



HAL
open science

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

Jérémy Omer, Michael Poss

► **To cite this version:**

Jérémy Omer, Michael Poss. Combinatorial Robust Optimization with Decision-Dependent Information Discovery and Polyhedral Uncertainty. 2023. hal-04097679v1

HAL Id: hal-04097679

<https://hal.science/hal-04097679v1>

Preprint submitted on 15 May 2023 (v1), last revised 22 Dec 2023 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

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

JÉRÉMY OMER

Univ Rennes, INSA Rennes, CNRS, IRMAR - UMR 6625, 35000 Rennes, France

MICHAEL POSS

LIRMM, University of Montpellier, CNRS, France

May 15, 2023

Abstract

Given a nominal combinatorial optimization problem, we consider a robust two-stages variant with cost uncertainty, called 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. We further assume that the uncertainty set is a polytope defined by upper bounds and a constant number of knapsack constraints. Given a compact linear programming formulation for the nominal problem, we provide a compact mixed-integer linear programming formulation for DDID. The formulation leads to polynomial-time algorithms for DDID when the number of possible observations is polynomially bounded. 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.

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 param-

eters, thereby reducing the total amount of uncertainty. The model has innumerable applications in urban planning, project management, resource allocations, scheduling, among many others. The first DDID models were motivated by applications in offshore oilfield exploitation [Jonsbråten, 1998] and production planning [Jonsbråten et al., 1998]. Subsequent examples have been considered in the literature and Vayanos et al. [2022] detail applications in a R&D project portfolio optimization problem, where a company must choose how to prioritize the projects in its pipeline [Solak et al., 2010, Colvin and Maravelias, 2008]. Vayanos et al. [2022] also describe a preference elicitation with real-valued recommendations where one can investigate how much users like any particular item. Vayanos et al. [2022] further apply the latter model to improve the US kidney allocation system. Even more recently, Paradiso et al. [2022] 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 by Paradiso et al. [2022] 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.

- $\mathcal{W} = \{w \in \{0, 1\}^n \mid Gw \leq g\}$ is the set characterizing the possible information discovery;
- $\Xi = \{\xi \in \mathbb{R}^n \mid A\xi \leq f\}$ is an uncertainty polytope defined by a given matrix A and right-hand-side f ;
- $\mathcal{Y} = \{y \in \mathbb{Z}^n \mid By \geq b, 0 \leq y \leq 1\}$ is the feasible set of a given combinatorial optimization problem, defined by matrix B and right-hand-side b ;
- $\mathcal{P} = \{y \in \mathbb{R}^n \mid By \geq b, 0 \leq y \leq 1\}$ is the relaxed polytope of \mathcal{Y} .

The DDID problem we consider is then defined by:

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

where c is a given cost vector and $\Xi(w, \bar{\xi}) = \{\xi \in \Xi \mid w \circ \xi = w \circ \bar{\xi}\}$, where $v \circ w = (v_1 w_1, \dots, v_n w_n)$ 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.

We further denote $\Phi(w)$ the outermost objective function, namely

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

If it is possible to observe every cost coefficient, i.e., if $(1, \dots, 1) \in \mathcal{W}$, it is trivially the 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, denoted $\mathbf{1}$.*

Well-known complexity results in robust combinatorial optimization [Kouvelis and Yu, 1997] imply that computing Φ is \mathcal{NP} -hard even when optimizing over \mathcal{Y} is easy. Therefore, we focus in this paper on knapsack uncertainty polytopes, which model Ξ with non-negativity constraints, upper bounds, and a small number of additional “knapsack” constraints (defined by non-negative coefficients). To simplify the presentation that follows, we assume throughout the presence of a single such constraint, represented by the coefficient vector $a > 0$ and right-hand-side $r > 0$. Notice that for a single knapsack constraint, we can assume $a > 0$ without loss of generality, because ξ_i is equal to its upper bound if $a_i = 0$.

Assumption 2. *We assume that Ξ is a knapsack polytope such that $A = \begin{pmatrix} a^T \\ \text{Id} \\ -\text{Id} \end{pmatrix}$ where $a_i > 0, \forall i \in [n]$ and Id is the identity matrix. We further denote the right-hand-sides by $\begin{pmatrix} r \\ d \\ 0 \end{pmatrix}$, with $r > 0$ and $d > 0$.*

Knapsack uncertainty is a relevant choice for addressing complex robust problems as it provides a good trade-off between its modelling capability and the computational complexity of the resulting optimization problems. On the one hand, the polytope is easy to apprehend by decision makers, and they can adapt the uncertainty budget r to their risk aversion, as illustrated by Pessoa et al. [2021]. On the other hand, the polytope often leads to more tractable optimization problems than arbitrary polytopes, both from the theoretical and numerical viewpoints (see more details in Section 1.1). Last, knapsack uncertainty is also motivated by probabilistic reasons, since it leads to safe approximations of otherwise hard probabilistic constraints, see [Bertsimas and Sim, 2004] and the extension to decision-dependent uncertainty by Poss [2013].

We detail next the important role of c in the objective function when using knapsack uncertainty sets, which complements the discussion on down-monotone completion by Poss [2018]. In fact, the following remark has implications beyond (DDID), and one should bear it in mind when addressing min-max robust problems.

Remark 1. *It is well-known (e.g. Poss [2018]) that one can replace Ξ by its down-monotone completion $\text{dm}(\Xi) = \{\zeta \in \mathbb{R}_+^n : \exists \xi \in \Xi \text{ such that } \zeta \leq \xi\}$ without affecting the optimal solution of a min-max robust problem, and the same applies to (DDID). This might suggest that we could remove the constant term c from the objective of the optimization problems without loss of generality. The difficulty with this removal is that the resulting down-monotone completion may require many more knapsack constraints. Let us illustrate this in the 3-dimensional example where $c = (1, 1, 1)$, and $\Xi = \{\xi \in [0, 1]^n \mid \xi_1 + \xi_2 + \xi_3 \leq 1\}$ is the 3-dimensional simplex. Removing the constant term in the objective would lead to considering the uncertainty set $\Xi' = c + \Xi$, defined as $\{\xi \in [1, 2]^n \mid (\xi_1 - 1) + (\xi_2 - 1) + (\xi_3 - 1) \leq 1\}$. While Ξ' is not a knapsack uncertainty set, its down-monotone completion is, and is given by*

$$\text{dm}(\Xi') = \{\zeta \in \mathbb{R}^n \mid \zeta_1 + \zeta_2 + \zeta_3 \leq 4, \zeta_1 + \zeta_2 \leq 3, \zeta_1 + \zeta_3 \leq 3, \zeta_2 + \zeta_3 \leq 3, 0 \leq \zeta \leq 2\},$$

having 4 knapsack constraints instead of 1 in the original description.

1.1 Literature Review

We contextualize next how (DDID) fits within the robust optimization landscape. Robust combinatorial optimization introduced by Kouvelis and Yu [1997] originally considered min-max optimization problems of the form

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

for discrete uncertainty sets Ξ . They proved in particular that (MIN-MAX) is \mathcal{NP} -hard even when Ξ consists of two points only 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 Kouvelis and Yu [1997], Ben-Tal and Nemirovski [1998] 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 [Bertsimas and Sim, 2003, 2004] and its extension to more general knapsack constraints [Poss, 2018]. Extending the seminal result of Bertsimas and Sim [2003], Poss [2018] showed that if the number of knapsack constraints (constraints different than upper bounds on ξ) characterizing Ξ is constant, then the min-max robust counterparts of polynomial problems remains polynomial, contrasting with the difficulty proved by Kouvelis and Yu [1997] for arbitrary sets. These rather theoretical results have been pursued for specific variants of the set [Goerigk et al., 2022b] and complemented by efficient algorithms that leverage the structure of the set, e.g. for vehicle routing [Gounaris et al., 2013, Pessoa et al., 2021], scheduling [Tadayon and Smith, 2015], lot-sizing [Agra et al., 2016] and inventory routing [Bertsimas and Thiele, 2006, Rodrigues et al., 2021], only to name a few.

After the basic robust models were introduced by Kouvelis and Yu [1997] and Ben-Tal and Nemirovski [1998], 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 [Nohadani and Sharma, 2018, Poss, 2013, Spacey et al., 2012]. 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 Yanikoglu et al. [2019]), providing exact [Ayoub and Poss, 2016, Zeng and Zhao, 2013, Zhen et al., 2018] or approximate solutions [Ben-Tal et al., 2004] in the case of fractional recourse. The case of integer recourse is particularly difficult and, apart from the recent exact algorithms by Arslan and Detienne [2022], Kämmerling and Kurtz [2020], 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 [Arslan et al., 2022, Subramanyam et al., 2020], Hanasusanto et al. [2015] were able to provide a compact reformulation for K -adaptability when only cost is uncertain. Bertsimas and Dunning [2016], Postek and den Hertog [2016] have also proposed to partition Ξ dynamically and heuristically. While the above references aim at solving generic problems, Goerigk et al. [2022a] have focused on specific problems and proposed tailored algorithms and complexity results.

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.

1.2 Contributions and structure of the paper

Our main result is a compact linear relaxations for $\Phi(w)$ that is exact whenever $\text{conv}(\mathcal{Y}) = \mathcal{P}$, where $\text{conv}(\mathcal{S})$ denotes the convex hull of any set \mathcal{S} . The linear programming formulation is then dualized and linearized to provide a compact mixed-integer linear programming reformulations for (DDID). An ad-hoc study is carried out to strengthen significantly the linearized MILP. Our results extend to a polytope Ξ described by a larger number of knapsack constraints. However, similarly to min-max robust combinatorial optimization [Poss, 2018], one should bear in mind that the dimensions of the resulting formulations will increase exponentially in that number.

Then, we discuss how these reformulations can be extended to 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. They also illustrate the efficiency of our heuristic column generation-based algorithm and exact cutting-plane algorithm.

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 [Vayanos et al., 2022]. 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 $\text{conv}(\mathcal{Y}) \subset \mathcal{P}$.

