *decision-dependent* uncertainty sets allows for the uncertainty set Ξ to depend on the decision variables [3, 28, 32, 40]. On the other hand, *two-stage* robust optimization splits the decision variables into the here-and-now decisions, and the wait-and-see ones, which can be fixed after ξ is known. Numerous papers have been published on the topic (see the survey by [48]), providing exact [5, 49, 50] or approximate solutions [6] in the case of fractional recourse. The case of integer recourse has remained particularly difficult and, apart from the recent exact algorithms by [2, 24], research has mostly focused on approximate solutions based on partitioning the uncertain set into K subsets and devising constant second-stage policies for each element of the partition, often referred to as K-adaptability. While most of these approaches lead to decomposition algorithms [4, 41], Hanasusanto et al.[21] were able to provide a compact reformulation for K-adaptability when only cost is uncertain. Different authors [8, 34] have also proposed to partition Ξ dynamically and heuristically. While the above references aim at solving generic problems, Goerigk et al. [17] have focused on specific problems and proposed tailored algorithms and complexity results, see also [16].

Problem (DDID) borrows ideas from both of the above extensions. On the one hand, its decisions happen in multiple stages, since the observation w is to be decided before revealing anything from Ξ, while y is chosen after the observed coefficients $w \circ \bar{\xi}$ have been revealed. The difference with classical two-stage robust optimization lies in the remaining uncertain parameters, ξ, to be revealed only after y is decided. Furthermore, the second-stage uncertainty set $\Xi(w, \bar{\xi})$ is decision-dependent. We mention that there exist other robust optimization problems in the literature involving decision-dependent uncertainty sets and multiple stages, such as controllable uncertainty [26] or explorable uncertainty [15].

1.2 Contributions and structure of the paper

Our main result is a linear programming relaxation for $\Phi(w)$ that is exact whenever $\text{conv}(\mathcal{Y}) = \mathcal{P}$ and compact when the number of rows of A is constant. The linear program for $\Phi(w)$ is then dualized and linearized to provide a mixed-integer linear programming relaxation for (DDID) that is exact whenever $\text{conv}(\mathcal{Y}) = \mathcal{P}$ and compact when the number of rows of A is constant. An ad-hoc study is carried out to strengthen significantly the linearized MILP. Interestingly, our linear programming relaxation involves an extension of the result of [33] to arbitrary uncertainty polytopes. Our extension essentially states that if the number of rows of A is constant, then solving (Min-Max) can be done by optimizing a polynomial number of different linear functions over $\mathcal{Y}$.

Then, we discuss how these relaxations can be made exact for problems for which $\text{conv}(\mathcal{Y}) \subset \mathcal{P}$. First, we propose a convexification approach based on a Dantzig–Wolfe reformulation of $\text{conv}(\mathcal{Y})$, leading to column generation and branch-and-price algorithms. Second, we propose a cutting-plane algorithm starting from $\mathcal{P}$ and iteratively strengthening the outer approximation through strong valid inequalities. Both approaches can be turned into heuristic algorithms by stopping the variable or constraint generation at any time.

The resulting exact and heuristic algorithms are assessed numerically on different problems motivated by the literature, namely the selection, the orienteering problem, and the spanning tree problem. Results illustrate how these approaches are able to obtain exact solutions, often for the first time, on instances inspired by the scientific literature [30, 45]. They also illustrate the efficiency of our branch-and-price and cutting-plane algorithms.

We provide also more theoretical insights into the problem. First, we illustrate extreme cases in which (DDID) is equal to either its min-max or max-min counterpart. The former case arises when considering linear programs rather than discrete problems, while the latter arises when the dimension of Ξ is too small, such as the factor model used in the literature [45]. Second, we show that computing $\Phi(w)$ can alternatively be done by optimizing a polynomial number of linear functions over $\mathcal{Y}$. This leads to polynomial time algorithms for (DDID) whenever the nominal problems are polynomially solvable and $|\mathcal{W}|$ is polynomially bounded. While this result is mostly of theoretical interest, since it relies on the ellipsoid algorithm, the underlying cutting-plane algorithm can be used to compute $\Phi(w)$ whenever $\text{conv}(\mathcal{Y}) \subset \mathcal{P}$. We note that a similar result has been found independently in [26, Theorem 4.3] in the context of controllable uncertainty.

The rest of the paper is structured as follows. In the next section, we detail the relationship between (DDID) and its min-max and max-min counterparts, and provide the counterpart of the result of [33] for general polytopes (thus not assuming that $A \geq 0$). We provide in Section 3 the polynomial-time algorithm and linear programming reformulation for $\Phi(w)$. We dualize and linearize this formulation in Section 4, and discuss in Section 5 extensions to problems for which $\text{conv}(\mathcal{Y}) \subset \mathcal{P}$. Section 6 presents our numerical experiments. The appendix contains a comparison with the algorithm of [30], the proofs of the linearization of $\Phi(w)$ and the dominance relationships used in the column generation algorithms. It also details the reformulation proposed by [45] for K-adaptability.

1.3 Notations

We let s be the number of rows of matrix A. For any $w \in \{0,1\}^n$, we denote by $\mathcal{W}^1$ the set of indices over which w is equal to 1, and $\mathcal{W}^0$ its complementary. For any integer n, we denote the set $\{1, \ldots, n\}$ by $[n]$. For any real number x, $[x]^+$ denotes the positive part of x. For any set $\mathcal{S}$, we denote its convex hull by $\text{conv}(\mathcal{S})$ and the set of its extreme points by $\text{ext}(\mathcal{S})$. The identity matrix is denoted Id, and $\mathbf{1}$ and $\mathbf{0}$ represent vectors of all ones and zeros, respectively.

2 Preliminary results and trivial cases

2.1 Relationship with robust counterparts

If no cost coefficient can be observed (i.e., $\mathcal{W} = \{\mathbf{0}\}$), we see that (DDID) falls down to (Min-Max). Going one step further, we note that when every cost coefficient can be observed (i.e., $\mathbf{1} \in \mathcal{W}$), (DDID) becomes the (robust) wait-and-see problem, formally defined as

$$z^{\text{WS}} = \max_{\xi \in \Xi} \min_{y \in \mathcal{Y}} \sum_{i \in [n]} (c_i + \xi_i) y_i, \tag{Wait\&See}$$

where a worst-case cost vector can be inferred preliminary to the solution of the combinatorial problem. As a consequence the optimal value of (DDID) can be bounded as follows.

$$\begin{aligned} &\max_{\xi \in \Xi} \min_{y \in \mathcal{Y}} \sum_{i \in [n]} (c_i + \xi_i) y_i \leq \min_{w \in \mathcal{W}} \max_{\bar{\xi} \in \Xi} \min_{y \in \mathcal{Y}} \max_{\xi \in \Xi(w, \bar{\xi})} \sum_{i \in [n]} (c_i + \xi_i) y_i \leq \min_{y \in \mathcal{Y}} \max_{\xi \in \Xi} \sum_{i \in [n]} (c_i + \xi_i) y_i \\ \iff \quad & z^{\text{WS}} \leq z^{\text{DDID}} \leq z^{\text{MM}}. \end{aligned} \tag{1}$$

We note that if $\mathbf{1} \in \mathcal{W}$, it is trivially an optimal solution to (DDID). As considering this trivial solution raises a technical special case in our reformulations, we rather assume it does not belong to $\mathcal{W}$.

Assumption 1. *Set $\mathcal{W}$ does not contain the vector of all ones.*

We illustrate below the above inequalities on the selection problem with budget uncertainty.

Example 2. Consider an instance of (DDID) where $\mathcal{Y} = \left\{ y \in \{0,1\}^5 \;\middle|\; \sum_{i \in [5]} y_i = 1 \right\}$ is the selection feasibility set, $\mathcal{W} = \left\{ w \in \{0,1\}^5 \;\middle|\; \sum_{i \in [5]} w_i = 1 \right\}$ amounts to choosing one item among 5 and the uncertainty is the budget uncertainty set from [9] with nominal values $c = (1,2,3,4,5)$ and deviations $d = (5,4,3,2,1)$, that is, $\Xi = \left\{ \xi \in \mathbb{R}^5 \;\middle|\; \sum_{i \in [5]} \frac{\xi_i}{d_i} \leq 1,\; 0 \leq \xi_i \leq d_i,\; i \in [5] \right\}$.

Let us first look at the optimal solution to (Min-Max). Since a unique item j is selected in any feasible solution $y \in \mathcal{Y}$,

$$\max_{\xi \in \Xi} \sum_{i \in [n]} (c_i + \xi_i) y_i = \max_{\xi \in \Xi} c_j + \xi_j = c_j + d_j = 6.$$

Hence, $z^{\text{MM}} = 6$. In the case of (Wait&See), the adversary needs to increase the value of the cheapest item j, thus solving $\max_{\xi \in \Xi} \min_j (c_j + \xi_j)$. After some linear algebra, one obtains $z^{\text{WS}} = 162/47 \approx 3.4$.

Consider now (DDID), where we select one item after having observed one of the items cost. Assume that we observe item 1 so the uncertain cost ξ_1 of item 1 is revealed. If y selects item 1, the solution cost is $c_1 + \bar{\xi}_1$. If, we select instead item $j \neq 1$, the resulting solution cost is $c_j + d_j(1 - \frac{\bar{\xi}_1}{d_1})$. The previous value is minimized for $j = 2$, yielding $2 + 4(1 - \frac{\bar{\xi}_1}{5})$. The worst-case scenario for $\bar{\xi}$ thus maximizes $\min\{c_1 + \bar{\xi}_1, 2 + 4(1 - \frac{\bar{\xi}_1}{5})\}$, which is a concave piece-wise linear function with maximum value $34/9$ reached at $\bar{\xi}_1 = 25/9$. Therefore, observing item 1 yields an objective value of $z^{\text{DDID}} = 34/9 \approx 3.8$, one readily verifies by examination that this is the optimal solution to the problem.

We detail next two situations in which one of the bounds is actually equal to z^{DDID}. First, consider the linear programming counterpart of (DDID), in which the feasibility set of the optimization problem consists of a polytope, $\mathcal{Q}$. In this context, it is well-known that

$$\min_{y \in \mathcal{Q}} \max_{\xi \in \Xi} \sum_{i \in [n]} (c_i + \xi_i) y_i = \max_{\xi \in \Xi} \min_{y \in \mathcal{Q}} \sum_{i \in [n]} (c_i + \xi_i) y_i, \tag{2}$$

meaning that the robust optimization problem is equivalent to (Wait&See). Combining (2) with (1) immediately shows that (DDID) is equivalent to (Min-Max) in this context.

Observation 2. *If $\mathcal{Y}$ is a polytope instead of a set of integer points, then $z^{\text{WS}} = z^{\text{DDID}} = z^{\text{MM}}$.*

Observation 2 illustrates the necessity to consider discrete variables for (DDID) to provide an advantage over the min-max approach. In particular, equality (2) does not hold if one optimizes over a discrete set rather than a polytope since in the former case the domain of variables y is no longer convex.

When we are not in one of the two extreme cases where $\mathcal{W} = \{\mathbf{0}\}$ or $\mathbf{1} \in \mathcal{W}$, we may still develop some geometrical intuition on the role of information discovery. It may indeed be convenient to see the process of observation as a reduction of the dimension of the uncertainty polytope. To be more accurate, we partition the uncertainty constraints as $A^+\xi \le a^+, A^=\xi \le a^=$ such that $A^=\xi = a^=$ for all $\xi \in \Xi$ and there exists $\xi \in \Xi$ with $A^+\xi < a^+$. The dimension of polytope Ξ is then given by $\dim(\Xi) = n - \mathrm{rank}(A^=)$. We also define e_i the i^{th} vector of the canonical basis and for $I \subset [n]$, $E_I \in \mathbb{R}^{|I| \times n}$ the matrix whose rows are the $e_i^T, i \in I$.

Observation 3. *Let $w \in \mathcal{W}$, $\bar{\xi} \in \Xi$ and $\mathcal{W}^1 = \left\{i \in [n] \;\middle|\; w_i = 1\right\}$, then*

$$\dim\left(\Xi(w, \bar{\xi})\right) \le n - \mathrm{rank}\begin{pmatrix} A^= \\ E_{\mathcal{W}^1} \end{pmatrix}.$$

Proof. By definition, $\Xi(w, \bar{\xi}) = \left\{\xi \in \mathbb{R}^n \;\middle|\; A^+\xi \le a^+, A^=\xi \le a^=, E_{\mathcal{W}^1}\xi = E_{\mathcal{W}^1}\bar{\xi}\right\}$. We know that for all $\xi \in \Xi(w, \bar{\xi})$, $\binom{A^=}{E_{\mathcal{W}^1}}\xi = \binom{a^=}{E_{\mathcal{W}^1}\bar{\xi}}$, so $\dim\left(\Xi(w, \bar{\xi})\right) \le n - \mathrm{rank}\binom{A^=}{E_{\mathcal{W}^1}}$. ◀

In the literature, it is usual to consider the information discovery set

$$\mathcal{W}^{\mathrm{sel}} = \left\{ w \in \{0,1\}^n \;\middle|\; \sum_{i \in [n]} w_i \le q \right\},$$

where one can select up to q cost coefficients, $q \in \mathbb{Z}_+$. This discovery set allows to provide a more specific description of information discovery. Indeed, we see that matrix $\binom{A^=}{E_{\mathcal{W}^1}}$ corresponds to the completion of the rows of $A^=$ with row vectors of the canonical basis of $\mathbb{R}^n$. We thus know that we may choose $w \in \mathcal{W}^{\mathrm{sel}}$ such that $\mathrm{rank}\binom{A^=}{E_{\mathcal{W}^1}} = \mathrm{rank}(A^=) + q$ if $\mathrm{rank}(A^=) > n - q$, and $\mathrm{rank}\binom{A^=}{E_{\mathcal{W}^1}} = n$ if $\mathrm{rank}(A^=) \le n - q$. We deduce the following.

Corollary 3. *There is $w \in \mathcal{W}^{\mathrm{sel}}$ such that, for any $\bar{\xi} \in \Xi$, $\dim\left(\Xi(w, \bar{\xi})\right) \le \max\{0, \dim(\Xi) - q\}$.*

Interestingly, Corollary 3 implies that picking $w^* \in \mathcal{W}^{\mathrm{sel}}$ (through basic linear algebra) that most reduces the dimension of Ξ may substantially simplify (DDID) when the dimension of Ξ is not greater than q.