The rest of the paper is structured as follows. We detail in the next section the relationship between (DDID) and its min-max and max-min counterparts. 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 the proofs of the linearization of $\Phi(w)$ and the dominance relationships used in the column generation algorithms. It provides also the counterpart of our reformulations for maximization problems, the reformulation proposed by Vayanos et al. [2022] for K -adaptability, and additional statistics on the algorithms applied to the orienteering problem.

2. Preliminary observations and trivial cases

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

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

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

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

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

Example 1. Consider an instance of (DDID) where $\mathcal{Y} = \{y \in \{0, 1\}^5 \mid \sum_{i \in [5]} y_i = 1\}$ is the selection feasibility set, $\mathcal{W} = \{w \in \{0, 1\}^5 \mid \sum_{i \in [5]} w_i = 1\}$ amounts to choose one item among 5 and the uncertainty is the budget uncertainty set from Bertsimas and Sim [2003] with nominal values $c = (1, 2, 3, 4, 5)$ and deviations $d = (5, 4, 3, 2, 1)$, that is, $\Xi = \{\xi \in \mathbb{R}^5 \mid \sum_{i \in [5]} \frac{\xi_i}{d_i} \leq 1, 0 \leq \xi_i \leq d_i, i \in [5]\}$.

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 can decide whether to select an 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 $z^{\text{DDID}} = 34/9 \approx 3.8$, and 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 polyhedron, \mathcal{P} . In this context, it is well-known that

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

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 1. *If $\mathcal{Y} = \mathcal{P}$, then $z^{\text{WS}} = z^{\text{DDID}} = z^{\text{MM}}$.*

Observation 1 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 \mathcal{Y} rather than \mathcal{P} 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 $A^+ \xi \leq a^+, A^- \xi \leq a^-$ 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 $\dim(\Xi) = n - \text{rank}(A^-)$. 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 2. *Let $w \in \mathcal{W}$, $\bar{\xi} \in \Xi$ and $\mathcal{W}^1 = \{i \in [n] \mid w_i = 1\}$, then*

$$\dim(\Xi(w, \bar{\xi})) \leq n - \text{rank} \begin{pmatrix} A^- \\ E_{\mathcal{W}^1} \end{pmatrix}.$$

Proof. By definition, $\Xi(w, \bar{\xi}) = \{\xi \in \mathbb{R}^n \mid A^+ \xi \leq a^+, A^- \xi \leq a^-, E_{\mathcal{W}^1} \xi = E_{\mathcal{W}^1} \bar{\xi}\}$. We know that for all $\xi \in \Xi(w, \bar{\xi})$, $\begin{pmatrix} A^- \\ E_{\mathcal{W}^1} \end{pmatrix} \xi = \begin{pmatrix} a^- \\ E_{\mathcal{W}^1} \bar{\xi} \end{pmatrix}$, so $\dim(\Xi(w, \bar{\xi})) \leq n - \text{rank} \begin{pmatrix} A^- \\ E_{\mathcal{W}^1} \end{pmatrix}$. \square

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

$$\mathcal{W}^{\text{sel}} = \left\{ w \in \{0, 1\}^n \mid \sum_{i \in [n]} w_i \leq q \right\},$$

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 $\begin{pmatrix} A^- \\ E_{\mathcal{W}^1} \end{pmatrix}$ corresponds to

the completion of the rows of A^\ominus with row vectors of the canonical basis of \mathbb{R}^n . We thus know that we may choose $w \in \mathcal{W}^{\text{sel}}$ such that $\text{rank} \begin{pmatrix} A^\ominus \\ E_{\mathcal{W}^1} \end{pmatrix} = \text{rank}(A^\ominus) + q$ if $\text{rank}(A^\ominus) > n - q$, and $\text{rank} \begin{pmatrix} A^\ominus \\ E_{\mathcal{W}^1} \end{pmatrix} = n$ if $\text{rank}(A^\ominus) \leq n - q$. We deduce the following.

Corollary 1. *There is $w \in \mathcal{W}^{\text{sel}}$ such that, for any $\bar{\xi} \in \Xi$, $\dim(\Xi(w, \bar{\xi})) \leq \max\{0, \dim(\Xi) - q\}$.*

Interestingly, Corollary 1 implies that picking $w^* \in \mathcal{W}^{\text{sel}}$ (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 3. *If $\mathcal{W} = \mathcal{W}^{\text{sel}}$ and $\dim(\Xi) \leq q$, then $z^{\text{DDID}} = z^{\text{WS}}$.*

Proof. Corollary 1 implies that there exists $w^* \in \mathcal{W}^{\text{sel}}$ such that $\Xi(w^*, \bar{\xi}) = \{\bar{\xi}\}, \forall \bar{\xi} \in \Xi$. Hence, $\Phi(w^*) = \max_{\bar{\xi} \in \Xi} \min_{y \in \mathcal{Y}} \sum_{i \in [n]} (c_i + \bar{\xi}_i) y_i = z^{\text{WS}}$ \square

We conclude the section by discussing how to solve (WAIT&SEE). First, if $\text{conv}(\mathcal{Y}) = \mathcal{P}$, then one can dualize the minimization problem, yielding a compact linear program. Interestingly, swapping the minimization and maximization in this case also shows that z^{WS} is actually equal to the optimal value of the linear relaxation of (MIN-MAX). Second, the case $\text{conv}(\mathcal{Y}) \subset \mathcal{P}$ can be handled by constraint generation, using the epigraphic reformulation

$$\max \left\{ \omega \mid \xi \in \Xi, \omega \leq \sum_{i \in [n]} (c_i + \xi_i) y_i, \forall y \in \mathcal{Y} \right\}, \quad (3)$$

and separating the constraints indexed by \mathcal{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

$$\Phi(w) = \max \omega \tag{4}$$

$$\text{s.t. } \omega \leq \min_{y \in \mathcal{Y}} \max_{\xi \in \Xi(w, \bar{\xi})} \sum_{i \in [n]} (c_i + \xi_i) y_i \tag{5}$$

$$\bar{\xi} \in \Xi. \tag{6}$$

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

$$\max_{\xi \in \Xi(w, \bar{\xi})} \xi^T y = \min_{(\nu, \pi, \gamma) \in \mathcal{D}(y)} r\nu + d^T \pi + (w \circ \bar{\xi})^T \gamma = \min_{(\nu, \pi, \gamma) \in \text{ext}(\mathcal{D}(y))} r\nu + d^T \pi + (w \circ \bar{\xi})^T \gamma, \tag{7}$$

where $\mathcal{D}(y) = \{(\nu, \pi, \gamma) \in \mathbb{R}^{2n+1} \mid a_i \nu + \pi_i + w_i \gamma_i \geq y_i, i \in [n]\}$ is the dual polytope and $\text{ext}(\mathcal{D}(y))$ the set of its extreme points. Plugging (7) into the right-hand side of (5) leads to reformulating $\Phi(w)$ as a linear program with many constraints

$$\max \omega \tag{8}$$

$$\text{s.t. } \omega \leq r\nu + d^T \pi + (w \circ \bar{\xi})^T \gamma, \quad \forall y \in \mathcal{Y}, (\nu, \pi, \gamma) \in \text{ext}(\mathcal{D}(y)) \tag{9}$$

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

We now study the complexity of the separation problem associated with constraints (9). Using (7) in the reverse direction, we see that a given candidate solution $(\omega^*, \bar{\xi}^*) \in \mathbb{R} \times \Xi$ is feasible for (9) if and only if

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

Thus, checking whether $(\omega^*, \bar{\xi}^*)$ is feasible amounts to solving a problem with same form as (MIN-MAX). Observing that $\Xi(w, \bar{\xi}^*)$ is also a knapsack uncertainty set, Theorem 3 from Poss [2018] implies that the right-hand-side of (11) can be computed by solving $\min_{y \in \mathcal{Y}} \xi^T y$ for at most $n + 1$ 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 [Schrijver, 2003], we obtain a polynomial

algorithm for computing $\Phi(w)$ whenever the nominal counterpart of the problem is polynomially solvable.

Proposition 1. *If the nominal counterpart is polynomially solvable, then $\Phi(w)$ can be computed in polynomial time.*

Corollary 2. *If $|\mathcal{W}|$ is polynomially bounded and the nominal counterpart is polynomially solvable, 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. □

We observe that the above algorithm is reminiscent of the one developed by Paradiso et al. [2022] to compute $\Phi(w)$, with two notable differences. First, the dualized formulation (8)–(10) requires only constraint generation, while the approach from Paradiso et al. [2022] relies on the linearization

$$\max \omega \tag{12}$$

$$\text{s.t. } \omega \leq \sum_{i \in [n]} (c_i + \xi_i(y)) y_i, \forall y \in \mathcal{Y} \tag{13}$$

$$\xi(y) \in \Xi(w, \bar{\xi}), \forall y \in \mathcal{Y} \tag{14}$$

$$\bar{\xi} \in \Xi, \tag{15}$$

where $\xi(y)$ plays the role of adjustable variables depending on y . Hence, Paradiso et al. [2022] generate constraints (13) and (14) as well as variables $\xi(y)$ in the course of their algorithm.

Second, we leverage the knapsack structure of $\Xi(w, \bar{\xi}^*)$ to reduce the separation to solving $n + 1$ nominal problems, while Paradiso et al. [2022] address the problem through MILP formulations. These two differences have a theoretical impact, since the running time of the algorithm from Paradiso et al. [2022] cannot be polynomially bounded in general under the assumptions of Corollary 2. From the numerical viewpoint, the supremacy of one algorithm over the other will depend on the sets \mathcal{Y} and Ξ .

Corollary 2 implies, for instance, that if \mathcal{P} is the matching polytope [Pulleyblank, William R., 1973] (see also [Schrijver, 2003, 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. [Schrijver, 2003, 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 Magnanti and Wolsey [1995]), or minimizing the weighted sum of completion times (e.g. [Queyranne and Schulz, 1994, 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 solve a compact linear program.

In what follows, we let \mathcal{W}^1 denote the set of indices over which w is equal to 1, and \mathcal{W}^0 its complementary. We also define $[n]_0 = [n] \cup \{0\}$. We first introduce an adaptation of a classical result from the literature [Bertsimas and Sim, 2003, Poss, 2018] that essentially shows how solving the maximization problem in (MIN-MAX) amounts to take the best out of $n + 1$ different objective functions. We underline that, while the number $n + 1$ could be reduced to $|\mathcal{W}^0| + 1$, carrying \mathcal{W}^0 throughout would lead to additional non-linearities in the next section. This contrasts with classical results that focus on reducing the number of subproblems as much as possible [Lee and Kwon, 2014].

Proposition 2. *Let $y \in \{0, 1\}^n$. We have*

$$\max_{\xi \in \Xi(w, \bar{\xi})} \sum_{i \in \mathcal{W}^0} (c_i + \xi_i) y_i = \min_{\ell \in [n]_0} \left\{ \left(r - \sum_{i \in \mathcal{W}^1} a_i \bar{\xi}_i \right) \alpha_\ell + \sum_{i \in \mathcal{W}^0} (c_i + \beta_{\ell, i}) y_i \right\},$$

where for each $\ell \in [n]$, $\alpha_\ell = 1/a_\ell$, and for each $i, \ell \in [n]$, $\beta_{\ell, i} = d_i [1 - a_i/a_\ell]^+$, while $\alpha_0 = 0$ and $\beta_0 = d$.

Proof. Observe first that

$$\max_{\xi \in \Xi(w, \bar{\xi})} \sum_{i \in \mathcal{W}^0} (c_i + \xi_i) y_i = \max_{\xi' \in \Xi'} \sum_{i \in \mathcal{W}^0} (c_i + \xi'_i) y_i, \quad (16)$$

where

$$\Xi' = \left\{ \xi' \in \mathbb{R}^{|\mathcal{W}^0|} \mid \sum_{i \in \mathcal{W}^0} a_i \xi'_i \leq r - \sum_{i \in \mathcal{W}^1} a_i \bar{\xi}_i, 0 \leq \xi'_i \leq d_i, i \in \mathcal{W}^0 \right\}$$

is the projection of $\Xi(w, \bar{\xi})$ into the subset of coordinates indexed by \mathcal{W}^0 . Let us denote $r - \sum_{i \in \mathcal{W}^1} a_i \bar{\xi}_i$ by \bar{r} . Notice that $\bar{\xi} \in \Xi$, implying that $\bar{r} \geq 0$, and by Assumption 1, $|\mathcal{W}^1| < n$. Hence, Ξ' is non-empty. Let us denote the dual variables of the constraints of Ξ' by α and π , respectively.

Dualizing the maximization problem in the right-hand-side of (16) and substituting π_i yields

$$\max_{\xi' \in \Xi'} \sum_{i \in \mathcal{W}^0} (c_i + \xi'_i) y_i = \sum_{i \in \mathcal{W}^0} c_i y_i + \min \left\{ \bar{r} \alpha + \sum_{i \in \mathcal{W}^0} d_i \pi_i \mid a_i \alpha + \pi_i \geq y_i, \forall i \in \mathcal{W}^0, \alpha, \pi \geq 0 \right\} \quad (17)$$

$$= \sum_{i \in \mathcal{W}^0} c_i y_i + \min \left\{ \bar{r} \alpha + \sum_{i \in \mathcal{W}^0} d_i [y_i - a_i \alpha]^+ \mid \alpha \geq 0 \right\}. \quad (18)$$

Notice next that $y_i \in \{0, 1\}$ for each $i \in \mathcal{W}^0$, so we can reformulate the objective of the minimization problem to make it linear in variables y

$$\sum_{i \in \mathcal{W}^0} c_i y_i + \min \left\{ \bar{r} \alpha + \sum_{i \in \mathcal{W}^0} y_i d_i [1 - a_i \alpha]^+ \mid \alpha \geq 0 \right\} \quad (19)$$

The objective function to be minimized in (19) is a piece-wise linear function of α , whose knickpoints are included in $0 \cup \{1/a_i, i \in [n]\}$, yielding the result. Notice that unlike Bertsimas and Sim [2003], Poss [2018], we take here a superset of the knickpoints as this leads to more efficient formulations in the next section. \square

We are now ready to prove the main result of this section.

Theorem 1. Let $w \in \mathcal{W}$. If $\text{conv}(\mathcal{Y}) = \mathcal{P}$, then

$$\Phi(w) = \begin{cases} \max & \omega \\ \text{s.t.} & \omega \leq \left(r - \sum_{i \in \mathcal{W}^1} a_i \bar{\xi}_i \right) \alpha_\ell + b^T \lambda_\ell - \sum_{i \in [n]} \pi_{\ell,i}, \forall \ell \in [n]_0 \\ & (B_{\cdot,i})^T \lambda_\ell - \pi_{\ell,i} \leq c_i + \bar{\xi}_i, \forall \ell \in [n]_0, \forall i \in \mathcal{W}^1 \\ & (B_{\cdot,i})^T \lambda_\ell - \pi_{\ell,i} \leq c_i + \beta_{\ell,i}, \forall \ell \in [n]_0, \forall i \in \mathcal{W}^0 \\ & \lambda_\ell, \pi_\ell \geq 0, \forall \ell \in [n]_0 \\ & \bar{\xi} \in \Xi \end{cases} \quad (20)$$

where for each $\ell \in [n]$, $\alpha_\ell = 1/a_\ell$, and for each $i, \ell \in [n]$, $\beta_{\ell,i} = d_i [1 - a_i/a_\ell]^+$, while $\alpha_0 = 0$ and $\beta_{i,0} = d_i, \forall i \in [n]$.

Proof. Observe that

$$\max_{\xi \in \Xi(w, \bar{\xi})} \sum_{i \in [n]} \xi_i y_i = \sum_{i \in \mathcal{W}^1} \bar{\xi}_i y_i + \max_{\xi \in \Xi(w, \bar{\xi})} \sum_{i \in \mathcal{W}^0} \xi_i y_i. \quad (21)$$

Applying (21) to the epigraphic reformulation (4)–(6) presented previously yields

$$\max \omega \quad (22)$$

$$\text{s.t. } \omega \leq \min_{y \in \mathcal{Y}} \sum_{i \in \mathcal{W}^1} (c_i + \bar{\xi}_i) y_i + \max_{\xi \in \Xi(w, \bar{\xi})} \sum_{i \in \mathcal{W}^0} (c_i + \xi_i) y_i \quad (23)$$

$$\bar{\xi} \in \Xi. \quad (24)$$

The main idea of the proof that follows reformulates (23) through two ingredients: we reformulate the maximization over ξ using Proposition 2 (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 Proposition 2 to the last term of (23) yields

$$\min_{\ell \in [n]_0} \left\{ \left(r - \sum_{i \in \mathcal{W}^1} a_i \bar{\xi}_i \right) \alpha_\ell + \sum_{i \in \mathcal{W}^0} (c_i + \beta_{\ell,i}) y_i \right\}.$$

Plugging the above expression into the right-hand-side of (23) and swapping the minimizations, we

obtain

$$(23) \Leftrightarrow \omega \leq \min_{\ell \in [n]_0} \left\{ \left(r - \sum_{i \in \mathcal{W}^1} a_i \bar{\xi}_i \right) \alpha_\ell + \min_{y \in \mathcal{Y}} \sum_{i \in \mathcal{W}^1} (c_i + \bar{\xi}_i) y_i + \sum_{i \in \mathcal{W}^0} (c_i + \beta_{\ell,i}) y_i \right\}.$$

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

$$(23) \Leftrightarrow \omega \leq \left(r - \sum_{i \in \mathcal{W}^1} a_i \bar{\xi}_i \right) \alpha_\ell + \min_{y \in \mathcal{Y}} \sum_{i \in \mathcal{W}^1} (c_i + \bar{\xi}_i) y_i + \sum_{i \in \mathcal{W}^0} (c_i + \beta_{\ell,i}) y_i, \forall \ell \in [n]_0. \quad (25)$$

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 $\mathcal{P} = \{y \in \mathbb{R}^n \mid By \geq b, 0 \leq y \leq 1\}$ in each constraint of (25). For each constraint $\ell \in [n]_0$, we then define the dual variables λ_ℓ and π_ℓ 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:

$$\begin{aligned} \omega \leq & \left(r - \sum_{i \in \mathcal{W}^1} a_i \bar{\xi}_i \right) \alpha_\ell + \max_{\lambda_\ell} b^T \lambda_\ell - \sum_{i \in [n]} \pi_{\ell,i} \\ \text{s.t.} & (B_{\cdot,i})^T \lambda_\ell - \pi_{\ell,i} \leq c_i + \bar{\xi}_i, \forall i \in \mathcal{W}^1 \\ & (B_{\cdot,i})^T \lambda_\ell - \pi_{\ell,i} \leq c_i + \beta_{\ell,i}, \forall i \in \mathcal{W}^0 \\ & \lambda_\ell, \pi_\ell \geq 0 \end{aligned}$$

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

$$\begin{aligned} \omega \leq & \left(r - \sum_{i \in \mathcal{W}^1} a_i \bar{\xi}_i \right) \alpha_\ell + b^T \lambda_\ell - \sum_{i \in [n]} \pi_{\ell,i} \\ & (B_{\cdot,i})^T \lambda_\ell - \pi_{\ell,i} \leq c_i + \bar{\xi}_i, \forall i \in \mathcal{W}^1 \\ & (B_{\cdot,i})^T \lambda_\ell - \pi_{\ell,i} \leq c_i + \beta_{\ell,i}, \forall i \in \mathcal{W}^0 \\ & \lambda_\ell, \pi_\ell \geq 0 \end{aligned}$$

Replacing (23) with the corresponding $n + 1$ sets of constraints provides the result. \square

4. Solving the full problem

We describe below the compact reformulations obtained for (DDID) by dualizing and linearizing the formulation proposed in Theorem 1 using classical techniques.