Observation 4. *If $\mathcal{W} = \mathcal{W}^{\mathrm{sel}}$ and $\dim(\Xi) \le q$, then $z^{\mathrm{DDID}} = z^{\mathrm{WS}}$.*

Proof. Corollary 3 implies that there exists $w^* \in \mathcal{W}^{\mathrm{sel}}$ such that $\Xi(w^*, \bar{\xi}) = \{\bar{\xi}\}, \forall\, \bar{\xi} \in \Xi$. Hence, $\Phi(w^*) = \max_{\bar{\xi} \in \Xi} \min_{y \in \mathcal{Y}} \sum_{i \in [n]} (c_i + \bar{\xi}_i) y_i = z^{\mathrm{WS}}$ ◀

2.2 Reformulating the robust counterpart

We generalize below a classical result from the literature [9, 33] that essentially shows that, when y is fixed, maximizing over Ξ in (Min-Max) amounts to take the minimum among $O(n^s)$ different affine functions of y. This result implies that solving (Min-Max) amounts to solve $O(n^s)$ nominal problems.

The result described in this section involves the following set. First, let $\mathcal{A}^{all} \subseteq \mathbb{R}^s$ contain the unique solution of each linearly independent subsystem of s equations of

$$\begin{pmatrix} A^T \\ A^T \\ \mathrm{Id} \end{pmatrix} \alpha = \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}. \tag{3}$$

We define $\mathcal{A} = \mathcal{A}^{all} \cap \mathbb{R}^s_+$, thus keeping only the non-negative vectors in $\mathcal{A}^{all}$.

Observation 5. *$|\mathcal{A}| \in O(n^s)$.*

Proof. Forming a linearly independent subsystem of s equations of (3) amounts to choose $0 \le k \le s$ rows of matrix A^T and $s - k$ rows of matrix Id. Then, for each row of matrix A^T that is chosen, we must further decide between the right-hand-side 0 and 1. We obtain $|\mathcal{A}| \le \sum_{k=0}^{s} \binom{n}{k} 2^k \le s \times n^s \times 2^s \in O(n^s)$. ◀

The next result is stated in the context of this paper, thus involving the decision-dependent uncertainty set $\Xi(w,\bar{\xi})$ and the set $\mathcal{W}^0$ of components that have not been observed. Nevertheless, one readily writes the counterpart of the result in the context of (Min-Max), by considering instead a polytope Ξ, together with $\mathcal{W}^0 = [n]$ and $\mathcal{W}^1 = \emptyset$.

Theorem 4. *Let $y \in \{0,1\}^n$. We have*

$$\max_{\xi \in \Xi(w,\bar{\xi})} \sum_{i \in \mathcal{W}^0} (c_i + \xi_i) y_i = \min_{\alpha \in \mathcal{A}} \left\{ \beta_{\alpha,0}(w,\bar{\xi}) + \sum_{i \in \mathcal{W}^0} (c_i + \beta_{\alpha,i}) y_i \right\},$$

where for each $\alpha \in \mathcal{A}$,

$$\beta_{\alpha,0}(w,\bar{\xi}) = \sum_{k \in [s]} \left(r_k - \sum_{i \in \mathcal{W}^1} a_{ki} \bar{\xi}_i \right) \alpha_k + \sum_{i \in \mathcal{W}^0} d_i \left[- \sum_{k \in [s]} a_{ki} \alpha_k \right]^+,$$

and for each $i \in [n], \alpha \in \mathcal{A}$,

$$\beta_{\alpha,i} = d_i \left(\left[1 - \sum_{k \in [s]} a_{ki} \alpha_k \right]^+ - \left[- \sum_{k \in [s]} a_{ki} \alpha_k \right]^+ \right).$$

Proof. Observe first that

$$\max_{\xi \in \Xi(w,\bar{\xi})} \sum_{i \in \mathcal{W}^0} (c_i + \xi_i) y_i = \max_{\xi' \in \Xi'} \sum_{i \in \mathcal{W}^0} (c_i + \xi'_i) y_i, \tag{4}$$

where

$$\Xi' = \left\{ \xi' \in \mathbb{R}^{|\mathcal{W}^0|} \;\middle|\; \sum_{i \in \mathcal{W}^0} a_{ki} \xi'_i \le r_k - \sum_{i \in \mathcal{W}^1} a_{ik} \bar{\xi}_i,\ k \in [s],\ 0 \le \xi'_i \le d_i,\ i \in \mathcal{W}^0 \right\}$$

is the projection of $\Xi(w,\bar{\xi})$ into the subset of coordinates indexed by $\mathcal{W}^0$. Let us denote $r_k - \sum_{i \in \mathcal{W}^1} a_{ki} \bar{\xi}_i$ by $\bar{r}_k$ for each $k \in [s]$. Notice that by Assumption 1, $\left|\mathcal{W}^1\right| < n$ so Ξ' is non-empty. Let us denote the dual variables of the constraints of Ξ' by α_k and π_i, respectively. Dualizing the maximization problem in the right-hand-side of (4) yields

$$\begin{aligned} \max_{\xi' \in \Xi'} \sum_{i \in \mathcal{W}^0} (c_i + \xi_i) y_i &= \sum_{i \in \mathcal{W}^0} c_i y_i + \min \left\{ \sum_{k \in [s]} \bar{r}_k \alpha_k + \sum_{i \in \mathcal{W}^0} d_i \pi_i \;\middle|\; \sum_{k \in [s]} a_{ki} \alpha_k + \pi_i \ge y_i,\ \forall\, i \in \mathcal{W}^0,\ \alpha, \pi \ge 0 \right\} \\ &= \sum_{i \in \mathcal{W}^0} c_i y_i + \min_{(\alpha,\pi) \in \mathcal{Q}(w,y)} \left\{ \sum_{k \in [s]} \bar{r}_k \alpha_k + \sum_{i \in \mathcal{W}^0} d_i \pi_i \right\}, \end{aligned} \tag{5}$$

where $\mathcal{Q}(w,y)$ is the polyhedron

$$\mathcal{Q}(w,y) = \left\{ (\alpha,\pi) \in \mathbb{R}^{s + |\mathcal{W}^0|} \;\middle|\; \sum_{k \in [s]} a_{ki} \alpha_k + \pi_i \ge y_i,\ \forall\, i \in \mathcal{W}^0,\ \alpha, \pi \ge 0 \right\}.$$

We show below that any extreme point (α^*, π^*) of $\mathcal{Q}(w,y)$ satisfies

$$\pi_i^* = \left[y_i - \sum_{k \in [s]} a_{ki} \alpha_k^* \right]^+ \tag{6}$$

for each $i \in \mathcal{W}^0$, and $\alpha^* \in \mathcal{A}$. Let $n^0 = |\mathcal{W}^0|$ denote the dimension of π. The former follows immediately from the fact that each π_i appears only in the constraints $\pi_i \ge 0$ and $\sum_{k \in [s]} a_{ki} \alpha_k + \pi_i \ge y_i$. Then, being an extreme point, (α^*, π^*) is necessarily the unique solution of a linearly independent subsystem of $n^0 + s$ equalities among the constraints of $\mathcal{Q}(w,y)$. Among these, n^0 are the equalities mentioned in (6). The remaining s equalities are taken among the non-negativity constraints $\left\{\alpha_k^* \geq 0 \;\middle|\; k \in [s]\right\}$, and the constraints of $\left\{\sum_{k\in[s]} a_{ki}\alpha_k^* + \pi_i^* \geq y_i \;\middle|\; i \in \mathcal{W}^0\right\}$ not used in (6), thus corresponding to $\pi_i^* = 0$. Therefore, $\alpha^* \in \mathcal{A}$ follows from $y \in \{0,1\}^n$.

Since the minimum of (5) is reached at least at one of the extreme point of $\mathcal{Q}(w,y)$, we have

$$\min_{(\alpha,\pi)\in\mathcal{Q}(w,y)} \left\{ \sum_{k\in[s]} \bar{r}_k\alpha_k + \sum_{i\in\mathcal{W}^0} d_i\pi_i \right\} \geq \min_{\alpha\in\mathcal{A}} \left\{ \sum_{k\in[s]} \bar{r}_k\alpha_k + \sum_{i\in\mathcal{W}^0} d_i \left[y_i - \sum_{k\in[s]} a_{ki}\alpha_k \right]^+ \right\}. \tag{7}$$

To prove the reverse inequality, we consider any $\alpha \in \mathcal{A} \subset \mathbb{R}^s_+$ and define $\pi_i = \left[y_i - \sum_{k\in[s]} a_{ki}\alpha_k\right]^+$ for $i \in [n]$, so $(\alpha,\pi) \in \mathcal{Q}(w,y)$.

Observing that $y_i \in \{0,1\}$ for each $i \in \mathcal{W}^0$, we can reformulate the objective of the right-hand-side of (7) to make it linear in y

$$\begin{aligned}
&\min_{\alpha\in\mathcal{A}} \left\{ \sum_{k\in[s]} \bar{r}_k\alpha_k + \sum_{i\in\mathcal{W}^0} d_i \left(y_i \left[1 - \sum_{k\in[s]} a_{ki}\alpha_k\right]^+ + (1-y_i)\left[-\sum_{k\in[s]} a_{ki}\alpha_k\right]^+ \right) \right\} \\
&= \min_{\alpha\in\mathcal{A}} \left\{ \sum_{k\in[s]} \bar{r}_k\alpha_k + \sum_{i\in\mathcal{W}^0} d_i \left[-\sum_{k\in[s]} a_{ki}\alpha_k\right]^+ + \sum_{i\in\mathcal{W}^0} d_i y_i \left(\left[1 - \sum_{k\in[s]} a_{ki}\alpha_k\right]^+ - \left[-\sum_{k\in[s]} a_{ki}\alpha_k\right]^+ \right) \right\}
\end{aligned}$$

◀

Applying Theorem 4 to (Min-Max) (hence $\mathcal{W}^0 = [n]$ and $\mathcal{W}^1 = \emptyset$) and switching the two minimizations immediately leads to the following.

Corollary 5. *Problem* (Min-Max) *amounts to solve $O(n^s)$ nominal problems $\min_{y\in\mathcal{Y}}(c+\beta_\alpha)^T y$.*

3 Computing Φ

3.1 Constraint generation

We present next an algorithm for computing Φ that relies on constraint generation. The first step of the approach described next applies an epigraphic reformulation to the outermost maximization problem

$$\Phi(w) = \max \quad \eta \tag{8a}$$

$$\text{s.t.} \quad \eta \leq \min_{y\in\mathcal{Y}} \max_{\xi\in\Xi(w,\bar{\xi})} \sum_{i\in[n]} (c_i+\xi_i)y_i \tag{8b}$$

$$\bar{\xi} \in \Xi. \tag{8c}$$

Then, we introduce dual variables α, π and γ so that linear programming duality yields

$$\max_{\xi\in\Xi(w,\bar{\xi})} \xi^T y = \min_{(\alpha,\pi,\gamma)\in\mathcal{D}(y)} r^T\alpha + d^T\pi + (w\circ\bar{\xi})^T\gamma = \min_{(\alpha,\pi,\gamma)\in\text{ext}(\mathcal{D}(y))} r^T\alpha + d^T\pi + (w\circ\bar{\xi})^T\gamma, \tag{9}$$

where $\mathcal{D}(y) = \left\{(\alpha,\pi,\gamma) \in \mathbb{R}^{2n+s} \;\middle|\; A^T\alpha + \text{Id}\,\pi + w\circ\gamma \geq y\right\}$ is the dual polytope. Plugging (9) into the right-hand side of (8b) leads to reformulating $\Phi(w)$ as a linear program with many constraints

$$\max \quad \eta \tag{10a}$$

$$\text{s.t.} \quad \eta \leq c^T y + r^T\alpha + d^T\pi + (w\circ\bar{\xi})^T\gamma, \quad \forall\, y\in\mathcal{Y}, (\alpha,\pi,\gamma)\in\text{ext}(\mathcal{D}(y)) \tag{10b}$$

$$\bar{\xi} \in \Xi. \tag{10c}$$

We now study the complexity of the separation problem associated with constraints (10b). Using (9) in the reverse direction, we see that a given candidate solution $(\eta^*, \bar{\xi}^*) \in \mathbb{R}\times\Xi$ is feasible for (10b) if and only if

$$\eta^* \leq \min_{y\in\mathcal{Y}} \max_{\xi\in\Xi(w,\bar{\xi}^*)} \sum_{i\in[n]} (c_i+\xi_i)y_i. \tag{11}$$

Thus, checking whether $(\eta^*, \bar{\xi}^*)$ is feasible amounts to solving a problem with same form as (Min-Max). Corollary 5 implies that the right-hand-side of (11) can be computed by solving $\min_{y\in\mathcal{Y}} \xi^T y$ for at most $O(n^s)$

vectors ξ. The overall approach leads to a cutting-plane algorithm for computing $\Phi(w)$, the separation problem of which is not harder than the nominal problem $\min_{y\in\mathcal{Y}} \xi^T y$. Then, using the equivalence between separation and optimization [38], we obtain a polynomial algorithm for computing $\Phi(w)$ whenever the nominal counterpart of the problem is polynomially solvable. We observe that the above algorithm is reminiscent of the one developed by [30] to compute $\Phi(w)$, the differences between the two are detailed in Appendix A.

Proposition 6. *If the nominal counterpart is polynomially solvable and s is constant, then $\Phi(w)$ can be computed in polynomial time.*

Assuming that the nominal counterpart is polynomially solvable, Observation 1 and Proposition 6 cover the two possible situations. If s is part of the input, computing $\Phi(w)$ is in general $\mathcal{NP}$-hard. Otherwise, it can be computed in polynomial time. Yet, the above algorithm involves the ellipsoid algorithm so it is hardly of any practical interest. For this reason, we provide in the next subsection an alternative approach to computing $\Phi(w)$, more amenable to numerical implementations. Before that, we further elaborate on the implications of Proposition 6.