Proposition 3. *If $\text{conv}(\mathcal{Y}) = \mathcal{P}$, then (DDID) is equivalent to*

$$\begin{aligned}
\min \quad & \sum_{\ell \in [n]_0} \left(r\alpha_\ell u_\ell + \sum_{i \in [n]} c_i y_{\ell,i} + \sum_{i \in [n]} \beta_{\ell,i} y_{\ell,i}^0 \right) + \sum_{i \in [n]} d_i \sigma_i + r\mu \\
\text{s.t.} \quad & \sum_{\ell \in [n]_0} u_\ell = 1 \\
& a_i \mu + \sigma_i \geq -a_i \sum_{\ell \in [n]_0} \alpha_\ell u_\ell + \sum_{\ell \in [n]_0} y_{\ell,i} - (1 - w_i), \quad \forall i \in [n] \\
& B y_\ell \geq u_\ell b, \quad \forall \ell \in [n]_0 \\
& y_{\ell,i} \leq u_\ell, \quad \forall \ell \in [n]_0, i \in [n] \\
& y_{\ell,i}^0 \geq y_{\ell,i} - w_i, \quad \forall \ell \in [n]_0, i \in [n] \\
& w \in \mathcal{W} \\
& u, y, y^0, \mu, \sigma \geq 0.
\end{aligned} \tag{DDID-WIP}$$

Proof. See Appendix A.1. □

We explain next how to improve formulation (DDID-WIP) in two aspects. First, writing down the complementarity conditions between (20) and its dual shows that $\mu > 0$ corresponds to the consumption of all r on the investigated costs, leaving no possible deviation after the decisions on y . Stated otherwise, $\mu > 0$ leads to a suboptimal dual solution since the uncertainty is completely unveiled before the solution of the combinatorial optimization problem, which means that the optimal value is that of (WAIT&SEE).

Proposition 4. *There is an optimal solution $(w^*, u^*, y^*, \mu^*, \sigma^*)$ to the compact formulation (DDID-WIP) such that $\mu^* = 0$.*

Proof. See Appendix A.2. □

Second, 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 on the two sets \mathcal{W}^1 and \mathcal{W}^0 .

Proposition 5. *If $\text{conv}(\mathcal{Y}) = \mathcal{P}$, then (DDID) is equivalent to*

$$\begin{aligned}
\min \quad & \sum_{\ell \in [n]_0} \left(r\alpha_\ell u_\ell + \sum_{i \in [n]} c_i y_{\ell,i} + \sum_{i \in [n]} \beta_{\ell,i} y_{\ell,i}^0 \right) + \sum_{i \in [n]} d_i \sigma_i \\
\text{s.t.} \quad & \sum_{\ell \in [n]_0} u_\ell = 1 \\
& \sigma_i \geq -a_i \sum_{\ell \in [n]_0} \alpha_\ell u_{\ell,i}^1 + \sum_{\ell \in [n]_0} y_{\ell,i}^1, & \forall i \in [n], \\
& B y_\ell \geq u_\ell b, & \forall \ell \in [n]_0, \\
& y_{\ell,i} \leq u_\ell, & \forall \ell \in [n]_0, i \in [n] \\
& u_\ell = u_{\ell,i}^0 + u_{\ell,i}^1, & \forall \ell \in [n]_0, i \in [n], \\
& \sum_{\ell \in [n]_0} u_{\ell,i}^0 \leq 1 - w_i, & \forall i \in [n], \quad \text{(DDID-SIP)} \\
& \sum_{\ell \in [n]_0} u_{\ell,i}^1 \leq w_i, & \forall i \in [n], \\
& G u_\ell^1 \leq u_\ell g & \forall \ell \in [n]_0, \\
& y_{\ell,i} = y_{\ell,i}^0 + y_{\ell,i}^1, & \forall \ell \in [n]_0, i \in [n] \\
& y_{\ell,i}^0 \leq u_{\ell,i}^0, & \forall \ell \in [n]_0, i \in [n] \\
& y_{\ell,i}^1 \leq u_{\ell,i}^1, & \forall \ell \in [n]_0, i \in [n] \\
& w \in \mathcal{W} \\
& u, u^0, u^1, y, y^0, y^1, \sigma \geq 0.
\end{aligned}$$

Proof. See Appendix A.3. □

We show below that the formulation of Proposition 5 is in general stronger than the one provided by Proposition 3. Numerical evidence shows that the inclusion may hold strictly.

Proposition 6. *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 A.4. □

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 $\mathcal{Y} = \{\tilde{y}_1, \dots, \tilde{y}_t\}$. Introducing the convex multipliers $\lambda_1^\ell, \dots, \lambda_t^\ell$ for each $\ell \in [n]_0$, we can substitute y_ℓ with $\sum_{s \in [t]} \lambda_{\ell,s} \tilde{y}_s$, and the constraints $By_\ell \geq u_\ell b$ with the convexification constraints, so (DDID-SIP) becomes

$$\begin{aligned}
\min \quad & \sum_{\ell \in [n]_0} \left(r\alpha_\ell u_\ell + \sum_{i \in [n]} \sum_{s \in [t]} c_i \lambda_{\ell,s} \tilde{y}_{s,i} + \sum_{i \in [n]} \beta_{\ell,i} y_{\ell,i}^0 \right) + \sum_{i \in [n]} d_i \sigma_i \\
\text{s.t.} \quad & \sum_{s \in [t]} \lambda_{\ell,s} = u_\ell, \quad \forall \ell \in [n]_0, & [\nu^\ell] \\
& \sum_{s \in [t]} \lambda_{\ell,s} \tilde{y}_{s,i} = y_{\ell,i}^0 + y_{\ell,i}^1, \quad \forall \ell \in [n]_0, i \in [n], & [\rho_i^\ell] \\
& \sum_{\ell \in [n]_0} u_\ell = 1 \\
& \sigma_i \geq -a_i \sum_{\ell \in [n]_0} \alpha_\ell u_{\ell,i}^1 + \sum_{\ell \in [n]_0} y_{\ell,i}^1, \quad \forall i \in [n], \\
& u_\ell \leq 1 - w_\ell, \quad \forall \ell \in [n], \\
& u_\ell = u_{\ell,i}^0 + u_{\ell,i}^1, \quad \forall \ell \in [n]_0, i \in [n], \\
& \sum_{\ell \in [n]_0} u_{\ell,i}^0 \leq 1 - w_i, \quad \forall i \in [n], \\
& \sum_{\ell \in [n]_0} u_{\ell,i}^1 \leq w_i, \quad \forall i \in [n], \\
& Gu_\ell^1 \leq u_\ell g, \quad \forall \ell \in [n]_0, \\
& y_{\ell,i}^0 \leq u_{\ell,i}^0, \quad \forall \ell \in [n]_0, i \in [n] \\
& y_{\ell,i}^1 \leq u_{\ell,i}^1, \quad \forall \ell \in [n]_0, i \in [n] \\
& w \in \mathcal{W} \\
& u, u^0, u^1, y^0, y^1, \sigma, \lambda \geq 0.
\end{aligned} \tag{DDID-CG}$$

where the right brackets denote the dual variables. 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, one may further reduce the value of t by observing that only minimal \tilde{y} , with respect to inclusion, need to be considered.

Observation 4. Consider $\tilde{y}_{s_1}, \tilde{y}_{s_2} \in \mathcal{Y}$ such that $\tilde{y}_{s_1} \leq \tilde{y}_{s_2}$ and let $\tilde{\mathcal{Y}} = \mathcal{Y} \setminus \{\tilde{y}_{s_2}\}$. Let \tilde{z}^{DDID} denote the optimal value of (DDID-CG) associated to $\tilde{\mathcal{Y}}$. We have $z^{\text{DDID}} = \tilde{z}^{\text{DDID}}$.

Proof. See Appendix B.1. □

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

Observation 5. Consider $\tilde{y}_s \in \mathcal{Y}$ such that

$$\min_{\xi \in \Xi} \sum_{i \in [n]} (c_i + \xi_i) \tilde{y}_{s,i} > z^{\text{MM}}, \quad (26)$$

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

Proof. See Appendix B.2. □

Despite Observations 4 and 5, 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 $T^\ell \subseteq [t], \ell \in [n]_0$, on the fly by exploiting dual information (see for instance Wolsey [2020]). Let us describe this idea more precisely in what follows, denoting by $\text{DLR}(T)$ the dual of the linear relaxation of (DDID-CG) associated to subsets T^0, \dots, T^n , while DLR denotes the dual of the full linear relaxation of (DDID-CG). Let v denote the vector of all dual variables and consider an optimal dual solution v^* of $\text{DLR}(T)$. Notice that DLR has the same variables as $\text{DLR}(T)$ 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 $\text{DLR}(T)$ are those associated with the primal variables $\lambda_{\ell,s}$ for each $\ell \in [n]_0$ and $s \in [t] \setminus T^\ell$, namely

$$\sum_{i \in [n]} c_i \tilde{y}_{s,i} - \sum_{i \in [n]} \rho_i^{\ell*} \tilde{y}_{s,i} - \nu^{\ell*} \geq 0. \quad (27)$$

Therefore, all constraints (27) are satisfied by v^* if and only if for each $\ell \in [n]_0$, the optimal solution of the following optimization problem is not smaller than $\nu^{\ell*}$

$$\min_{y \in \mathcal{Y}} \left\{ \sum_{i \in [n]} c_i y_i - \sum_{i \in [n]} \rho_i^{\ell*} y_i \right\}. \quad (28)$$

If, on the contrary, there exists $\ell \in [n]_0$ for which we are able to identify $\tilde{y}_{s'} \in \mathcal{Y}$ such that the corresponding dual constraint (27) is violated by v^* , we add the corresponding index s' to T^ℓ and solve the resulting linear program $\text{DLR}(T)$ 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. While this particular type of branch-and-price algorithm is of a rather simple type, as it involves no branching on the set of dynamic variables, λ , its efficient implementation involves clever engineering techniques, such as node selection, heuristic, and stabilization, which is beyond the scope of the current work. Instead, we limit ourselves to the column generation at the root node only, and then feed the resulting MILP to a solver. Doing so, we end up with a heuristic algorithm for (DDID), because the returned observation w^* will have considered only a subset of $\text{conv}(\mathcal{Y})$. Another direct consequence of the above discussion is that the above column generation algorithm provides yet another way to compute $\Phi(w)$.

Observation 6. $\Phi(w)$ 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 3. 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

$$z_{\mathcal{P}'}^{\text{DDID}} = \min_{\theta \in \Theta(\mathcal{P}')} f(\theta), \quad (29)$$

Observe that when $\mathcal{P}' = \text{conv}(\mathcal{Y})$, the condition of Proposition 5 is satisfied and (29) coincides with the exact reformulation (DDID-SIP), so $z_{\mathcal{P}'}^{\text{DDID}} = z^{\text{DDID}}$ in this case.

Observation 7. *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. \square

The algorithm starts with $\mathcal{P}^0 = \mathcal{P}$ and solves (29), 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 $\ell \in [n]_0$, either $u_\ell = 0$ and $y_\ell^* = 0$ or $y_\ell^* \in \text{conv}(\mathcal{Y})$. Hence, we can use Assumption 3 to check whether $\theta^* \in \Theta(\text{conv}(\mathcal{Y}))$. If this is the case, Observation 7 implies that θ^* is optimal for the exact formulation $\min_{\theta \in \Theta(\text{conv}(\mathcal{Y}))} f(\theta)$ so $z_{\mathcal{P}^0}^{\text{DDID}} = z^{\text{DDID}}$. Otherwise, we rely on the oracle from Assumption 3 to obtain a separating hyperplane $h^T y \leq h^0$, define

$$\mathcal{P}^1 = \mathcal{P}^0 \cap \{y \in \mathbb{R}^n \mid h^T y \leq h^0\},$$

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

Proof. For any $w \in \mathcal{W}$, we have that $\Phi(w) \geq z^{\text{DDID}} \geq z_{\mathcal{P}^i}^{\text{DDID}}$, where the second inequality follows from Observation 7. Combining the above with $\Phi(w^*) \leq z_{\mathcal{P}^i}^{\text{DDID}}$ proves the result. \square

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 [Bezanson et al., 2017], using JuMP [Dunning et al., 2017] to interface the mixed integer linear programming (MILP) solver CPLEX 20.01. We ran our experiments on a processor Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz, letting CPLEX handle the parallelism and reporting the total CPU times. 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 Selection problem

We first experiment the reformulation from Proposition 3 with the selection problem, where the decision maker wishes to choose p out of n items, so $\mathcal{Y}^{\text{sel}} = \{y \in \{0, 1\}^n \mid \sum_{i \in [n]} y_i = p\}$. The selection problem has been used in numerous papers addressing complex robust variants [Goerigk and Lendl, 2021, Goerigk et al., 2022a,b], including DDID itself under the name of two-stage robust best box selection [Vayanos et al., 2022], in which $p = 1$.

We use the budget uncertainty set of Bertsimas and Sim [2003, 2004], defined as $\Xi^\Gamma = \{\xi \in \mathbb{R}^n \mid \sum_{i \in [n]} \xi_i \leq \Gamma, 0 \leq \xi_i \leq 1\}$, largely used in the scientific literature on robust combinatorial optimization. We further consider the selection set for information discovery, $\mathcal{W}^{\text{sel}} = \{w \in \{0, 1\}^n \mid \sum_{i \in [n]} w_i \leq q\}$, 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.

n	uncertainty				p				q				Γ			
	$n/2$		n		$n/10$		$n/5$		$n/10$		$n/5$		$n/10$		$n/5$	
10	8.3	11	7.2	12	6.6	11	8.9	13	8.9	10	6.6	13	9.0	19	6.5	4.4
20	8.4	10	6.5	13	7.0	11	7.8	12	9.7	8.5	5.2	15	7.6	19	7.2	4.4
30	11	6.5	7.7	10	8.4	7.7	9.9	9.2	11	6.3	7.6	11	11	14	7.6	2.9
40	8.2	6.2	6.7	8.1	5.4	5.4	9.6	8.9	8.6	5.7	6.4	8.6	9.5	11	5.4	3.0
50	8.6	6.8	6.6	9.3	6.1	6.1	9.1	10	9.1	6.1	6.1	10	9.7	13	5.5	2.7

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

n	uncertainty				p				q				Γ			
	$n/2$		n		$n/10$		$n/5$		$n/10$		$n/5$		$n/10$		$n/5$	
10	0.05	28	0.05	36	0.05	39	0.05	26	0.05	33	0.05	32	0.05	28	0.05	37
20	0.12	45	0.13	61	0.1	60	0.15	45	0.12	54	0.13	52	0.12	50	0.13	55
30	0.26	37	0.69	56	0.39	54	0.57	39	0.32	47	0.64	46	0.35	43	0.6	50
40	0.66	39	2.8	48	0.92	48	2.6	39	1.3	44	2.2	43	0.75	40	2.7	47
50	4.3	47	62	58	13	59	53	47	14	53	53	52	6.8	50	59	55

(a) Weak formulation (DDID-WIP)

n	uncertainty				p				q				Γ			
	$n/2$		n		$n/10$		$n/5$		$n/10$		$n/5$		$n/10$		$n/5$	
10	0.05	0.01	0.06	0.14	0.05	0.0	0.06	0.15	0.05	0.0	0.06	0.15	0.05	0.09	0.06	0.06
20	0.06	0.26	0.1	0.26	0.08	0.29	0.09	0.24	0.08	0.39	0.08	0.14	0.09	0.48	0.08	0.05
30	0.16	0.08	0.26	0.39	0.18	0.16	0.24	0.3	0.18	0.16	0.24	0.3	0.22	0.35	0.2	0.11
40	0.31	0.06	0.49	0.16	0.27	0.11	0.53	0.11	0.37	0.14	0.44	0.09	0.41	0.19	0.39	0.04
50	0.75	0.14	1.6	0.22	0.79	0.14	1.5	0.22	1.0	0.21	1.3	0.15	1.1	0.25	1.2	0.11

(b) Strong formulation (DDID-SIP)

Table 2: Average solution times in seconds and root gaps in %.

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 often below 10%, and hardly higher than 20%. 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 leads to smaller gaps, often significantly.

Table 2 reports the solution times in seconds 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 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.

n	K	uncertainty		p		q		Γ	
		$\lceil n/2 \rceil$	n	$\lceil n/10 \rceil$	$n/5$	$\lceil n/10 \rceil$	$n/5$	$\lceil n/10 \rceil$	$n/5$
10	2	0.06	0.08	0.06	0.17	0.07	0.32	0.07	0.32
	3	0.21	0.27	0.12	1.8	0.26	17	0.19	14
15	2	0.57	0.62	0.17	0.91	0.32	0.6	0.32	0.61
	3	27	33	1.8	56	17	25	14	31

Table 3: Average solution times in seconds for the K -adaptability reformulation presented in Section D.

We compared the above results with our own implementation of the K -adaptability reformulation proposed Vayanos et al. [2022], see Section D 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 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 results presented in [Vayanos et al., 2022, 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. [2022] model uncertainty by projecting a 4-dimensional box into \mathbb{R}^n , specifically,

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

for given affine mappings ψ_i , and $L = 4$ risk factors. As stated in Observation 3, (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 Vayanos et al. [2022]. 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 [Vayanos et al., 2020, Figure 3]. Another source of simplification in Vayanos et al. [2022] 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.

6.2 Orienteering problem

Our second set of experiments focuses on a particular routing problem considered by Paradiso et al. [2022]: the orienteering problem. That problem is a special case of the maximization counterpart detailed in Section C, which we specify next by defining \mathcal{Y}^{max} and Ξ^{max} (the maximization coun-

terparts of \mathcal{Y} and Ξ). 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}^{max} is an elementary path from 0 to $n + 1$ having a total weight that does not exceed T . Introducing binary variable x_e to model the use of edge e in the path, and denoting the star of node i as $E(i) = \{e \in E \mid i \in e\}$, we formulate \mathcal{Y} as

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

where “subtour elimination constraints” denotes any set of constraints preventing cycles in y (e.g. Taccari [2016]). Polytope \mathcal{P}^{max} is obtained from \mathcal{Y}^{max} by removing the integrality restrictions on y and z and projecting the resulting polytope on variables y . Furthermore, we follow Paradiso et al. [2022], and define $c = 0$, $\mathcal{W} = \{w \in \{0, 1\}^n \mid \sum_{i \in [n]} w_i \leq q\}$, and

$$\Xi^{max} = \left\{ \xi \in \mathbb{R}^n \mid \sum_{i \in [n]} \xi_i \geq 1, 0 \leq \xi_i \leq U, \forall i \in [n] \right\},$$

for some given $U > 0$ and $q = \lceil \delta n \rceil$ for some given $\delta \in (0, 1)$. Clearly, $\text{conv}(\mathcal{Y}^{max}) \subset \mathcal{P}^{max}$, so we cannot apply the results from Section C to this problem.