Corollary 7. *If $|\mathcal{W}|$ is polynomially bounded, the nominal counterpart is polynomially solvable, and s is constant, then* (DDID) *can be solved in polynomial time.*

Proof. Enumerate all $w \in \mathcal{W}$ and compute $\Phi(w)$ for each of them, then return the minimum value. ◀

Corollary 7 implies, for instance, that if $\mathcal{P}$ is the matching polytope [35] (see also [38, Theorem 25.5]), the resulting (DDID) is easy. We note that the case of the matching polytope is particular in the sense that, although one can efficiently optimize over this polytope (e.g. [38, Section 25.5c]), its description requires exponentially many inequalities in general. Contrasting with the previous example, the polytopes of many polynomial combinatorial optimization problems can be described by polynomially many inequalities. This is the case for the shortest path problem, the minimum spanning tree problem (using the extended multi-commodity flow formulation [27]), or minimizing the weighted sum of completion times (e.g. [36, Section 4.1]), to name a few. For such problems, we provide below an alternative way to compute Φ that involves solving a compact linear program.

3.2 Linear programming formulation

We focus next on an optimization problem having a feasibility set described by a known polynomial number of linear inequalities, meaning that $\text{conv}(\mathcal{Y}) = \mathcal{P}$. We prove that under this additional assumption, $\Phi(w)$ amounts to solving a compact linear program.

Theorem 8. *Let $w \in \mathcal{W}$. If $\text{conv}(\mathcal{Y}) = \mathcal{P}$, then*

$$\Phi(w) = \begin{cases} \max & \eta \\ \text{s.t.} & \eta \le \sum_{k\in[s]} \left(r_k - \sum_{i\in\mathcal{W}^1} a_{ki}\bar{\xi}_i \right) \alpha_k + \sum_{i\in\mathcal{W}^0} d_i \left[-\sum_{k\in[s]} a_{ki}\alpha_k \right]^+ + b^T\lambda_\alpha - \sum_{i\in[n]} \pi_{\alpha,i}, \ \forall\, \alpha\in\mathcal{A} \\ & (B_{\cdot,i})^T\lambda_\alpha - \pi_{\alpha,i} \le c_i + \bar{\xi}_i, \qquad \forall\, \alpha\in\mathcal{A},\ \forall\, i\in\mathcal{W}^1 \\ & (B_{\cdot,i})^T\lambda_\alpha - \pi_{\alpha,i} \le c_i + \beta_{\alpha,i}, \quad \forall\, \alpha\in\mathcal{A},\ \forall\, i\in\mathcal{W}^0 \\ & \bar{\xi} \in \Xi \\ & \lambda, \pi \ge 0 \end{cases} \tag{12}$$

where for each $i \in [n], \alpha \in \mathcal{A}$, $\beta_{\alpha,i} = \left[1 - \sum_{k\in[s]} a_{ki}\alpha_k\right]^+ - \left[-\sum_{k\in[s]} a_{ki}\alpha_k\right]^+$.

Proof. Observe that

$$\max_{\xi\in\Xi(w,\bar{\xi})} \sum_{i\in[n]} \xi_i y_i = \sum_{i\in\mathcal{W}^1} \bar{\xi}_i y_i + \max_{\xi\in\Xi(w,\bar{\xi})} \sum_{i\in\mathcal{W}^0} \xi_i y_i. \tag{13}$$

Applying (13) to the epigraphic reformulation (8b) presented previously yields

$$\max \quad \eta \tag{14a}$$

$$\text{s.t.} \quad \eta \le \min_{y\in\mathcal{Y}} \left(\sum_{i\in\mathcal{W}^1} (c_i + \bar{\xi}_i) y_i + \max_{\xi\in\Xi(w,\bar{\xi})} \sum_{i\in\mathcal{W}^0} (c_i + \xi_i) y_i \right) \tag{14b}$$

$$\bar{\xi} \in \Xi. \tag{14c}$$

The main idea of the proof that follows reformulates (14b) through two ingredients: we reformulate the maximization over ξ using Theorem 4 (thus minimizing y over $\mathcal{Y}$ to use that y is binary), and dualize the minimization over y (thus using that $\text{conv}(\mathcal{Y}) = \mathcal{P}$ to minimize y over $\mathcal{P}$ instead of $\mathcal{Y}$).

Let us now work out the details of the above two ideas. Applying Theorem 4 to the last term of (14b) yields

$$\max_{\xi \in \Xi(w,\bar{\xi})} \sum_{i \in \mathcal{W}^0} (c_i + \xi_i) y_i = \min_{\alpha \in \mathcal{A}} \left\{ \beta_{\alpha,0}(w,\bar{\xi}) + \sum_{i \in \mathcal{W}^0} (c_i + \beta_{\alpha,i}) y_i \right\}.$$

Plugging the above expression into the right-hand-side of (14b) and swapping the minimizations, we obtain

$$\text{(14b)} \iff \eta \leq \min_{\alpha \in \mathcal{A}} \left\{ \beta_{\alpha,0}(w,\bar{\xi}) + \min_{y \in \mathcal{Y}} \sum_{i \in \mathcal{W}^1} (c_i + \bar{\xi}_i) y_i + \sum_{i \in \mathcal{W}^0} (c_i + \beta_{\alpha,i}) y_i \right\}.$$

The above may be written equivalently with $n+1$ independent constraints:

$$\text{(14b)} \iff \eta \leq \beta_{\alpha,0}(w,\bar{\xi}) + \min_{y \in \mathcal{Y}} \sum_{i \in \mathcal{W}^1} (c_i + \bar{\xi}_i) y_i + \sum_{i \in \mathcal{W}^0} (c_i + \beta_{\alpha,i}) y_i, \forall\, \alpha \in \mathcal{A}. \tag{15}$$

Thanks to the integrality of $\mathcal{P}$, we can relax the integrality restrictions in $\mathcal{Y}$ and replace the inner minimization over $\mathcal{Y}$ by the minimization over $\mathcal{P} = \left\{ y \in \mathbb{R}^n \;\middle|\; By \geq b, 0 \leq y \leq 1 \right\}$ in each constraint of (15). For each constraint $\alpha \in \mathcal{A}$, we then define the dual variables λ_α and π_α associated respectively with constraints $By \geq b$ and $y \leq 1$. We then dualize the minimization problem over $\mathcal{P}$ to get the following equivalent constraint:

$$\begin{aligned}
\eta \leq \beta_{\alpha,0}(w,\bar{\xi}) + \max_{\lambda_\alpha, \pi_\alpha} \quad & b^T \lambda_\alpha - \sum_{i \in [n]} \pi_{\alpha,i} \\
\text{s.t.} \quad & (B_{\cdot,i})^T \lambda_\alpha - \pi_{\alpha,i} \leq c_i + \bar{\xi}_i, && \forall\, i \in \mathcal{W}^1 \\
& (B_{\cdot,i})^T \lambda_\alpha - \pi_{\alpha,i} \leq c_i + \beta_{\alpha,i}, && \forall\, i \in \mathcal{W}^0 \\
& \lambda_\alpha, \pi_\alpha \geq 0
\end{aligned}$$

The maximization over λ_α and π_α is in the right-hand-side of a $\leq$ inequality, so the above is equivalent to the following set of constraints.

$$\begin{aligned}
& \eta \leq \beta_{\alpha,0}(w,\bar{\xi}) + b^T \lambda_\alpha - \sum_{i \in [n]} \pi_{\alpha,i} \\
& (B_{\cdot,i})^T \lambda_\alpha - \pi_{\alpha,i} \leq c_i + \bar{\xi}_i, \qquad \forall\, i \in \mathcal{W}^1 \\
& (B_{\cdot,i})^T \lambda_\alpha - \pi_{\alpha,i} \leq c_i + \beta_{\alpha,i}, \qquad \forall\, i \in \mathcal{W}^0 \\
& \lambda_\alpha, \pi_\alpha \geq 0
\end{aligned}$$

Replacing (14b) with the corresponding $|\mathcal{A}|$ sets of constraints provides the result. ◀

4 Solving the full problem

We describe below the compact reformulations obtained for (DDID) by dualizing and linearizing the formulation proposed in Theorem 8 using classical techniques.

Proposition 9. *If* $\text{conv}(\mathcal{Y}) = \mathcal{P}$, *then* (DDID) *is equivalent to* (DDID-WIP), *defined as*

$$\begin{aligned}
\min \quad & \sum_{\alpha \in \mathcal{A}} \left(r^T \alpha\, u_\alpha + \sum_{i \in [n]} d_i \left[- \sum_{k \in [s]} a_{ki} \alpha_k \right]^+ u^0_{\alpha,i} + \sum_{i \in [n]} c_i y_{\alpha,i} + \sum_{i \in [n]} \beta_{\alpha,i} y^0_{\alpha,i} \right) + d^T \sigma + r^T \mu \\
\text{s.t.} \quad & \sum_{\alpha \in \mathcal{A}} u_\alpha = 1 \\
& \sum_{k \in [s]} a_{ki} \mu_k + \sigma_i \geq - \sum_{k \in [s]} \sum_{\alpha \in \mathcal{A}} a_{ki} \alpha_k u_\alpha + \sum_{\alpha \in \mathcal{A}} y_{\alpha,i} - M_i (1 - w_i), \quad \forall\, i \in [n] \\
& B y_\alpha \geq u_\alpha b, \quad \forall\, \alpha \in \mathcal{A} \\
& y_{\alpha,i} \leq u_\alpha, \quad \forall\, \alpha \in \mathcal{A}, i \in [n]
\end{aligned}$$

$$
\begin{aligned}
& y^0_{\alpha,i} \geq y_{\alpha,i} - w_i, \quad \forall\, \alpha \in \mathcal{A}, i \in [n] \\
& u^0_{\alpha,i} \geq u_\alpha - w_i, \quad \forall\, \alpha \in \mathcal{A}, i \in [n] \\
& w \in \mathcal{W} \\
& u, u^0, y, y^0, \mu, \sigma \geq 0,
\end{aligned} \tag{DDID-WIP}
$$

where $M_i = 1 + \max_{\alpha \in \mathcal{A}} \big\{-\sum_{k\in[s]} a_{ki}\alpha_k\big\}$.

Proof. See Appendix B.1. ◀

Formulation (DDID-WIP) includes the decision on w with the most natural linearization of the products involving those variables. It is possible though to strengthen this formulation by projecting both y and u variables into the two sets $\mathcal{W}^1$ and $\mathcal{W}^0$.

Proposition 10. *If* $\text{conv}(\mathcal{Y}) = \mathcal{P}$*, then* (DDID) *is equivalent to* (DDID-SIP)*, defined as*

$$
\begin{aligned}
\min \quad & \sum_{\alpha\in\mathcal{A}} \left(r^T \alpha\, u_\alpha + \sum_{i\in[n]} d_i \left[-\sum_{k\in[s]} a_{ki}\alpha_k \right]^+ u^0_{\alpha,i} + \sum_{i\in[n]} c_i y_{\alpha,i} + \sum_{i\in[n]} \beta_{\alpha,i} y^0_{\alpha,i} \right) + d^T\sigma + r^T\mu \\
\text{s.t.} \quad & \sum_{\alpha\in\mathcal{A}} u_\alpha = 1 \\
& \sum_{k\in[s]} a_{ki}\mu_k + \sigma_i \geq -\sum_{k\in[s]}\sum_{\alpha\in\mathcal{A}} a_{ki}\alpha_k u^1_{\alpha,i} + \sum_{\alpha\in\mathcal{A}} y^1_{\alpha,i}, \quad \forall\, i \in [n], \\
& By_\alpha \geq u_\alpha b, \quad \forall\, \alpha \in \mathcal{A}, \\
& u_\alpha = u^0_{\alpha,i} + u^1_{\alpha,i}, \quad \forall\, \alpha \in \mathcal{A}, i \in [n], \\
& \sum_{\alpha\in\mathcal{A}} u^0_{\alpha,i} \leq 1 - w_i, \quad \forall\, i \in [n], \\
& \sum_{\alpha\in\mathcal{A}} u^1_{\alpha,i} \leq w_i, \quad \forall\, i \in [n], \\
& Gu^1_\alpha \leq u_\alpha g \quad \forall\, \alpha \in \mathcal{A}, \\
& y_{\alpha,i} = y^0_{\alpha,i} + y^1_{\alpha,i}, \quad \forall\, \alpha \in \mathcal{A}, i \in [n] \\
& y^0_{\alpha,i} \leq u^0_{\alpha,i}, \quad \forall\, \alpha \in \mathcal{A}, i \in [n] \\
& y^1_{\alpha,i} \leq u^1_{\alpha,i}, \quad \forall\, \alpha \in \mathcal{A}, i \in [n] \\
& w \in \mathcal{W} \\
& u, u^0, u^1, y, y^0, y^1, \sigma, \mu \geq 0.
\end{aligned} \tag{DDID-SIP}
$$

Proof. See Appendix B.2. ◀

We show below that the formulation of Proposition 10 is in general stronger than the one provided by Proposition 9. Numerical evidence shows that the inclusion may hold strictly.

Proposition 11. *Let* $\mathcal{S}^{\text{weak}}$ *and* $\mathcal{S}^{\text{strong}}$ *denote the projections on* w *of the formulations* (DDID-WIP) *and* (DDID-SIP)*, respectively. It holds that* $\mathcal{S}^{\text{strong}} \subseteq \mathcal{S}^{\text{weak}}$.

Proof. See Appendix B.3. ◀

The results from the previous section illustrate that whenever $\text{conv}(\mathcal{Y}) = \mathcal{P}$ and $|\mathcal{W}|$ is polynomially bounded, (DDID) is polynomially solvable. The reformulations obtained above show instead that when we assume only that $|\mathcal{W}|$ can be formulated as a mixed-integer set, s is constant, and $\text{conv}(\mathcal{Y}) = \mathcal{P}$, then (DDID) is in $\mathcal{NP}$.

5 Extensions to the case $\text{conv}(\mathcal{Y}) \subset \mathcal{P}$

The results presented so far rely on the fact that $\text{conv}(\mathcal{Y}) = \mathcal{P}$, meaning that we know a compact description for the convex hull of the set of all feasible solutions to the nominal optimization problem. In particular, the nominal problem, which optimizes a linear function over $\mathcal{Y}$, has so far been assumed to be polynomially solvable. We present next two possible extensions of our reformulations that can address DDID counterparts of problems for which such a compact description is not known.

5.1 Convexification

Our first approach to handle $\mathcal{NP}$-hard problems amounts to consider a Dantzig–Wolfe reformulation of set $\mathcal{Y}$. Let us enumerate this set as $\mathcal{Y} = \{\widetilde{y}_1, \ldots, \widetilde{y}_t\}$. Introducing the convex multipliers $\lambda_{\alpha,1}, \ldots, \lambda_{\alpha,t}$ for each $\alpha \in \mathcal{A}$, we can substitute y_α with $\sum_{s\in[t]} \lambda_{\alpha,s}\widetilde{y}_s$, and the constraints $By_\alpha \geq u_\alpha b$ with the convexification constraints, so (DDID-SIP) becomes

$$
\begin{aligned}
\min \quad & \sum_{\alpha\in\mathcal{A}} \left(r^T\alpha\, u_\alpha + \sum_{i\in[n]} d_i \left[-\sum_{k\in[s]} a_{ki}\alpha_k \right]^+ u^0_{\alpha,i} + \sum_{i\in[n]}\sum_{s\in[t]} c_i \lambda_{\alpha,s}\widetilde{y}_{s,i} + \sum_{i\in[n]} \beta_{\alpha,i} y^0_{\alpha,i} \right) + d^T\sigma + r^T\mu \\
\text{s.t.} \quad & \sum_{s\in[t]} \lambda_{\alpha,s} = u_\alpha, \quad \forall\, \alpha \in \mathcal{A}, && [\nu^\alpha] \\
& \sum_{s\in[t]} \lambda_{\alpha,s}\widetilde{y}_{s,i} = y^0_{\alpha,i} + y^1_{\alpha,i}, \quad \forall\, \alpha \in \mathcal{A}, i \in [n], && [\rho^\alpha_i] \\
& \sum_{\alpha\in\mathcal{A}} u_\alpha = 1 \\
& \sum_{k\in[s]} a_{ki}\mu_k + \sigma_i \geq -\sum_{k\in[s]}\sum_{\alpha\in\mathcal{A}} a_{ki}\alpha_k u^1_{\alpha,i} + \sum_{\alpha\in\mathcal{A}} y^1_{\alpha,i}, \quad \forall\, i \in [n], \\
& u_\alpha = u^0_{\alpha,i} + u^1_{\alpha,i}, \quad \forall\, \alpha \in \mathcal{A}, i \in [n], \\
& \sum_{\alpha\in\mathcal{A}} u^0_{\alpha,i} \leq 1 - w_i, \quad \forall\, i \in [n], \\
& \sum_{\alpha\in\mathcal{A}} u^1_{\alpha,i} \leq w_i, \quad \forall\, i \in [n], \\
& Gu^1_\alpha \leq u_\alpha g, \quad \forall\, \alpha \in \mathcal{A}, \\
& y^0_{\alpha,i} \leq u^0_{\alpha,i}, \quad \forall\, \alpha \in \mathcal{A}, i \in [n] \\
& y^1_{\alpha,i} \leq u^1_{\alpha,i}, \quad \forall\, \alpha \in \mathcal{A}, i \in [n] \\
& w \in \mathcal{W} \\
& u, u^0, u^1, y^0, y^1, \sigma, \lambda \geq 0, && \text{(DDID-CG)}
\end{aligned}
$$

where the right brackets denote dual variables. Observe that (DDID-CG) is a valid formulation for (DDID). Formulation (DDID-CG) can be used in two different ways. First, for some strongly constrained problems, it may happen that t is a moderately large integer so the formulation can be directly fed into a solver. In this case, if all costs are positive, one may further reduce the value of t by observing that only minimal $\widetilde{y}$, with respect to inclusion, need to be considered.

Observation 6. *Suppose $c + \xi \geq 0$ for each $\xi \in \Xi$ and consider $\widetilde{y}_{s_1}, \widetilde{y}_{s_2} \in \mathcal{Y}$ such that $\widetilde{y}_{s_1} \leq \widetilde{y}_{s_2}$ and let $\widetilde{\mathcal{Y}} = \mathcal{Y} \setminus \{\widetilde{y}_{s_2}\}$. Let $\widetilde{z}^{\text{DDID}}$ denote the optimal value of* (DDID-CG) *associated to $\widetilde{\mathcal{Y}}$. We have $z^{\text{DDID}} = \widetilde{z}^{\text{DDID}}$.*

Proof. See Appendix C.1. ◀

We can filter $\mathcal{Y}$ similarly by relying on bounds.

Observation 7. *Consider $\widetilde{y}_s \in \mathcal{Y}$ such that*

$$\min_{\xi\in\Xi} \sum_{i\in[n]} (c_i + \xi_i)\widetilde{y}_{s,i} > z^{\text{MM}}, \tag{16}$$

and let $\widetilde{\mathcal{Y}} = \mathcal{Y} \setminus \{\widetilde{y}_s\}$ and $\widetilde{z}^{\text{DDID}}$ denote the optimal value of (DDID) *associated to $\widetilde{\mathcal{Y}}$. We have $z^{\text{DDID}} = \widetilde{z}^{\text{DDID}}$.*

Proof. See Appendix C.2. ◀

Despite Observations 6 and 7, one cannot expect, in general, to be able to handle the entire problem at once. This leads to considering column-generation based algorithms which, essentially, generate appropriate subsets $T^\alpha \subseteq [t], \alpha \in \mathcal{A}$, on the fly by exploiting dual information (see for instance [46]). Let us describe this idea more precisely in what follows, denoting by DLR(T) the dual of the linear relaxation of (DDID-CG) associated to subsets $T^\alpha, \alpha \in \mathcal{A}$, while DLR denotes the dual of the full linear relaxation of (DDID-CG). Let υ denote the vector of all dual variables and consider an optimal dual solution υ^* of DLR(T). Notice that DLR has the same variables as DLR(T) but contains additional constraints. Hence, solution v^* is feasible for DLR as soon as it satisfies these additional constraints. In fact, the only constraints of DLR that are missing in DLR(T) are those associated with the primal variables $\lambda_{\alpha,s}$ for each $\alpha \in \mathcal{A}$ and $s \in [t] \setminus T^\alpha$, namely

$$\sum_{i\in[n]} c_i \widetilde{y}_{s,i} - \sum_{i\in[n]} \rho_i^{\alpha *} \widetilde{y}_{s,i} - \nu^{\alpha *} \geq 0. \tag{17}$$

Therefore, all constraints (17) are satisfied by v^* if and only if for each $\alpha \in \mathcal{A}$, the optimal solution of the following optimization problem is not smaller than $\nu^{\alpha *}$

$$\min_{y\in\mathcal{Y}} \left\{ \sum_{i\in[n]} c_i y_i - \sum_{i\in[n]} \rho_i^{\alpha *} y_i \right\}. \tag{18}$$

If, on the contrary, there exists $\alpha \in \mathcal{A}$ for which we are able to identify $\widetilde{y}_{s'} \in \mathcal{Y}$ such that the corresponding dual constraint (17) is violated by v^*, we add the corresponding index s' to T^α and solve the resulting linear program DLR(T) again.

The above procedure generates all required variables of (DDID-CG) at the root node of the branch-and-bound tree solving the problem. However, these do not cover all the variables that may be generated in the subsequent linear programs that result from adding the branching constraints. Repeating the above procedure at each node of the branch-and-bound tree leads to a branch-and-price algorithm. We underline that this is a rather easy branch-and-price algorithm in the sense the branching is not done on the dynamically generated variables. Another direct consequence of the above discussion is that the above column generation algorithm provides yet another way to compute $\Phi(w)$.

Observation 8. $\Phi(w)$ *can be computed by fixing* w *in* (DDID-CG) *and solving the resulting linear program with column generation.*

5.2 Cutting-plane algorithm

The second approach to handling $\text{conv}(\mathcal{Y}) \subset \mathcal{P}$ involves the iterative generation of $\text{conv}(\mathcal{Y})$ through valid inequalities, essentially cycling between the solution of a sequence of problems of type (DDID-SIP) and the separation of solutions y from $\text{conv}(\mathcal{Y})$. The first ingredient of this algorithm is thus a separation oracle for $\text{conv}(\mathcal{Y})$ as detailed next.

Assumption 2. *Given* $y \in \mathbb{R}^n$*, we have a separation oracle that returns either* true *if* $y \in \text{conv}(\mathcal{Y})$ *or a hyperplane separating* y *from* $\text{conv}(\mathcal{Y})$*.*

The second ingredient of the algorithm is the extension of (DDID-SIP) to any polytope

$$\mathcal{P}' = \{ y \in \mathbb{R}^n \mid B'y \geq b', 0 \leq y \leq 1 \}.$$

Specifically, we introduce the concatenated decision vector $\theta = (w, u, u^0, u^1, y, y^0, y^1, \sigma)$ and define $\Theta(\mathcal{P}')$ as the feasible set defined by all constraints of (DDID-SIP), using (B', b') instead of (B, b). Introducing further f for the objective function of (DDID-SIP), we can formulate the following MILP

$$z_{\mathcal{P}'}^{\text{DDID}} = \min_{\theta\in\Theta(\mathcal{P}')} f(\theta), \tag{19}$$

Observe that when $\mathcal{P}' = \text{conv}(\mathcal{Y})$, the condition of Proposition 10 is satisfied and (19) coincides with the exact reformulation (DDID-SIP), so $z_{\mathcal{P}'}^{\text{DDID}} = z^{\text{DDID}}$ in this case.

Observation 9. *If* $\text{conv}(\mathcal{Y}) \subset \mathcal{P}'$*, then* $\min_{\theta\in\Theta(\mathcal{P}')} f(\theta)$ *is a relaxation of the exact formulation* $\min_{\theta\in\Theta(\text{conv}(\mathcal{Y}))} f(\theta)$*.*

Proof. We see that $\text{conv}(\mathcal{Y}) \subseteq \mathcal{P}'$ implies $\Theta(\text{conv}(\mathcal{Y})) \subseteq \Theta(\mathcal{P}')$, proving the statement. ◀

The algorithm starts with $\mathcal{P}^0 = \mathcal{P}$ and solves (19), yielding the optimal solution θ^* and its cost $z_{\mathcal{P}^0}^{\text{DDID}}$. Then, observe that $\theta^* \in \Theta(\text{conv}(\mathcal{Y}))$ if and only if for each $\alpha \in \mathcal{A}$, either $u_\alpha = 0$ and $y_\alpha^* = 0$ or $y_\alpha^* \in \text{conv}(\mathcal{Y})$. Hence, we can use Assumption 2 to check whether $\theta^* \in \Theta(\text{conv}(\mathcal{Y}))$. If this is the case, Observation 9 implies that θ^* is optimal for the exact formulation $\min_{\theta \in \Theta(\text{conv}(\mathcal{Y}))} f(\theta)$ so $z^{\text{DDID}}_{\mathcal{P}^0} = z^{\text{DDID}}$. Otherwise, we rely on the oracle from Assumption 2 to obtain a separating hyperplane $h^T y \leq h^0$, define

$$\mathcal{P}^1 = \mathcal{P}^0 \cap \left\{ y \in \mathbb{R}^n \;\middle|\; h^T y \leq h^0 \right\},$$

and repeat the procedure.

We note that an alternative stopping criterion involves the computation of $\Phi(w^*)$ at each iteration, which can be computed by using one of the algorithms proposed in Section 3 or the column-generation algorithm described in Section 5.1.

Observation 10. *Let $\theta^* = (w^*, u^*, u^{0*}, u^{1*}, y^*, y^{0*}, y^{1*}, \sigma^*, \mu^*)$ be the solution returned at the i-th iteration of the algorithm. If $\Phi(w^*) \leq z^{\text{DDID}}_{\mathcal{P}^i}$, then $z^{\text{DDID}}_{\mathcal{P}^i} = z^{\text{DDID}}$.*

Proof. For any $w \in \mathcal{W}$, we have that $\Phi(w) \geq z^{\text{DDID}} \geq z^{\text{DDID}}_{\mathcal{P}^i}$, where the second inequality follows from Observation 9. Combining the above with $\Phi(w^*) \leq z^{\text{DDID}}_{\mathcal{P}^i}$ proves the result. ◀

The resulting algorithm is finitely convergent if the oracle returns facet-defining inequalities. In practice, one may interrupt the algorithm at any time and consider the solution w^* returned after a certain number of iterations.

6 Numerical experiments

We next describe the numerical assessment of the different formulations and algorithms presented thus far. All our experiments have been realized in Julia language [12], using JuMP [14] to interface the mixed integer linear programming (MILP) solver CPLEX 20.01. We ran our experiments on a processor Intel Xeon E312xx (Sandy Bridge) using 1 cpu at 2.3Ghz and reporting the total CPU times in seconds or centiseconds, depending on the problem. We set the same time limit to two hours in all our experiments. The source code of every algorithm is publicly available at `https://plmlab.math.cnrs.fr/mposs/ddid/`.

6.1 A further simplification when $s = 1$

All the numerical experiments reported in the following consider variants of the budget uncertainty set, so $s = 1$. It so happens that in this case, writing down the complementarity conditions between (12) and its dual, we are always able to construct an optimal solution where $\mu = 0$.

Proposition 12. *When $s = 1$, there is an optimal solution $(w^*, u^*, u^{0*}, u^{1*}, y^*, y^{0*}, y^{1*}, \mu^*, \sigma^*)$ to the compact formulation* (DDID-WIP) *such that $\mu^* = 0$.*

Proof. See Appendix D. ◀

6.2 Selection problem

We first experiment the reformulation from Proposition 9 with the selection problem, where the decision maker wishes to choose p out of n items, so $\mathcal{Y}^{\text{sel}} = \left\{ y \in \{0,1\}^n \;\middle|\; \sum_{i\in[n]} y_i = p \right\}$. The selection problem has been used in numerous papers addressing complex robust variants [17, 18, 19], including DDID itself under the name of two-stage robust best box selection [45], in which $p = 1$.

We use the budget uncertainty set of [9, 10], defined as $\Xi^\Gamma = \left\{ \xi \in \mathbb{R}^n \;\middle|\; \sum_{i\in[n]} \xi_i \leq \Gamma, 0 \leq \xi \leq 1 \right\}$, largely used in the scientific literature on robust combinatorial optimization. We further consider the selection set for information discovery, $\mathcal{W}^{\text{sel}} = \left\{ w \in \{0,1\}^n \;\middle|\; \sum_{i\in[n]} w_i \leq q \right\}$, where one can investigate up to q items. We consider $n \in \{10, 20, 30, 40, 50\}$, $p, q, \Gamma \in \{n/10, n/5\}$ and generate randomly 10 instances (meaning the generation of vectors c, d and f in $[0,1]^n$) for each quadruplet of parameters. For each instance, we further consider a variant where only $n/2$ parameters are uncertain, the other being fixed to their nominal values.