Therefore, we consider instead the convex hull formulation described in Section 5.1 and test the two approaches described in that section. First, we consider the exact algorithm, based on the full enumeration of the elements in \mathcal{Y} . We use a maximization counterpart of Observation 4, thus enumerating only the maximal paths in \mathcal{Y} . However, Observation 5 could not be leveraged. Indeed, the maximization counterpart of (26) becomes $\max_{\xi \in \Xi^{max}} \sum_{i \in [n]} (c_i + \xi_i) \tilde{y}_{s,i} < \max_{y \in \mathcal{Y}^{max}} \min_{\xi \in \Xi^{max}} \sum_{i \in [n]} (c_i + \xi_i) \tilde{y}_{s,i}$. With the above definitions of c and Ξ^{max} , the condition becomes $\min(1, U|\tilde{y}_s|) < \max(0, 1 - U(n - \max_{y \in \mathcal{Y}^{max}} |y|))$, which is never satisfied for the values of n and U provided in Table 6, even when $|\tilde{y}_s| = 1$ and $\max_{y \in \mathcal{Y}^{max}} |y| = 1$. Second, we consider the heuristic based on the column-generation at the root node. The “subtour elimination constraints” used in our experiments are the classical Dantzig-Fulkerson-Johnson subtour elimination

constraints, separated exactly at the integer node of the branch-and-cut-tree, together with the generalized cutset inequalities, separated heuristically at fractional nodes using Tarjan’s algorithm as detailed in Taccari [2016]. We test the algorithms on a subset of the instances from Paradiso et al. [2022], consisting of complete graphs with 10 to 31 nodes (excluding the depots), and further described in the appendix.

instance	δ	Opt (#)		time (s) (solved)			time (s) (all)	gap (%) w.r.to	
		CB	conv	CB	conv	CG	CG	CB	max(CB, conv)
TS2N10	0.25	<i>9/9</i>	9/9	<i>5.4</i>	0.1	0.41	0.41	0.0	0.0
	0.5	<i>9/9</i>	9/9	<i>3.5</i>	0.1	0.64	0.64	0.0	0.0
	0.75	<i>9/9</i>	9/9	<i>2.3</i>	0.11	0.69	0.69	0.0	0.0
TS1N15	0.25	<i>14/14</i>	14/14	<i>122</i>	72	0.74	0.74	0.0	0.0
	0.5	<i>14/14</i>	14/14	<i>69</i>	71	1.6	1.6	0.2	0.2
	0.75	<i>14/14</i>	14/14	<i>13</i>	70	3.1	3.1	0.0	0.0
TS3N16	0.25	<i>14/14</i>	13/14	<i>66</i>	463	2.1	2.0	0.0	0.0
	0.5	<i>14/14</i>	13/14	<i>94</i>	464	5.9	5.5	0.2	0.2
	0.75	<i>14/14</i>	13/14	<i>18</i>	464	6.5	6.1	0.0	0.0
TS2N19	0.25	<i>6/11</i>	7/11	<i>2534</i>	101	1.8	2.1	0.5	0.5
	0.5	<i>8/11</i>	7/11	<i>3122</i>	612	29	24	0.2	0.2
	0.75	<i>11/11</i>	7/11	<i>602</i>	718	33	30	0.0	0.0
TS1N30	0.25	<i>6/18</i>	6/18	<i>0.36</i>	0.02	2.1	16	-4.1	-0.3
	0.5	<i>6/18</i>	6/18	<i>0.33</i>	0.02	2.1	81	-2.7	-1.1
	0.75	<i>10/18</i>	6/18	<i>233</i>	707	63	260	-0.3	-0.3
TS3N31	0.25	<i>6/20</i>	4/20	–	–	–	42	-5.4	-0.8
	0.5	<i>6/20</i>	4/20	–	–	–	225	-3.3	-0.6
	0.75	<i>8/20</i>	4/20	<i>2994</i>	4.0	65	409	-0.6	-0.3

Table 4: Numerical results on the orienteering problem. Italicized results have been provided by Paradiso [2023] and have been run on a configuration different than ours. Entries “–” indicate no subset of instances solved by all methods.

The results are presented in Table 4. Columns CB, conv and CG respectively denote the combinatorial Benders algorithm from Paradiso et al. [2022], the exact convexification and the column-generation algorithm from Section 5.1. The columns “Time” report the average solution times over the subset of instances solved by all methods, which may be empty, leading to the entry “–”. The additional column “Time (all)” reports the average solution times on all instances. The columns “gap” report the cost difference between $\Phi(w)$ for the solution w returned by CG and (i) the best solution returned by CB, and (ii) the best known solution. Notice that the results reported for CB have been carried out using a different configuration (processor, CPLEX, parallelism), so the comparison between the respective columns should be made with care.

Overall, our results indicate that `conv`, despite its exhaustive enumeration, is somewhat competitive, solving nearly as many instances as `CB`, some of them faster (e.g. TS2N19). More importantly, its heuristic variant `CG` scales well with the dimensions of the instances, solving them in a few minutes for the largest. The results also indicate that the best solution returned by `CG` is almost always optimal, often beating the best solutions returned by `CB` for the most difficult instances (corresponding to the negative gaps). We report additional statistics on algorithms `conv` and `CG` in Appendix E, including the excellent root gap of the MILP reformulation.

6.3 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 Magnanti and Wolsey [1995], denoted `compact`. This formulation is compact and known to be exact, so Proposition 5 applies. The second approach, `CG`, relies on the column generation heuristic, already illustrated for the orienteering problem. 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 Magnanti and Wolsey [1995]. For a given solution θ^* of the current relaxation, we separate constraints of the subtour formulation by following the algorithm described by Magnanti and Wolsey [1995]. Given that the multi-commodity flow formulation is exact, for each $u_\ell^* > 0$ the maximum flow from one arbitrary root to any other vertex must be equal to u_ℓ^* if $y_\ell^* \in \text{conv}(\mathcal{Y})$. 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 $\ell \in [n]_0$ in `compact`).

The three methods are compared on a benchmark similar to that used by Focke et al. [2020]. 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.

name	nodes	edges	Γ	q	time (s)			cols	cuts	gap (%)	
					compact	CG	CP	CG	CP	root	CG
burma14.tsp	14	51	3	3	19	2.8	4.0	42	104	0.11	1.7
ulysses22.tsp	22	85	4	4	148	16	25	114	430	0.0	6.6
bays29.tsp	29	105	6	6	–	34	200	53	636	0.15	0.49
swiss42.tsp	42	159	8	8	–	118	102	249	0	0.2	2.4
eil51.tsp	51	186	10	10	–	906	3478	572	561	0.21	0.95

Table 5: Results for the MST presenting solution times, numbers of cuts and columns generated, root gaps and optimality gaps for CG.

Table 5 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 could not solve the three largest instances because of memory issues. In addition to the good root gaps and the good optimality gaps of CG, the table illustrates the solving capability of CP, which is able to solve exactly larger instances than compact. The good performance of CP is partly due to the strong initial relaxation since few subtour inequalities are generated, sometimes even 0.

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.

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 knapsack uncertainty polytope. 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 and

row generation algorithms. Our experiments have again illustrated the interest of these algorithms. On the one hand, the heuristic based on column generation has provided good results on the orienteering instances considered by Paradiso et al. [2022], providing better primal bounds than previously known for some of them. On the other hand, the cutting plane algorithm 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 than the dimension of the uncertainty set.

This work leads to several interesting open questions for future works. On the numerical side, the excellent dual bound provided by the column generation algorithm calls for extending the latter into an exact branch-and-price algorithm, 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 knapsack uncertainty sets, 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 4 with BitArrays. For the purpose of Open Access, a CC-BY public copyright license has been applied by the authors to the present document and will be applied to all subsequent versions up to the Author Accepted Manuscript arising from this submission.



References

- A. Agra, M. C. Santos, D. Nace, and M. Poss. A dynamic programming approach for a class of robust optimization problems. *SIAM J. Optim.*, 26(3):1799–1823, 2016. doi: 10.1137/15M1007070. URL <https://doi.org/10.1137/15M1007070>.
- A. N. Arslan and B. Detienne. Decomposition-based approaches for a class of two-stage robust binary optimization problems. *INFORMS J. Comput.*, 34(2):857–871, 2022. doi: 10.1287/ijoc.2021.1061. URL <https://doi.org/10.1287/ijoc.2021.1061>.
- A. N. Arslan, M. Poss, and M. Silva. Min-sup-min robust combinatorial optimization with few recourse solutions. *INFORMS J. Comput.*, 34(4):2212–2228, 2022. doi: 10.1287/ijoc.2021.1156. URL <https://doi.org/10.1287/ijoc.2021.1156>.
- J. Ayoub and M. Poss. Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Comput. Manag. Science*, 13(2):219–239, 2016. doi: 10.1007/s10287-016-0249-2. URL <https://doi.org/10.1007/s10287-016-0249-2>.
- A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.
- A. Ben-Tal, A. P. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Math. Program.*, 99(2):351–376, 2004. doi: 10.1007/s10107-003-0454-y. URL <https://doi.org/10.1007/s10107-003-0454-y>.
- D. Bertsimas and I. Dunning. Multistage robust mixed-integer optimization with adaptive partitions. *Oper. Res.*, 64(4):980–998, 2016. doi: 10.1287/opre.2016.1515. URL <https://doi.org/10.1287/opre.2016.1515>.
- D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Math. Program.*, 98(1-3):49–71, 2003. doi: 10.1007/s10107-003-0396-4. URL <https://doi.org/10.1007/s10107-003-0396-4>.
- D. Bertsimas and M. Sim. The price of robustness. *Oper. Res.*, 52(1):35–53, 2004. doi: 10.1287/opre.1030.0065. URL <https://doi.org/10.1287/opre.1030.0065>.
- D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Operations research*, 54(1):150–168, 2006.
- J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.
- M. Colvin and C. T. Maravelias. A stochastic programming approach for clinical trial planning in new drug development. *Computers & chemical engineering*, 32(11):2626–2642, 2008.
- I. Dunning, J. Huchette, and M. Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017. doi: 10.1137/15M1020575.
- J. Focke, N. Megow, and J. Meißner. Minimum spanning tree under explorable uncertainty in theory and experiments. *ACM J. Exp. Algorithmics*, 25:1–20, 2020. doi: 10.1145/3422371. URL <https://doi.org/10.1145/3422371>.
- M. Goerigk and S. Lendl. Robust combinatorial optimization with locally budgeted uncertainty. *Open J. Math. Optim.*, 2:1–18, 2021. doi: 10.5802/ojmo.5. URL <https://doi.org/10.5802/ojmo.5>.