We first illustrate in Table 1 the distance between z^{DDID} and the bounds z^{WS} and z^{MM}. These results illustrate that for our instances, the average gaps are mostly below 10%. As expected, looking at column q we see how investigating more parameters moves z^{DDID} towards z^{WS}. We also see that larger values of Γ lead to smaller gaps, often significantly.

Table 2 reports the solution times in centiseconds and root gaps in % for the two formulations presented in Section 4. We see immediately the importance of strengthening the formulation as described in (DDID-SIP). This reduces the root gaps to close to 0% on average, thereby reducing the solving times by more than one order

Table 1 Average relative gaps in %. Left and right values are $100 \times \frac{z^{\text{DDID}} - z^{\text{WS}}}{z^{\text{DDID}}}$ (denoted WS) and $100 \times \frac{z^{\text{MM}} - z^{\text{DDID}}}{z^{\text{DDID}}}$ (denoted MM), respectively.

n	uncertainty				p				q				Γ			
	$n/2$		n		$n/10$		$n/5$		$n/10$		$n/5$		$n/10$		$n/5$	
	WS	MM	WS	MM	WS	MM	WS	MM	WS	MM	WS	MM	WS	MM	WS	MM
10	9	8	8	9	9	8	9	9	10	7	7	10	9	13	8	4
20	8	5	5	8	5	4	9	8	8	5	6	7	8	11	5	2
30	7	5	5	8	4	5	8	8	6	5	5	7	8	10	4	2
40	8	6	6	10	5	6	9	10	8	7	6	10	9	13	5	3
50	9	6	6	9	5	5	10	9	9	6	6	9	9	12	5	3

Table 2 Average solution times in centiseconds (T) and root gaps in % (gap).

(a) Weak formulation (DDID-WIP)

n	uncertainty				p				q				Γ			
	$n/2$		n		$n/10$		$n/5$		$n/10$		$n/5$		$n/10$		$n/5$	
	T	gap	T	gap	T	gap	T	gap	T	gap	T	gap	T	gap	T	gap
10	2	35	2	39	2	45	2	29	2	37	2	37	2	34	3	40
20	8	31	14	46	8	43	14	34	10	39	11	38	9	36	12	41
30	26	33	74	46	31	42	68	37	41	40	58	39	35	36	64	43
40	74	34	716	50	218	47	572	38	259	43	531	42	166	39	624	46
50	679	42	17883	55	5377	53	13184	43	2344	49	16217	48	1165	45	17396	51

(b) Strong formulation (DDID-SIP)

n	uncertainty				p				q				Γ			
	$n/2$		n		$n/10$		$n/5$		$n/10$		$n/5$		$n/10$		$n/5$	
	T	gap	T	gap	T	gap	T	gap	T	gap	T	gap	T	gap	T	gap
10	2	0.02	2	0.05	2	0.0	2	0.08	2	0.0	2	0.08	2	0.01	2	0.06
20	5	0.02	7	0.16	5	0.04	7	0.15	6	0.1	7	0.09	7	0.14	6	0.05
30	12	0.05	17	0.06	12	0.08	17	0.04	14	0.08	16	0.04	16	0.1	13	0.01
40	23	0.01	44	0.16	26	0.12	41	0.05	30	0.12	37	0.05	39	0.15	28	0.02
50	59	0.08	88	0.12	46	0.02	102	0.18	61	0.06	87	0.14	86	0.16	61	0.04

of magnitude. Looking more precisely at Table 2b, we see that only p and the proportion of uncertain items have a significant impact on the solution times. Unreported results show that (DDID-SIP) scales well for larger instances, solving problems with up to 200 items in a couple of minutes.

Table 3 Average solution times in centiseconds (T) and root gaps in % (gap) for the K-adaptability reformulation presented in Section E.

n	K	uncertainty				p				q				Γ			
		$\lceil n/2 \rceil$		n		$\lceil n/10 \rceil$		$n/5$		$\lceil n/10 \rceil$		$n/5$		$\lceil n/10 \rceil$		$n/5$	
		T	gap	T	gap	T	gap	T	gap	T	gap	T	gap	T	gap	T	gap
10	2	6	6	7	7	3	6	20	5	7	6	44	6	7	10	36	6
	3	65	7	74	7	8	7	550	6	76	6	3474	7	63	10	2122	7
15	2	79	4	82	6	20	5	132	7	44	6	80	6	36	6	97	3
	3	6147	5	5994	8	550	6	11172	9	3474	7	5257	8	2122	7	7974	4

We compared the above results with our own implementation of the K-adaptability reformulation proposed in [45], see Section E for details of the resulting formulation. The solution times are presented in Table 3 for $n \in \{10, 15\}$; larger values of n are not presented as many instances could not be solved be solved in one hour for $n = 20$. These results illustrate that the reformulations for K-adaptability are several orders of magnitude slower than the exact reformulations proposed in this paper. The results of Table 3 might seem contradictory with the

Table 4 Instances taken from [30], N being equal to n.

name	U	T
TS1N15	0.10	$\{5, 10, \dots, 70\}$
TS2N10	0.20	$\{15, 20, 23, 25, 27, 30, 32, 35, 38\}$
TS3N16	0.10	$\{5, 10, \dots, 80\}$
TS1N30	0.05	$\{5, 10, \dots, 70, 73, 80, 85\}$
TS2N19	0.15	$\{15, 20, 23, 25, 27, 30, 32, 35, 38, 40, 45\}$
TS3N31	0.05	$\{15, 20, \dots, 120\}$

results presented in [45, Table 1], which report instances of up to 50 items being solved in a few seconds for $K \in [10]$. However, notice that Vayanos et al. [45] model uncertainty by projecting a 4-dimensional box into $\mathbb{R}^n$, specifically,

$$\Xi^{\text{factor}} = \left\{\xi \in \mathbb{R}^n \;\middle|\; \exists\, \zeta \in [-1,1]^L : \xi_i = \psi_i(\zeta)\right\},$$

for given affine mappings ψ_i, and $L = 4$ risk factors. As stated in Observation 4, (DDID) is then equivalent to (WAIT&SEE) as soon as 4 items or more can be investigated, probably explaining the relative simplicity of the instances tested in [45]. In fact, a preliminary version of their work used instead $L \in \{20, 30\}$ factors, reporting solution times more aligned with those presented in Table 3, see [44, Figure 3]. Another source of simplification in [45] is that they consider $p = 1$, while we consider larger values of this parameter here, which appears to have a significant impact on the solution times.

6.3 Orienteering problem

Our second set of experiments focuses on a particular routing problem considered by [30]: the orienteering problem. That problem is most naturally stated as the maximization problem

$$\max_{w \in \mathcal{W}^{\text{OP}}} \min_{\bar{\xi} \in \Xi^{\text{OP}}} \max_{y \in \mathcal{Y}^{\text{OP}}} \min_{\xi \in \Xi^{\text{OP}}(w,\bar{\xi})} \sum_{i \in [n]} (c_i + \xi_i) y_i,$$

which we specify next by defining $\mathcal{Y}^{\text{OP}}, \Xi^{\text{OP}}$, and $\mathcal{W}^{\text{OP}}$. Consider a complete and undirected graph with $n+2$ nodes, numbered from 0 to $n+1$, where nodes 0 and $n+1$ denote the start and destination nodes, respectively, so $[n]$ indexes all nodes different from the depot. We denote by t_{ij} the travel time of edge $\{i, j\}$ and by T the maximum travel time. Any feasible element in $\mathcal{Y}^{\text{OP}}$ is an elementary path from 0 to $n+1$ having a total weight that does not exceed T. Introducing binary variable z_e to model the use of edge e in the path, and denoting the star of node i as $E(i) = \left\{e \in E \;\middle|\; i \in e\right\}$, we formulate $\mathcal{Y}^{\text{OP}}$ as

$$\mathcal{Y}^{\text{OP}} = \left\{ y \in \{0,1\}^n \;\middle|\; \begin{array}{l} \exists\, z \in \{0,1\}^{|E|} \text{ s.t. } t^T z \le T, \\ \displaystyle\sum_{e \in E(0)} z_e = \sum_{e \in E(n+1)} z_e = 1, \; \sum_{e \in E(i)} z_e = 2y_i, \;\; \forall\, i \in [n], \\ \text{subtour elimination constraints} \end{array} \right\},$$

where "subtour elimination constraints" denotes any set of constraints preventing cycles in y (e.g. [42]). Polytope $\mathcal{P}^{\text{OP}}$ is obtained from $\mathcal{Y}^{\text{OP}}$ by removing the integrality restrictions on y and z and projecting the resulting polytope on variables y. Furthermore, we follow [30], and define $c = 0$, $\mathcal{W}^{\text{OP}} = \left\{w \in \{0,1\}^n \;\middle|\; \sum_{i \in [n]} w_i \le q\right\}$, and

$$\Xi^{\text{OP}} = \left\{ \xi \in \mathbb{R}^n \;\middle|\; \sum_{i \in [n]} \xi_i \ge 1, \; 0 \le \xi_i \le U, \; \forall\, i \in [n] \right\},$$

for some given $U > 0$ and $q = \lceil \delta n \rceil$ for some given $\delta \in (0, 1)$. We observe that $\text{conv}(\mathcal{Y}^{\text{OP}}) = \mathcal{P}^{\text{OP}}$ does not hold in general, so we cannot apply Proposition 9 to this problem.

Therefore, we consider instead the convex hull formulation described in Section 5.1 and test the two exact approaches described in that section. First, we consider algorithm based on the full enumeration of the elements in $\mathcal{Y}^{\text{OP}}$. We consider the counterpart of Observation 6 for non-positive costs, thus enumerating only the maximal paths in $\mathcal{Y}^{\text{OP}}$. However, Observation 7 could not be leveraged. Indeed, the maximization counterpart of (16) becomes $\max_{\xi\in\Xi^{\text{OP}}}\sum_{i\in[n]}(c_i+\xi_i)\widetilde{y}_{s,i} < \max_{y\in\mathcal{Y}^{\text{OP}}}\min_{\xi\in\Xi^{\text{OP}}}\sum_{i\in[n]}(c_i+\xi_i)y_{s,i}$. With the above definitions of c and Ξ^{OP}, the condition becomes $\min(1,U|\widetilde{y}_s|) < \max\left(0, 1-U(n-\max_{y\in\mathcal{Y}^{\text{OP}}}|y|)\right)$, which is never satisfied for the values of n and U provided in Table 4, even when $|\widetilde{y}_s|=1$ and $\max_{y\in\mathcal{Y}^{\text{OP}}}|y|=1$. Second, we consider the branch-and-price algorithm described in the section. We test the two algorithms on a subset of the instances from [30], consisting of complete graphs with 10 to 31 nodes (excluding the depots), and further described in Table 4.

Table 5 Numerical results on the orienteering problem. Italicized results have been provided by [29] and have been run on a configuration different than ours (4.0GHz Intel i7-6000K processor, using CPLEX 12.10).

instance	δ	Opt (#)			time (s) (solved)		solved at root
		CB	conv	B&P	conv	B&P	B&P
TS2N10	0.25	*9/9*	9/9	9/9	0.13	0.71	7/9
	0.5	*9/9*	9/9	9/9	0.14	0.82	8/9
	0.75	*9/9*	9/9	9/9	0.13	0.44	9/9
TS1N15	0.25	*14/14*	14/14	14/14	82	3.6	12/14
	0.5	*14/14*	14/14	14/14	82	4.3	13/14
	0.75	*14/14*	14/14	14/14	81	4.6	13/14
TS3N16	0.25	*14/14*	13/14	14/14	528	19	12/14
	0.5	*14/14*	13/14	14/14	520	4.8	12/14
	0.75	*14/14*	13/14	14/14	518	20	11/14
TS2N19	0.25	*6/11*	7/11	11/11	891	147	7/11
	0.5	*8/11*	7/11	11/11	876	29	8/11
	0.75	*11/11*	7/11	11/11	877	291	9/11
TS1N30	0.25	*6/18*	6/18	17/18	1443	225	11/18
	0.5	*6/18*	6/18	17/18	1183	1008	11/18
	0.75	*10/18*	6/18	18/18	1110	193	11/18
TS3N31	0.25	*6/20*	3/20	18/20	99	568	16/20
	0.5	*6/20*	3/20	18/20	100	534	16/20
	0.75	*8/20*	3/20	18/20	98	728	16/20

The results are presented in Table 5. Columns `CB`, `conv` and `B&P` respectively denote the combinatorial Benders algorithm from [30], the exact convexification and the branch-and-price algorithm from Section 5.1. The columns "Time" report average solution times over the subset of instances solved by all methods. The reported times are rounded to the second apart from the smallest instances for which we keep one more digit. The column "solved at root" reports the numbers of instances that have been solved at the root node of `B&P`. Notice that the results reported for `CB` have been carried out using a different configuration (processor and version of CPLEX used), so the comparison between the respective columns should be made carefully.

Overall, our results indicate that `B&P` is the most efficient algorithm, solving nearly all instances to optimality within the time limit, many of them at the root node already. They also show that `conv`, despite its exhaustive enumeration, is somewhat competitive, altough it solves less instances to optimality thant the other algorithms. The reported times show that `B&P` is typically faster than `conv` on the instances that are solved by both algorithms, apart from the smallest instances, and the largest ones. In the latter case, either `conv` is very quick because the low value of T limits the number of elements of $\mathcal{Y}^{\text{OP}}$, or it takes a very long time. Solution times of `CB` are not included as they are run on a different configuration, making them hard to interpret.