- M. Goerigk, A. Kasperski, and P. Zielinski. Robust two-stage combinatorial optimization problems under convex second-stage cost uncertainty. *J. Comb. Optim.*, 43(3):497–527, 2022a. doi: 10.1007/s10878-021-00776-4. URL <https://doi.org/10.1007/s10878-021-00776-4>.
- M. Goerigk, S. Lendl, and L. Wulf. Two-stage robust optimization problems with two-stage uncertainty. *Eur. J. Oper. Res.*, 302(1):62–78, 2022b. doi: 10.1016/j.ejor.2021.12.046. URL <https://doi.org/10.1016/j.ejor.2021.12.046>.
- K.-S. Goetzmann, S. Stiller, and C. Telha. Optimization over integers with robustness in cost and few constraints. In *International Workshop on Approximation and Online Algorithms*, pages 89–101. Springer, 2012.
- C. E. Gounaris, W. Wiesemann, and C. A. Floudas. The robust capacitated vehicle routing problem under demand uncertainty. *Oper. Res.*, 61(3):677–693, 2013. doi: 10.1287/opre.1120.1136. URL <https://doi.org/10.1287/opre.1120.1136>.
- G. A. Hanasusanto, D. Kuhn, and W. Wiesemann. K-adaptability in two-stage robust binary programming. *Operations Research*, 63(4):877–891, 2015.
- T. W. Jonsbråten. Optimization models for petroleum field exploitation. 1998.
- T. W. Jonsbråten, R. J. Wets, and D. L. Woodruff. A class of stochastic programs with decision dependent random elements. *Annals of Operations Research*, 82:83–106, 1998.
- N. Kämmerling and J. Kurtz. Oracle-based algorithms for binary two-stage robust optimization. *Comput. Optim. Appl.*, 77(2):539–569, 2020. doi: 10.1007/s10589-020-00207-w. URL <https://doi.org/10.1007/s10589-020-00207-w>.
- P. Kouvelis and G. Yu. Robust discrete optimization: past successes and future challenges. In *Robust discrete optimization and its applications*, pages 333–356. Springer, 1997.
- T. Lee and C. Kwon. A short note on the robust combinatorial optimization problems with cardinality constrained uncertainty. *4OR*, 12(4):373–378, 2014. doi: 10.1007/s10288-014-0270-7. URL <https://doi.org/10.1007/s10288-014-0270-7>.
- T. L. Magnanti and L. A. Wolsey. Optimal trees. *Handbooks in operations research and management science*, 7: 503–615, 1995.
- O. Nohadani and K. Sharma. Optimization under decision-dependent uncertainty. *SIAM Journal on Optimization*, 28(2):1773–1795, 2018.
- R. Paradiso. personal communication, 2023.
- R. Paradiso, A. Georghiou, S. Dabia, and D. Tönissen. Exact and approximate schemes for robust optimization problems with decision dependent information discovery. *arXiv preprint arXiv:2208.04115*, 2022.
- A. A. Pessoa, M. Poss, R. Sadykov, and F. Vanderbeck. Branch-cut-and-price for the robust capacitated vehicle routing problem with knapsack uncertainty. *Oper. Res.*, 69(3):739–754, 2021. doi: 10.1287/opre.2020.2035. URL <https://doi.org/10.1287/opre.2020.2035>.
- M. Poss. Robust combinatorial optimization with variable budgeted uncertainty. *4OR*, 11(1):75–92, 2013. doi: 10.1007/s10288-012-0217-9. URL <https://doi.org/10.1007/s10288-012-0217-9>.
- M. Poss. Robust combinatorial optimization with knapsack uncertainty. *Discret. Optim.*, 27:88–102, 2018. doi: 10.1016/j.disopt.2017.09.004. URL <https://doi.org/10.1016/j.disopt.2017.09.004>.

- K. Postek and D. den Hertog. Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS J. Comput.*, 28(3):553–574, 2016. doi: 10.1287/ijoc.2016.0696. URL <https://doi.org/10.1287/ijoc.2016.0696>.
- Pulleyblank, William R. *Faces Of Matching Polyhedra*. PhD thesis, 1973. URL <http://hdl.handle.net/10012/10971>.
- M. Queyranne and A. S. Schulz. *Polyhedral approaches to machine scheduling*. Citeseer, 1994.
- F. Rodrigues, A. Agra, C. Requejo, and E. Delage. Lagrangian duality for robust problems with decomposable functions: The case of a robust inventory problem. *INFORMS J. Comput.*, 33(2):685–705, 2021. doi: 10.1287/ijoc.2020.0978. URL <https://doi.org/10.1287/ijoc.2020.0978>.
- A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.
- S. Solak, J.-P. B. Clarke, E. L. Johnson, and E. R. Barnes. Optimization of r&d project portfolios under endogenous uncertainty. *European Journal of Operational Research*, 207(1):420–433, 2010.
- S. A. Spacey, W. Wiesemann, D. Kuhn, and W. Luk. Robust software partitioning with multiple instantiation. *INFORMS Journal on Computing*, 24(3):500–515, 2012.
- A. Subramanyam, C. E. Gounaris, and W. Wiesemann. K-adaptability in two-stage mixed-integer robust optimization. *Mathematical Programming Computation*, 12(2):193–224, 2020.
- L. Taccari. Integer programming formulations for the elementary shortest path problem. *European Journal of Operational Research*, 252(1):122–130, 2016.
- B. Tadayon and J. C. Smith. Algorithms and complexity analysis for robust single-machine scheduling problems. *J. Sched.*, 18(6):575–592, 2015. doi: 10.1007/s10951-015-0418-0. URL <https://doi.org/10.1007/s10951-015-0418-0>.
- P. Vayanos, A. Georghiou, and H. Yu. Robust optimization with decision-dependent information discovery. *arXiv preprint arXiv:2004.08490v1*, 2020.
- P. Vayanos, A. Georghiou, and H. Yu. Robust optimization with decision-dependent information discovery. *arXiv preprint arXiv:2004.08490v3*, 2022.
- L. A. Wolsey. *Integer programming*. John Wiley & Sons, 2020.
- I. Yanikoglu, B. L. Gorissen, and D. den Hertog. A survey of adjustable robust optimization. *Eur. J. Oper. Res.*, 277(3):799–813, 2019. doi: 10.1016/j.ejor.2018.08.031. URL <https://doi.org/10.1016/j.ejor.2018.08.031>.
- B. Zeng and L. Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461, 2013.
- J. Zhen, D. Den Hertog, and M. Sim. Adjustable robust optimization via fourier–motzkin elimination. *Operations Research*, 66(4):1086–1100, 2018.

A. Proof of Section 4

A.1 Proof of Proposition 3

Consider the linear program introduced in Theorem 1, and let us introduce dual variables u_ℓ and $y_{\ell,i}$ 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)

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

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

$$\min \quad \sum_{\ell \in [n]_0} \left(r\alpha_\ell u_\ell + \sum_{i \in [n]} c_i y_{\ell,i} + \sum_{i \in [n]} \beta_{\ell,i} y_{\ell,i} (1 - w_i) \right) + \sum_{i \in [n]} d_i \sigma_i + r\mu \quad (30)$$

$$\text{s.t.} \quad \sum_{\ell \in [n]_0} u_\ell = 1 \quad (31)$$

$$a_i \mu + \sigma_i \geq w_i \left(-a_i \sum_{\ell \in [n]_0} \alpha_\ell u_\ell + \sum_{\ell \in [n]_0} y_{\ell,i} \right), \forall i \in [n] \quad (32)$$

$$B y_\ell \geq u_\ell b, \forall \ell \in [n]_0 \quad (33)$$

$$w \in \mathcal{W} \quad (34)$$

$$u, y, \mu, \sigma \geq 0. \quad (35)$$

Next, we linearize the product by w_i in (32) with a big-M term which yields:

$$a_i \mu + \sigma_i \geq -a_i \sum_{\ell \in [n]_0} \alpha_\ell u_\ell + \sum_{\ell \in [n]_0} y_{\ell,i} - M(1 - w_i), \forall i \in [n].$$

Given that $y_{\ell,i} \leq u_\ell, \forall i \in [n]$, we have $\sum_{\ell \in [n]_0} y_{\ell,i} \leq \sum_{\ell \in [n]_0} u_\ell = 1, \forall i \in [n]$. As a consequence, M can be set to 1 in (32). We conclude by introducing variables $y_{\ell,i}^0$ to represent the products $y_{\ell,i}(1 - w_i)$, and adding the linearization constraints $y_{\ell,i}^0 \geq y_{\ell,i} - w_i$.

A.2 Proof of Proposition 4

Referring to Theorem 1 and to the proof of Proposition 3, we will consider the pair of primal-dual adversary formulations given by

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

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

Let $(\bar{\xi}^*, \lambda^*, \pi^*)$ and $(u^*, y^*, \mu^*, \sigma^*)$ be a pair of optimal solutions to $\mathcal{P}(w)$ and $\mathcal{D}(w)$. Furthermore, recall the notations $\alpha_\ell = 1/a_\ell$ for each $\ell \in [n]$, and for each $i, \ell \in [n]$, $\beta_{\ell,i} = d_i [1 - a_i/a_\ell]^+$, while $\alpha_0 = 0$ and $\beta_{i,0} = d_i, \forall i \in [n]$.