6.4 Minimum spanning tree

Our last benchmark focuses on the DDID counterpart of the minimum spanning tree problem (MST), on which we illustrate and compare the three solution methods presented in Sections 4 and 5. As a first approach, we solve the compact MILP given by (DDID-SIP) for the directed multicommodity flow formulation of the MST, see [27], denoted `compact`. This formulation is compact and known to be exact, so Proposition 10 applies. The second approach, `CG`, relies on the column generation heuristic (not combined with the exact generation of

Table 6 Results for the MST presenting solution times, numbers of cuts and columns generated, root gaps and optimality gaps for `CG`.

name	nodes	edges	Γ	q	time (s)			cols	cuts	gap (%)	
					`compact`	`CG`	`CP`	`CG`	`CP`	root	`CG`
burma14.tsp	14	51	3	3	43	3.3	5.2	29	104	0.11	0.0
ulysses22.tsp	22	85	4	4	923	23	67	196	516	0.0	0.0
bays29.tsp	29	105	6	6	4850	29	316	42	530	0.15	0.24
swiss42.tsp	42	159	8	8	7110	132	100	276	0	0.2	0.21
eil51.tsp	51	186	10	10	93 080	1434	7920	769	1122	0.21	0.14

columns at each node of the branch-and-bound tree as for the orienteering). Each column added corresponds to an optimal tree returned by the Kruskal algorithm. The third approach, `CP`, is a cutting plane algorithm following the scheme described in Section 5.2. We consider the subtour formulation of the MST, see [27]. For a given solution θ^* of the current relaxation, we separate constraints of the subtour formulation by following the algorithm described by [27]. Given that the multi-commodity flow formulation is exact, for each $u^*_\alpha > 0$ the maximum flow from one arbitrary root to any other vertex must be equal to u^*_α if $y^*_l \in \text{conv}(\mathcal{Y})$. Otherwise, the minimum cut provides a subtour constraint to be added to the relaxed formulation. To speed-up the cutting plane generation, the initial relaxation of `CP` includes one set of aggregated multicommodity flow constraints (instead of one set of constraints for each $\alpha \in \mathcal{A}$ in `compact`).

The three methods are compared on a benchmark similar to that used by [15]. Each instance corresponds to an instance of the TSPLib, where each vertex has a given position and the nominal costs of the edges are given by the distances separating their two endpoints. The deviation are then set as 50% of the nominal values. We limit the density of the graphs by considering only the 6 closest neighbors of each vertex.

Table 6 presents solution times and statistics for the three algorithms. In the last two "gap" columns, "root" shows the relative difference between the optimal value found by `CP` or `compact` and that of the linear relaxation of `compact` and "CG" shows the relative difference between the best value obtained by `CG` and the optimal value found by `CP` or `compact`. The results indicate that `compact` is one to two orders of magnitude slower than `CP`. The good performance of `CP` is partly due to the strong initial relaxation since few subtour inequalities are generated, sometimes even 0. The results also indicate that `CG` returns near-optimal solutions for all instances, providing even exact solutions for the smallest two. Again, this good performance of `CG` is due to the excellent root gap, equal to 0 on one instance.

7 Conclusion

Decision-Dependent Information Discovery is a recent approach to situations where the decision maker can investigate some of the parameters before taking her actual decision. While the applications for the model are countless, the resulting optimization problems have remained very difficult to solve, even for the budget uncertainty polytope.

We have provided in this paper new efficient solution algorithms for the problem assuming that only the costs are uncertain, and that they belong to a polytope defined by a small number of constraints other than individual bounds. We have proposed a compact MILP formulation for the DDID counterpart of a nominal optimization problem that has a compact linear description. We have illustrated the reformulation on the selection problem, solving exactly instances with 50 items in one second on average, significantly improving over the literature. We have extended our reformulations to problems for which no compact linear formulation is available (such as $\mathcal{NP}$-hard problems) through column generation, branch-and-price, and row generation algorithms. Our experiments have again illustrated the interest of these algorithms. On the one hand, the branch-and-price algorithm applied to the orienteering instances considered by [30] has successfully solved nearly all instances to optimality. On the other hand, the cutting plane algorithm applied to the minimum-spanning tree problem has proved successful in solving exactly larger problem than possible with the compact reformulation alone.

In addition to these numerically-oriented results and formulations, we have also improved the theoretical understanding of DDID, showing that the problem is easy as soon as the nominal problem is polynomially solvable and the number of possible investigations is polynomially bounded. We have also clarified the link between DDID, the usual min-max counterpart, and the max-min wait-and-see counterpart, showing how DDID falls down to the latter when the number of components being investigated is not smaller than the dimension of the uncertainty set.

This work leads to several interesting open questions for future works. On the numerical side, the excellent results obtained by the exact branch-and-price algorithm call for generalizing the latter to other applications, hopefully leading to an efficient way to solve exactly DDID even when the nominal problem is $\mathcal{NP}$-hard. On the theoretical side, DDID inherits the $\mathcal{NP}$-hardness of the min-max problem for arbitrary uncertainty sets. However, its complexity is still unknown for a constant number of constraints, even in situations as simple as the selection problem with budget uncertainty.

Acknowledgment

The authors thank Rosario Paradiso for sharing his detailed computational results. They also thank Boris Detienne for useful suggestions regarding the efficient application of Observation 6 with BitArrrays.

A Comparison with the algorithm of [30]

First, the dualized formulation (10a)–(10c) requires only constraint generation, while the approach from [30] relies on the linearization

$$
\begin{aligned}
\max \quad & \eta && (20)\\
\text{s.t.} \quad & \eta \leq \sum_{i\in[n]} (c_i + \xi_i(y))\, y_i, \quad \forall\, y \in \mathcal{Y} && (21)\\
& \xi(y) \in \Xi(w,\bar{\xi}), \qquad \forall\, y \in \mathcal{Y} && (22)\\
& \bar{\xi} \in \Xi, && (23)
\end{aligned}
$$

where $\xi(y)$ plays the role of adjustable variables depending on y. Hence, Paradiso et al. [30] generate constraints (21) and (22) as well as variables $\xi(y)$ in the course of their algorithm. Second, we leverage Corollary 5 to reduce the separation to solving $O(n^s)$ nominal problems, while Paradiso et al. [30] address the problem though MILP formulations.

These two differences have a theoretical impact, since the running time of the algorithm from [30] cannot be polynomially bounded in general under the assumptions of Proposition 6. From the numerical viewpoint, the supremacy of one algorithm over the other will depend on the sets $\mathcal{Y}$ and Ξ.

B Proofs of Section 4

B.1 Proof of Proposition 9

Consider the linear program introduced in Theorem 8, and let us introduce dual variables u_α and $y_{\alpha,i}$ for the first three groups of constraints, together with μ and σ_i for the constraints defining Ξ. Dualizing the linear program yields (with the primal variables indicated into brackets)

$$
\begin{aligned}
\min \quad & \sum_{\alpha\in\mathcal{A}} \left(\left(r^T\alpha + \sum_{i\in\mathcal{W}^0} d_i \left[-\sum_{k\in[s]} a_{ki}\alpha_k \right]^+ \right) u_\alpha + \sum_{i\in\mathcal{W}^1} c_i y_{\alpha,i} + \sum_{i\in\mathcal{W}^0} (c_i + \beta_{\alpha,i}) y_{\alpha,i} \right) + d^T\sigma + r^T\mu \\
\text{s.t.} \quad & \sum_{\alpha\in\mathcal{A}} u_\alpha = 1 && [\eta]\\
& \sum_{k\in[s]} a_{ki}\mu_k + \sigma_i \geq -\sum_{k\in[s]}\sum_{\alpha\in\mathcal{A}} a_{ki}\alpha_k u_\alpha + \sum_{\alpha\in\mathcal{A}} y_{\alpha,i}, \quad \forall\, i \in \mathcal{W}^1 && [\bar{\xi}_i]\\
& \sum_{k\in[s]} a_{ki}\mu_k + \sigma_i \geq 0, \quad \forall\, i \in \mathcal{W}^0 && [\bar{\xi}_i]\\
& B y_\alpha \geq u_\alpha b, \quad \forall\, \alpha \in \mathcal{A} && [\lambda_\alpha]\\
& y_{\alpha,i} \leq u_\alpha, \quad \forall\, \alpha \in \mathcal{A}, i \in [n] && [\pi_\alpha]\\
& u, y, \mu, \sigma \geq 0.
\end{aligned}
$$

Notice that the constraints corresponding to $\bar{\xi}_i$ for $i \in \mathcal{W}^0$ are redundant and can be relaxed. Then, we introduce variables $w \in \mathcal{W}$ to represent $\mathcal{W}^1$ and $\mathcal{W}^0$, so the above problem is rewritten as

$$\begin{aligned}
\min \quad & \sum_{\alpha \in \mathcal{A}} \left(r^T \alpha\, u_\alpha + \sum_{i \in [n]} (1-w_i) d_i \left[-\sum_{k \in [s]} a_{ki}\alpha_k \right]^+ u_\alpha + \sum_{i \in [n]} c_i y_{\alpha,i} + \sum_{i \in [n]} \beta_{\alpha,i} y_{\alpha,i}(1-w_i) \right) + d^T\sigma + r^T\mu && (24)\\
\text{s.t.} \quad & \sum_{\alpha \in \mathcal{A}} u_\alpha = 1 && (25)\\
& \sum_{k \in [s]} a_{ki}\mu_k + \sigma_i \geq w_i \left(-\sum_{k \in [s]} \sum_{\alpha \in \mathcal{A}} a_{ki}\alpha_k u_\alpha + \sum_{\alpha \in \mathcal{A}} y_{\alpha,i} \right), \quad \forall\, i \in [n] && (26)\\
& By_\alpha \geq u_\alpha b, \quad \forall\, \alpha \in \mathcal{A} && (27)\\
& w \in W && (28)\\
& u, y, \mu, \sigma \geq 0. && (29)
\end{aligned}$$

Next, we linearize the product by w_i in (26) with a big-M term which yields:

$$\sum_{k \in [s]} a_{ki}\mu_k + \sigma_i \geq -\sum_{k \in [s]} \sum_{\alpha \in \mathcal{A}} a_{ki}\alpha_k u_\alpha + \sum_{\alpha \in \mathcal{A}} y_{\alpha,i} - M(1-w_i), \quad \forall\, i \in [n].$$

Given that $y_{\alpha,i} \leq u_\alpha, \forall\, i \in [n]$, we have $\sum_{\alpha \in \mathcal{A}} y_{\alpha,i} \leq \sum_{\alpha \in \mathcal{A}} u_\alpha = 1, \forall\, i \in [n]$. As a consequence, M can be set to $1 + \max_{\alpha \in \mathcal{A}} \{-\sum_{k \in [s]} a_{ki}\alpha_k\}$. We conclude by introducing variables $y^0_{\alpha,i}$ and $u^0_{\alpha,i}$ to represent the products $y_{\alpha,i}(1-w_i)$ and $u_\alpha(1-w_i)$, respectively, and adding the linearization constraints $y^0_{\alpha,i} \geq y_{\alpha,i} - w_i$ and $u^0_{\alpha,i} \geq u_\alpha - w_i$.

B.2 Proof of Proposition 10

We show that the model is a valid linearization of the intermediary model (24)–(29) of the proof of Proposition 9, having removed variable μ in accordance with Proposition 12. For this, we introduce $u^1_{i,\alpha} := w_i u_\alpha$, $u^0_{i,\alpha} := (1-w_i)u_\alpha$, $y^1_{i,\alpha} := w_i y_{\alpha,i}$, $y^0_{i,\alpha} := (1-w_i)y_{\alpha,i}$. Variables u^1 and y^1 stand for the decisions whose cost coefficients have been investigated whereas u^0 and y^0 stand for the others. The definitions of u^1 and u^0 may then be enforced in the model by adding the constraints $u_\alpha = u^0_{\alpha,i} + u^1_{\alpha,i}$, $u^0_{\alpha,i} \leq 1 - w_i$ and $u^1_{\alpha,i} \leq w_i$, for all $\alpha \in \mathcal{A}$ and $i \in [n]$. Similar constraints could be added to linearize y^0 and y^1, but we instead leverage constraints $y_{\alpha,i} \leq u_\alpha, \alpha \in \mathcal{A}, i \in [n]$, to add the tighter constraints $y_{\alpha,i} = y^0_{\alpha,i} + y^1_{\alpha,i}$, $y^0_{\alpha,i} \leq u^0_{\alpha,i}$ and $y^1_{\alpha,i} \leq u^1_{\alpha,i}$ for all $\alpha \in \mathcal{A}$ and $i \in [n]$. The objective function (24) and constraints (26) are then naturally linearized using the definitions of y^0, u^1 and y^1. Finally, constraints $Gu^1_\alpha \leq u_\alpha g$ are not necessary, but they are valid inequalities obtained by multiplying $Gw \leq g$ by u_α for each $\alpha \in \mathcal{A}$.

B.3 Proof of Proposition 11

We consider a feasible solution to the linear relaxation of (DDID-SIP) given by vectors $\bar{w}, \bar{u}, \bar{u}^0, \bar{u}^1, \bar{y}, \bar{y}^0, \bar{y}^1, \bar{\sigma}, \bar{\mu}$ and we consider its projection on the variables of (DDID-WIP), $\bar{w}, \bar{u}, \bar{u}^0, \bar{y}, \bar{y}^0, \bar{\sigma}, \bar{\mu}$. The satisfaction of most constraints is immediate, but some verifications need to be carried out for constraints $y^0_{\alpha,i} \geq y_{\alpha,i} - w_i$, $u^0_{\alpha,i} \geq u_\alpha - w_i$, and $\sum_{k \in [s]} a_{ki}\mu_k + \sigma_i \geq -\sum_{k \in [s]} \sum_{\alpha \in \mathcal{A}} a_{ki}\alpha_k u_\alpha + \sum_{\alpha \in \mathcal{A}} y_{\alpha,i} - M_i(1-w_i)$. For the first, we use that $\bar{y} = \bar{y}^0 + \bar{y}^1$ and $\bar{y}^1 \leq \bar{u}^1$ to show that $\bar{y}^0_{\alpha,i} \geq \bar{y}_{\alpha,i} - \bar{u}^1_{\alpha,i} \geq \bar{y}_{\alpha,i} - \bar{w}_i$ for all $\alpha \in \mathcal{A}, i \in [n]$. We show similarly for the second that $\bar{u}^0_{\alpha,i} \geq \bar{u}_{\alpha,i} - \bar{u}^1_{\alpha,i} \geq \bar{u}_{\alpha,i} - \bar{w}_i$ for all $\alpha \in \mathcal{A}, i \in [n]$.

To show that the last constraints are satisfied, we infer the following sequence of inequalities from the linear constraints of (DDID-SIP).

$$
\begin{aligned}
&-\sum_{k\in[s]}\sum_{\alpha\in\mathcal{A}} a_{ki}\alpha_k \bar{u}_\alpha + \sum_{\alpha\in\mathcal{A}} \bar{y}_{\alpha,i} - M_i(1-\bar{w}_i)\\
&= -\sum_{k\in[s]}\sum_{\alpha\in\mathcal{A}} a_{ki}\alpha_k \bar{u}^1_{\alpha,i} + \sum_{\alpha\in\mathcal{A}} \bar{y}^1_{\alpha,i} - \sum_{k\in[s]}\sum_{\alpha\in\mathcal{A}} a_{ki}\alpha_k \bar{u}^0_{\alpha,i} + \sum_{\alpha\in\mathcal{A}} \bar{y}^0_{\alpha,i} - \left(1+\max_{\alpha\in\mathcal{A}}\left\{-\sum_{k\in[s]} a_{ki}\alpha_k\right\}\right)(1-\bar{w}_i)\\
&\le -\sum_{k\in[s]}\sum_{\alpha\in\mathcal{A}} a_{ki}\alpha_k \bar{u}^1_{\alpha,i} + \sum_{\alpha\in\mathcal{A}} \bar{y}^1_{\alpha,i} - \sum_{k\in[s]}\sum_{\alpha\in\mathcal{A}} a_{ki}\alpha_k \bar{u}^0_{\alpha,i} + \sum_{\alpha\in\mathcal{A}} \bar{u}^0_{\alpha,i} - \left(1+\max_{\alpha\in\mathcal{A}}\left\{-\sum_{k\in[s]} a_{ki}\alpha_k\right\}\right)\sum_{\alpha\in\mathcal{A}} \bar{u}^0_{\alpha,i}\\
&\le -\sum_{k\in[s]}\sum_{\alpha\in\mathcal{A}} a_{ki}\alpha_k \bar{u}^1_{\alpha,i} + \sum_{\alpha\in\mathcal{A}} \bar{y}^1_{\alpha,i}\\
&\le \sum_{k\in[s]} a_{ki}\mu_k + \sigma_i.
\end{aligned}
$$

C Proofs of Section 5.1

C.1 Proof of Observation 6

To prove inequality $z^{\mathrm{DDID}} \ge \tilde{z}^{\mathrm{DDID}}$, let $(u^*, u^{0*}, u^{1*}, y^{0*}, y^{1*}, \sigma^*, \mu^*, \lambda^*)$ be an optimal solution to (DDID-CG) associated to $\mathcal{Y}$. We construct a solution $(u^*, u^{0*}, u^{1*}, y^{0*}, y^{1\prime}, \sigma^*, \mu^*, \lambda')$ to the formulation associated to $\widetilde{\mathcal{Y}}$ by setting $\lambda'_s = \lambda^*_s$ for $s \in [t] \setminus \{s_1, s_2\}$, $\lambda'_{s_1} = \lambda^*_{s_1} + \lambda^*_{s_2}$ (notice variable λ'_{s_2} does not exist in the new model) and $y^{1\prime}_{\alpha,i} = \sum_{s\in[t]} \lambda'_{\alpha,s}\widetilde{y}_{s,i} - y^{0*}_{\alpha,i}$, $\alpha \in \mathcal{A}, i \in [n]$. Observe that $\widetilde{y}_{s_1} \le \widetilde{y}_{s_2}$ implies that $\sum_{s\in[t]} \lambda'_{\alpha,s}\widetilde{y}_{s,i} \le \sum_{s\in[t]} \lambda'_{\alpha,s}\widetilde{y}_{s,i}$ and $y^{1\prime}_{\alpha,i} \le y^{1*}_{\alpha,i}$ for all $\alpha \in \mathcal{A}, i \in [n]$. A consequence, one readily verifies that $(u^*, u^{0*}, u^{1*}, y^{0*}, y^{1\prime}, \sigma^*, \mu^*, \lambda')$ is feasible and its cost is not larger than that of $(u^*, u^{0*}, u^{1*}, y^{0*}, y^{1*}, \sigma^*, \mu^*, \lambda^*)$. The reverse inequality is even more direct, plugging the solution obtained for $\widetilde{\mathcal{Y}}$ into the formulation associated to $\mathcal{Y}$.

C.2 Proof of Observation 7

Let $(w^*, \bar{\xi}^*, y^*, \xi^*)$ be an optimal solution to (DDID). If $y^* \neq \widetilde{y}_s$, the result is immediate. Otherwise, we detail next the resulting contradiction. Notice first that (16) implies $\max_{\xi\in\Xi} \sum_{i\in[n]}(c_i+\xi_i)\widetilde{y}_{s,i} > z^{\mathrm{MM}}$, and therefore $\min_{y\in\tilde{\mathcal{Y}}} \max_{\xi\in\Xi} \sum_{i\in[n]}(c_i+\xi_i)y_i = \min_{y\in\mathcal{Y}} \max_{\xi\in\Xi} \sum_{i\in[n]}(c_i+\xi_i)y_i$. On the one hand, $\widetilde{\mathcal{Y}} \subset \mathcal{Y}$ implies $z^{\mathrm{DDID}} \le \tilde{z}^{\mathrm{DDID}}$. On the other hand, we have that

$$
\begin{aligned}
z^{\mathrm{DDID}} = \Phi(w^*) &= \sum_{i\in[n]} (c_i + \xi^*_i)\widetilde{y}_{s,i}\\
&\ge \min_{\xi\in\Xi} \sum_{i\in[n]} (c_i + \xi_i)\widetilde{y}_{s,i}\\
&> \min_{y\in\mathcal{Y}} \max_{\xi\in\Xi} \sum_{i\in[n]} (c_i + \xi_i) y_i\\
&= \min_{y\in\tilde{\mathcal{Y}}} \max_{\xi\in\Xi} \sum_{i\in[n]} (c_i + \xi_i) y_i\\
&= \min_{y\in\tilde{\mathcal{Y}}} \max_{\bar{\xi}\in\Xi} \max_{\xi\in\Xi(w^*,\bar{\xi})} \sum_{i\in[n]} (c_i + \xi_i) y_i\\
&\ge \max_{\bar{\xi}\in\Xi} \min_{y\in\tilde{\mathcal{Y}}} \max_{\xi\in\Xi(w^*,\bar{\xi})} \sum_{i\in[n]} (c_i + \xi_i) y_i\\
&\ge \min_{w\in\mathcal{W}} \max_{\bar{\xi}\in\Xi} \min_{y\in\tilde{\mathcal{Y}}} \max_{\xi\in\Xi(w,\bar{\xi})} \sum_{i\in[n]} (c_i + \xi_i) y_i = \tilde{z}^{\mathrm{DDID}},
\end{aligned}
$$

where the third inequality arises from (16).

D Proof of Proposition 12

We consider in this proof the case $s = 1$. In that case, we see that we can assume $a_i > 0$ for each $i \in [n]$ otherwise $\xi_i = d_i$ in any optimal solution to the adversarial problem. We obtain the uncertainty polytope

$$\{\xi \in \mathbb{R}^n \mid a^T\xi \leq r, 0 \leq \xi \leq d\},$$

where r is now a scalar. In this case, we see that any $\mathcal{A} = 0 \cup \{1/a_i \mid i \in [n]\}$, so we use throughout the proof the notations $\alpha_\ell = 1/a_\ell$ for each $\ell \in [n]$, and for each $i, \ell \in [n]$, $\beta_{\ell,i} = d_i\left[1 - a_i/a_\ell\right]^+$, while $\alpha_0 = 0$ and $\beta_{i,0} = d_i, \forall\, i \in [n]$. We further note $[n] \cup \{0\}$ as $[n]_0$.

Referring to the counterparts of Theorem 8 and to the proof of Proposition 9 to the above setting, we will consider the pair of primal-dual adversary formulations given by

$$\mathbf{P}(w): \begin{cases} \max & \eta \\ \text{s.t.} & \eta \leq \left(r - \sum_{i\in\mathcal{W}^1} a_i\bar{\xi}_i\right)\alpha_\ell + b^T\lambda_\ell - \sum_{i\in[n]}\pi_{\ell,i}, \quad \forall\, \ell \in [n]_0 \quad [u_\ell] \\ & (B_{\cdot,i})^T\lambda_\ell - \pi_{\ell,i} \leq c_i + \bar{\xi}_i, \quad \forall\, \ell \in [n]_0,\ \forall\, i \in \mathcal{W}^1 \quad [y_{\ell,i}] \\ & (B_{\cdot,i})^T\lambda_\ell - \pi_{\ell,i} \leq c_i + \beta_{\ell,i}, \quad \forall\, \ell \in [n]_0,\ \forall\, i \in \mathcal{W}^0 \quad [y_{\ell,i}] \\ & a^T\bar{\xi} \leq r \quad [\mu] \\ & \bar{\xi} \leq d \quad [\sigma] \\ & \bar{\xi} \geq 0, \lambda_\ell, \pi_\ell \geq 0, \quad \forall\, \ell \in [n]_0, \end{cases}$$

$$\mathbf{D}(w): \begin{cases} \min & \sum_{\ell\in[n]_0}\left(r\alpha_\ell u_\ell + \sum_{i\in[n]} c_i y_{\ell,i} + \sum_{i\in\mathcal{W}^0}\beta_{\ell,i}y_{\ell,i}\right) + \sum_{i\in[n]} d_i\sigma_i + r\mu \\ \text{s.t.} & \sum_{\ell\in[n]_0} u_\ell = 1 \quad [\eta] \\ & a_i\mu + \sigma_i \geq -a_i\sum_{\ell\in[n]_0}\alpha_\ell u_\ell + \sum_{\ell\in[n]_0} y_{\ell,i}, \quad \forall\, i \in \mathcal{W}^1 \quad [\bar{\xi}_i] \\ & By_\ell \geq u_\ell b, \quad \forall\, \ell \in [n]_0 \quad [\lambda_\ell] \\ & y_{\ell,i} \leq u_\ell, \quad \forall\, \ell \in [n]_0, i \in [n] \quad [\pi_\ell] \\ & u, y, \mu, \sigma \geq 0, \end{cases}$$

and let $(\bar{\xi}^*, \lambda^*, \pi^*)$ and $(u^*, y^*, \mu^*, \sigma^*)$ be a pair of optimal solutions to $\mathbf{P}(w)$ and $\mathbf{D}(w)$.

Assume now that $\mu^* > 0$. We show next that it means that for each $\ell' \in [n]$ and $i' \in \mathcal{W}^0$, $\beta_{\ell',i'}y^*_{\ell',i'} = 0$. Let $\ell' \in [n]$ such that $u^*_{\ell'} > 0$ and $i' \in \mathcal{W}^0$ such that $y^*_{\ell',i'} > 0$. By complementarity, we have $(B_{\cdot,i'})^T\lambda^*_{\ell'} - \pi^*_{\ell',i'} = c'_i + \beta_{\ell',i'}$. If

$$\beta_{\ell',i'} > 0, \tag{30}$$

we build a new solution of $\mathbf{P}(w^*)$, $(\eta^*, \bar{\xi}^*, \lambda', \pi')$, by slightly modifying λ^* and π^*. We set $\lambda'_{\ell'} := \lambda^*_{i'}$ and $\pi'_{\ell'} := \pi^*_{i'}$ while keeping the other components of π^* and λ^* unchanged. Observe that $\mu^* > 0$ implies par complementarity that $\sum_{i\in\mathcal{W}^1} a_i\bar{\xi}^*_i = r$, so the constraints dual to u_ℓ can be simplified to

$$\eta^* \leq b^T\lambda^*_\ell - \sum_{i\in[n]}\pi^*_{\ell,i}, \quad \forall\, \ell \in [n]_0. \tag{31}$$

Observing further that (30) is equivalent to $a_{\ell'} > a_{i'}$ and using (31), one can verify that $(\eta^*, \bar{\xi}^*, \lambda', \pi')$ is feasible for $\mathbf{P}(w^*)$ and $(B_{\cdot,i})^T\lambda'_{\ell'} - \pi'_{\ell',i'} < c_{i'} + \beta_{\ell',i'}$. However, by complementarity, we also get $(B_{\cdot,i'})^T\lambda'_{\ell'} - \pi'_{\ell',i'} = c_{i'} + \beta_{\ell',i'}$, a contradiction. Therefore, $\beta_{\ell',i'} = 0$ (and thus $\left[1 - \frac{a_{i'}}{a_{\ell'}}\right]^+ = 0$) for all $i' \in \mathcal{W}^0$ and $\ell' \in [n]_0$ such that $y^*_{\ell',i'} > 0$. Transposing the above reasoning to $\ell' = 0$, we have $\beta_{0,i'} > 0$ so that $y^*_{0,i'} = 0$ for each $i' \in \mathcal{W}^0$.

Using the above, we get that the objective value of $\mathbf{D}(w^*)$ is given by

$$r\left(\mu^* + \sum_{\ell\in[n]}\alpha_\ell u^*_\ell\right) + \sum_{i\in[n]}\left(d_i\sigma^*_i + c_i\sum_{\ell\in[n]} y^*_{\ell,i}\right). \tag{32}$$

The rest of the proof constructs an algorithm that iteratively modifies the dual solution without changing the variables σ^* and the values of the sums $\mu^* + \sum_{\ell\in[n]} \alpha_\ell u_\ell^*$ and $\sum_{\ell\in[n]} y_{\ell,i}^*$ Thanks to (32), the algorithm does therefore not modify the cost of the solution.

The dual constraint of $\bar{\xi}_i, i \in \mathcal{W}^1$ may be rewritten as:

$$\mu^* \geq \frac{1}{a_i}\left(-\sigma_i^* + \sum_{\ell\in[n]} y_{\ell,i}^*\right) - \sum_{\ell\in[n]} \alpha_\ell u_\ell^*,$$

which must be active for at least one element of $\mathcal{W}^1$, which we denote j. Recalling $1/a_j = \alpha_j$, this means in particular that

$$\mu^* \leq \alpha_j \sum_{\ell\in[n]} y_{\ell,j}^* - \sum_{\ell\in[n]} \alpha_\ell u_\ell^* \leq \sum_{\ell\in[n]} (\alpha_j - \alpha_\ell)\, u_\ell^*.$$

As a consequence, there is $\ell \in [n]_0$ such that $u_\ell^* > 0$ and $\alpha_\ell < \alpha_j$. Using the above, we build another optimal solution of $\mathbf{D}(w^*)$, (y', u', μ', σ^*), where $\mu' = 0$, by iteratively decreasing the values of nonzero variables u_ℓ, and increasing the value of u_j while keeping constant the value of $\mu^* + \sum_{\ell\in[n]} \alpha_\ell u_\ell^*$. The iterative construction is formalized in Algorithm 1. At each step, one index ℓ such that $u_\ell^* > 0$ and $\alpha_\ell < \alpha_j$ is considered. The first computed value, δ, is the largest decrease of u'_k such that $u'_k \geq 0$ and $\mu' \geq 0$ at the end of the algorithm. The update of y'_k and y'_j then guarantee that $y'_\ell \leq u'_\ell, \forall\, \ell \in [n]$, $By'_\ell \geq u'_\ell b, \forall\, \ell \in [n]$, and $\sum_{\ell\in[n]} y'_\ell = \sum_{\ell\in[n]} y_\ell^*$. Finally, the update of μ' is such that $\mu' + \sum_{\ell\in[n]} \alpha_\ell u'_\ell = \mu^* + \sum_{\ell\in[n]} \alpha_\ell u_\ell^*$. It is then straightforward to verify that each step keeps (y', u', μ', σ^*) feasible and leaves its objective value unchanged. Moreover, we can verify that either $\mu' = 0$ at the end of an iteration or $\delta = u_k^*$. At the end of the algorithm, we thus have

$$\begin{aligned}\mu' &\leq \mu^* - \sum_{k\in K} (\alpha_j - \alpha_k)\, u_k^* \\ &\leq \sum_{\ell\in[n]} (\alpha_j - \alpha_\ell)\, u_\ell^* - \sum_{\ell\in[n]: a_\ell > a_j} (\alpha_j - \alpha_\ell)\, u_\ell^* \leq 0.\end{aligned}$$

Algorithm 1: Construction of an optimal solution where $\mu' = 0$

initialization: $y' := y^*, u' := u^*, \mu' := \mu^*, K = \left\{k \in [n] \;\middle|\; u_k^* > 0, \alpha_k < \alpha_j\right\}$.
for $k \in K$ **do**
  $\delta := \min\left\{u_k^*, \frac{\mu'}{\alpha_j - \alpha_k}\right\}$
  $u'_k \leftarrow u'_k - \delta$
  $u'_j \leftarrow u'_j + \delta$
  $y'_k \leftarrow y'_k - \frac{\delta}{u_k^*} y_k^*$
  $y'_j \leftarrow y'_j + \frac{\delta}{u_k^*} y_k^*$
  $\mu' \leftarrow \mu' - \delta\,(\alpha_j - \alpha_k)$
  if $\mu' = 0$ **then**
    break
return (y', u', μ')

E Compact reformulation for selection problem with K-adaptability

The K-adaptability approximation amounts to pre-select K recourse policies and choose the best of them upon realization of the uncertain parameters. Applied to (DDID), one obtains

$$z^{K\text{adapt}} = \min_{\substack{w\in\mathcal{W}\\ y^k\in\mathcal{Y}, k\in[K]}} \max_{\bar{\xi}\in\Xi} \min_{k\in[K]} \max_{\xi\in\Xi(w,\bar{\xi})} \sum_{i\in[n]} (c_i + \xi_i) y_i. \qquad (K\text{-ADAPT})$$

Consider the sets $\mathcal{Y}^{\text{sel}}$, $\mathcal{W}^{\text{sel}}$ and the budget uncertainty polytope

$$\Xi = \left\{\xi \in \mathbb{R}^n \;\middle|\; \sum_{i\in[n]} \frac{\xi_i - c_i}{d_i} \leq \Gamma, \xi \leq c + d, -\xi \leq -c\right\}.$$

Applying [45, Corollary 1] to (K-ADAPT) together with the symmetry breaking constraints detailed in Section EC.3.1. of [45], leads to the formulation presented below. For readability, we subdivide the dual variables β into β^Γ, β^{ub} (for the upper bounds on ξ) and β^{lb} (for the lower bounds on ξ) and similarly for β^k. We model the constraints $\overline{\gamma}_i^k = w_i\gamma_i^k$ with indicator constraints to avoid the burden of computing tight big M. Furthermore, we define $\Gamma' = \Gamma + \sum_{i\in[n]} c_i/d_i$.

$$
\begin{aligned}
\min\quad & \Gamma'\left(\beta^\Gamma + \sum_{k\in[K]}\beta^{k,\Gamma}\right) + \sum_{i\in[n]}(c_i+d_i)\left(\beta_i^{ub} + \sum_{k\in[K]}\beta_i^{k,ub}\right) - \sum_{i\in[n]} c_i\left(\beta_i^{lb} + \sum_{k\in[K]}\beta_i^{k,lb}\right) \\
\text{s.t.}\quad & \sum_{i\in[n]} w_i = q \\
& \sum_{i\in[n]} y_i^k = p, && \forall\, k\in[K] \\
& \sum_{k\in[K]} \alpha_k = 1 \\
& \frac{\beta^{k,\Gamma}}{d_i} + \beta_i^{k,ub} - \beta_i^{k,lb} + \overline{\gamma}_i^k = \overline{y}_i^k, && \forall\, i\in[n], k\in[K] \\
& \frac{\beta^{\Gamma}}{d_i} + \beta_i^{ub} - \beta_i^{lb} = \sum_{k\in[K]} \overline{\gamma}_i^k, && \forall\, i\in[n] \\
& w_i = 1 \implies \overline{\gamma}_i^k = \gamma_i^k, && \forall\, i\in[n], k\in[K] \\
& w_i = 0 \implies \overline{\gamma}_i^k = 0, && \forall\, i\in[n], k\in[K] \\
& \overline{y}_i^k \le y_i^k, \quad \overline{y}_i^k \le \alpha_k, \quad \overline{y}_i^k \ge \alpha_k - 1 + y_i^k && \forall\, i\in[n], k\in[K] \\
& z_i^{k,k+1} \le y_i^k + y_i^{k+1}, \quad z_i^{k,k+1} \le 2 - y_i^k - y_i^{k+1} && \forall\, i\in[n], k\in[K-1] \\
& z_i^{k,k+1} \ge y_i^k - y_i^{k+1}, \quad z_i^{k,k+1} \ge y_i^{k+1} - y_i^k && \forall\, i\in[n], k\in[K-1] \\
& y_i^k \ge y_i^{k+1} - \sum_{j=1}^{i-1} z_j^{k,k+1} && \forall\, i\in[n], k\in[K-1] \\
& \beta\ge 0, \beta^k \ge 0, \alpha\ge 0,\ y, z \text{ binary}
\end{aligned}
$$

References

1 Agostinho Agra, Marcio Costa Santos, Dritan Nace, and Michael Poss. A Dynamic Programming Approach for a Class of Robust Optimization Problems. *SIAM J. Optim.*, 26(3):1799–1823, 2016.

2 Ayşe N. Arslan and Boris Detienne. Decomposition-Based Approaches for a Class of Two-Stage Robust Binary Optimization Problems. *INFORMS J. Comput.*, 34(2):857–871, 2022.

3 Ayşe N. Arslan and Michael Poss. Uncertainty reduction in robust optimization. *Oper. Res. Lett.*, 55: article no. 107131, 2024.

4 Ayşe N. Arslan, Michael Poss, and Marco Silva. Min-Sup-Min Robust Combinatorial Optimization with Few Recourse Solutions. *INFORMS J. Comput.*, 34(4):2212–2228, 2022.

5 Josette Ayoub and Michael Poss. Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Comput. Manag. Sci.*, 13(2):219–239, 2016.

6 Aharon Ben-Tal, A. P. Goryashko, E. Guslitzer, and Arkadi Nemirovski. Adjustable robust solutions of uncertain linear programs. *Math. Program.*, 99(2):351–376, 2004.

7 Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. *Math. Oper. Res.*, 23(4):769–805, 1998.

8 Dimitris Bertsimas and Iain Dunning. Multistage Robust Mixed-Integer Optimization with Adaptive Partitions. *Oper. Res.*, 64(4):980–998, 2016.

9 Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Math. Program.*, 98(1-3):49–71, 2003.

10 Dimitris Bertsimas and Melvyn Sim. The Price of Robustness. *Oper. Res.*, 52(1):35–53, 2004.

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

12 Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *SIAM Rev.*, 59(1):65–98, 2017.

13 Matthew Colvin and Christos T. Maravelias. A stochastic programming approach for clinical trial planning in new drug development. *Comput. Chem. Eng.*, 32(11):2626–2642, 2008.

14 Iain Dunning, Joey Huchette, and Miles Lubin. JuMP: A Modeling Language for Mathematical Optimization. *SIAM Rev.*, 59(2):295–320, 2017.

15 Jacob Focke, Nicole Megow, and Julie Meißner. Minimum Spanning Tree under Explorable Uncertainty in Theory and Experiments. *ACM J. Exp. Algorithm.*, 25: article no. 1.14 (20 pages), 2020.

16 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`.

17 Marc Goerigk, Adam Kasperski, and Pawel Zielinski. Robust two-stage combinatorial optimization problems under convex second-stage cost uncertainty. *J. Comb. Optim.*, 43(3):497–527, 2022.

18 Marc Goerigk and Stefan Lendl. Robust Combinatorial Optimization with Locally Budgeted Uncertainty. *Open J. Math. Optim.*, 2: article no. 3 (18 pages), 2021.

19 Marc Goerigk, Stefan Lendl, and Lasse Wulf. Two-Stage robust optimization problems with two-stage uncertainty. *Eur. J. Oper. Res.*, 302(1):62–78, 2022.

20 Chrysanthos E. Gounaris, Wolfram Wiesemann, and Christodoulos A. Floudas. The Robust Capacitated Vehicle Routing Problem Under Demand Uncertainty. *Oper. Res.*, 61(3):677–693, 2013.

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

22 Tore W. Jonsbråten. Optimization models for petroleum field exploitation, 1998.

23 Tore W. Jonsbråten, Roger J. B. Wets, and David L. Woodruff. A class of stochastic programs withdecision dependent random elements. *Ann. Oper. Res.*, 82:83–106, 1998.

24 Nicolas Kämmerling and Jannis Kurtz. Oracle-based algorithms for binary two-stage robust optimization. *Comput. Optim. Appl.*, 77(2):539–569, 2020.

25 Panos Kouvelis and Gang Yu. Robust discrete optimization: past successes and future challenges. In *Robust discrete optimization and its applications*, pages 333–356. Springer, 1997.

26 Eva Ley, Rebecca Marx, Maximilian Merkert, Tim Niemann, and Sebastian Stiller. Robust Optimization Under Controllable Uncertainty. 2023. available at Optimization Online.

27 Thomas L. Magnanti and Laurence A. Wolsey. Optimal trees. In *Network models*, volume 7 of *Handbooks in Operations Research and Management Science*, pages 503–615. North-Holland, 1995.

28 Omid Nohadani and Kartikey Sharma. Optimization under decision-dependent uncertainty. *SIAM J. Optim.*, 28(2):1773–1795, 2018.

29 Rosario Paradiso. personnal communication, 2023.

30 Rosario Paradiso, Angelos Georghiou, Said Dabia, and Denise Tönissen. Exact and Approximate Schemes for Robust Optimization Problems with Decision Dependent Information Discovery. `https://arxiv.org/abs/2208.04115`, 2022.

31 Artur Alves Pessoa, Michael Poss, Ruslan Sadykov, and François Vanderbeck. Branch-Cut-and-Price for the Robust Capacitated Vehicle Routing Problem with Knapsack Uncertainty. *Oper. Res.*, 69(3):739–754, 2021.

32 Michael Poss. Robust combinatorial optimization with variable budgeted uncertainty. *4OR*, 11(1):75–92, 2013.

33 Michael Poss. Robust combinatorial optimization with knapsack uncertainty. *Discrete Optim.*, 27:88–102, 2018.

34 Krzysztof Postek and Dick den Hertog. Multistage Adjustable Robust Mixed-Integer Optimization via Iterative Splitting of the Uncertainty Set. *INFORMS J. Comput.*, 28(3):553–574, 2016.

35 William R. Pulleyblank. Faces Of Matching Polyhedra, 1973.

36 Maurice Queyranne and Andreas S. Schulz. *Polyhedral approaches to machine scheduling*. Citeseer, 1994.

37 Filipe Rodrigues, Agostinho Agra, Cristina Requejo, and Erick Delage. Lagrangian Duality for Robust Problems with Decomposable Functions: The Case of a Robust Inventory Problem. *INFORMS J. Comput.*, 33(2):685–705, 2021.

38 A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

39 Senay Solak, John-Paul B. Clarke, Ellis L. Johnson, and Earl R. Barnes. Optimization of R&D project portfolios under endogenous uncertainty. *Eur. J. Oper. Res.*, 207(1):420–433, 2010.

40 Simon A. Spacey, Wolfram Wiesemann, Daniel Kuhn, and Wayne Luk. Robust software partitioning with multiple instantiation. *INFORMS J. Comput.*, 24(3):500–515, 2012.

41 Anirudh Subramanyam, Chrysanthos E. Gounaris, and Wolfram Wiesemann. K-adaptability in two-stage mixed-integer robust optimization. *Math. Program. Comput.*, 12(2):193–224, 2020.

42 Leonardo Taccari. Integer programming formulations for the elementary shortest path problem. *Eur. J. Oper. Res.*, 252(1):122–130, 2016.

43 Bita Tadayon and J. Cole Smith. Algorithms and Complexity Analysis for Robust Single-Machine Scheduling Problems. *J. Sched.*, 18(6):575–592, 2015.

44 Phebe Vayanos, Angelos Georghiou, and Han Yu. Robust optimization with decision-dependent information discovery.

https://arxiv.org/abs/2004.08490v1, 2020.
45 Phebe Vayanos, Angelos Georghiou, and Han Yu. Robust optimization with decision-dependent information discovery. https://arxiv.org/abs/2004.08490v3, 2022.
46 Laurence A. Wolsey. *Integer programming.* John Wiley & Sons, 2020.
47 Hande Yaman. Short Paper - A Note on Robust Combinatorial Optimization with Generalized Interval Uncertainty. *Open J. Math. Optim.*, 4: article no. 4 (7 pages), 2023.
48 Ihsan Yanikoglu, Bram L. Gorissen, and Dick den Hertog. A survey of adjustable robust optimization. *Eur. J. Oper. Res.*, 277(3):799–813, 2019.
49 Bo Zeng and Long Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Oper. Res. Lett.*, 41(5):457–461, 2013.
50 Jianzhe Zhen, Dick den Hertog, and Melvyn Sim. Adjustable robust optimization via Fourier–Motzkin elimination. *Oper. Res.*, 66(4):1086–1100, 2018.