Assume now that $\mu^* > 0$. By complementarity, this means that $\sum_{i \in \mathcal{W}_*^1} a_i \bar{\xi}_i^* = r$. As a consequence, $\omega^* \leq b^T \lambda_\ell^* - \sum_{i \in [n]} \pi_{\ell,i}^*, \forall \ell \in [n]_0$. Now, let $\tilde{\ell} \in [n]$ such that $u_{\tilde{\ell}}^* > 0$ and $i \in \mathcal{W}^0$ such that $y_{\tilde{\ell},i}^* > 0$. By complementarity, we have $(B_{\cdot,i})^T \lambda_{\tilde{\ell}}^* - \pi_{\tilde{\ell},i}^* = c_i + \beta_{\tilde{\ell},i}$. If

$$\beta_{\tilde{\ell},i} > 0, \quad (36)$$

we build a new solution, $(\omega^*, \bar{\xi}^*, \lambda', \pi')$, of $\mathcal{P}(w^*)$ by slightly modifying λ^* and π^* . We set $\lambda'_{\tilde{\ell}} := \lambda_{\tilde{\ell}}^*$ and $\pi'_{\tilde{\ell}} := \pi_{\tilde{\ell}}^*$ while keeping the other components of π^* and λ^* unchanged. Observing that (36) is equivalent to $a_{\tilde{\ell}} > a_i$, one can verify that $(\omega^*, \bar{\xi}^*, \lambda', \pi')$ is feasible for $\mathcal{P}(w^*)$ and $(B_{\cdot,i})^T \lambda'_{\tilde{\ell}} - \pi'_{\tilde{\ell},i} < c_i + \beta_{\tilde{\ell},i}$. However, by complementarity, we also get $(B_{\cdot,i})^T \lambda'_{\tilde{\ell}} - \pi'_{\tilde{\ell},i} = c_i + \beta_{\tilde{\ell},i}$, a contradiction. Therefore, $\beta_{\tilde{\ell},i} = 0$ (and thus $\left[1 - \frac{a_i}{a_{\tilde{\ell}}}\right]^+ = 0$) for all $i \in \mathcal{W}^0$ and $\ell \in [n]_0$ such that $y_{\ell,i}^* > 0$. Transposing the above reasoning to $\tilde{\ell} = 0$, 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 $\mathcal{D}(w^*)$ is given by

$$r \left(\mu^* + \sum_{\ell \in [n]} \alpha_\ell u_\ell^* \right) + \sum_{i \in [n]} \left(d_i \sigma_i^* + c_i \sum_{\ell \in [n]} y_{\ell,i}^* \right).$$

The dual constraint of $\bar{\xi}_i, i \in \mathcal{W}^1$ may then 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 $\mathcal{D}(w^*)$, (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

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

initialization: $y' := y^*, u' := u^*, \mu' := \mu^*, K = \{k \in [n] \mid u_k^* > 0, \alpha_k < \alpha_j\}$.

- 1 **for** $k \in K$ **do**
- 2 $\delta := \min \left\{ u_k^*, \frac{\mu'}{\alpha_j - \alpha_k} \right\}$
- 3 $u'_k \leftarrow u'_k - \delta$
- 4 $u'_j \leftarrow u'_j + \delta$
- 5 $y'_k \leftarrow y'_k - \frac{\delta}{u_k^*} y_k^*$
- 6 $y'_j \leftarrow y'_j + \frac{\delta}{u_k^*} y_k^*$
- 7 $\mu' \leftarrow \mu' - \delta (\alpha_j - \alpha_k)$
- 8 **if** $\mu' = 0$ **then**
- 9 **break**
- 10 **return** (y', u', μ')

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

A.3 Proof of Proposition 5

We show that the model is a valid linearization of the intermediary model (30)–(35) of the proof of Proposition 3, having removed variable μ in accordance with Proposition 4. For this, we introduce $u_{i,\ell}^1 := w_i u_\ell, u_{i,\ell}^0 := (1 - w_i) u_\ell, y_{i,\ell}^1 := w_i y_{\ell,i}, y_{i,\ell}^0 := (1 - w_i) y_{\ell,i}$. 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 $u_\ell = u_{\ell,i}^0 + u_{\ell,i}^1, u_{\ell,i}^0 \leq 1 - w_i$ and $u_{\ell,i}^1 \leq w_i$, for all $\ell \in [n]_0$ and $i \in [n]$. Similar constraints could be added to

linearize y^0 and y^1 , but we instead leverage constraints $y_{\ell,i} \leq u_{\ell}, \ell \in [n]_0, i \in [n]$, to add the tighter constraints $y_{\ell,i} = y_{\ell,i}^0 + y_{\ell,i}^1, y_{\ell,i}^0 \leq u_{\ell,i}^0$ and $y_{\ell,i}^1 \leq u_{\ell,i}^1$ for all $\ell \in [n]_0$ and $i \in [n]$. The objective function (30) and constraints (32) are then naturally linearized using the definitions of y^0, u^1 and y^1 . Finally, constraints $Gu_{\ell}^1 \leq u_{\ell}g$ are not necessary, but they are valid inequalities obtained by multiplying $Gw \leq g$ by u_{ℓ} for each $\ell \in [n]_0$.

A.4 Proof of Proposition 6

We consider a feasible solution to the linear relaxation of (DDID-SIP) given by vectors $\bar{w}, \bar{u}, \bar{u}^0, \bar{u}^1, \bar{y}, \bar{y}^0, \bar{y}^1, \bar{\sigma}$ and we consider its projection on the variables of (DDID-WIP), $\bar{w}, \bar{u}, \bar{y}, \bar{y}^0, \bar{\sigma}, \bar{\mu}$ where $\bar{\mu} = 0$. The satisfaction of most constraints is immediate, but some verifications need to be carried out for constraints $y_{\ell,i}^0 \geq y_{\ell,i} - w_i$ and $\sigma_i \geq -a_i \sum_{\ell \in [n]_0} \alpha_{\ell} u_{\ell} + \sum_{\ell \in [n]_0} y_{\ell,i} - (1 - w_i)$. For the former, we use that $\bar{y} = \bar{y}^0 + \bar{y}^1$ and $\bar{y}^1 \leq \bar{u}^1$ to show that $\bar{y}_{\ell,i}^0 \geq \bar{y}_{\ell,i} - \bar{u}_{\ell,i}^1 \geq \bar{y}_{\ell,i} - \bar{w}_i$ for all $\ell \in [n]_0, i \in [n]$. To show that the latter constraints are satisfied, we infer the following sequence of inequalities from the linear constraints of (DDID-SIP).

$$\begin{aligned}
& -a_i \sum_{\ell \in [n]_0} \alpha_{\ell} \bar{u}_{\ell} + \sum_{\ell \in [n]_0} \bar{y}_{\ell,i} - (1 - \bar{w}_i) \\
&= -a_i \sum_{\ell \in [n]_0} \alpha_{\ell} \bar{u}_{\ell,i}^1 + \sum_{\ell \in [n]_0} \bar{y}_{\ell,i}^1 - a_i \sum_{\ell \in [n]_0} \alpha_{\ell} \bar{u}_{\ell,i}^0 + \sum_{\ell \in [n]_0} \bar{y}_{\ell,i}^0 - (1 - \bar{w}_i) \\
&\leq -a_i \sum_{\ell \in [n]_0} \alpha_{\ell} \bar{u}_{\ell,i}^1 + \sum_{\ell \in [n]_0} \bar{y}_{\ell,i}^1 - a_i \sum_{\ell \in [n]_0} \alpha_{\ell} \bar{u}_{\ell,i}^0 + \sum_{\ell \in [n]_0} \bar{u}_{\ell,i}^0 - \sum_{\ell \in [n]_0} \bar{u}_{\ell,i}^0 \\
&\leq -a_i \sum_{\ell \in [n]_0} \alpha_{\ell} \bar{u}_{\ell,i}^1 + \sum_{\ell \in [n]_0} \bar{y}_{\ell,i}^1 \\
&\leq \sigma_i.
\end{aligned}$$

B. Proofs of Section 5.1

B.1 Proof of Observation 4

To prove inequality $z^{\text{DDID}} \geq \tilde{z}^{\text{DDID}}$, let $(u^*, u^{0*}, u^{1*}, y^{0*}, y^{1*}, \sigma^*, \lambda^*)$ be an optimal solution to (DDID-CG) associated to \mathcal{Y} . We construct a solution $(u^*, u^{0*}, u^{1*}, y^{0*}, y^1, \sigma^*, \lambda')$ to the formulation associated to $\tilde{\mathcal{Y}}$ 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 $y_{\ell,i}^1 = \sum_{s \in [t]} \lambda'_{\ell,s} \tilde{y}_{s,i} - y_{\ell,i}^{*0}$, $\ell \in [n]_0, i \in [n]$. Observe that $\tilde{y}_{s_1} \leq \tilde{y}_{s_2}$ implies that $\sum_{s \in [t]} \lambda'_{\ell,s} \tilde{y}_{s,i} \leq \sum_{s \in [t]} \lambda'_{\ell,s} \tilde{y}_{s,i}$ and $y_{\ell,i}^1 \leq y_{\ell,i}^{*1}$ for all $\ell \in [n]_0, i \in [n]$. As a consequence, one readily verifies that $(u^*, u^{0*}, u^{1*}, y^{0*}, y^{1*}, \sigma^*, \lambda')$ is feasible and its cost is not larger than that of $(u^*, u^{0*}, u^{1*}, y^{0*}, y^{1*}, \sigma^*, \lambda^*)$. The reverse inequality is even more direct, plugging the solution obtained for $\tilde{\mathcal{Y}}$ into the formulation associated to \mathcal{Y} .

B.2 Proof of Observation 5

Let $(w^*, \bar{\xi}^*, y^*, \xi^*)$ be an optimal solution to (DDID). If $y^* \neq \tilde{y}_s$, the result is immediate. Otherwise, we detail next the resulting contradiction. Notice first that (26) implies $\max_{\xi \in \Xi} \sum_{i \in [n]} (c_i + \xi_i) \tilde{y}_{s,i} > z^{\text{MM}}$, 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, $\tilde{\mathcal{Y}} \subset \mathcal{Y}$ implies $z^{\text{DDID}} \leq \tilde{z}^{\text{DDID}}$. On the other hand, we have that

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

where the third inequality arises from (26).

C. Maximization counterpart

C.1 Context

We mention next two natural ways to define a maximization counterpart to (DDID). To obtain more symmetry between minimization and maximization results, we define the maximization feasibility set as $\mathcal{Y}^{max} = \{y \in \mathbb{Z}^n \mid By \leq b, 0 \leq y \leq 1\} \subseteq \{0, 1\}^n$ and $\mathcal{P}^{max} = \{y \in \mathbb{R}^n \mid By \leq b, 0 \leq y \leq 1\} \subseteq \{0, 1\}^n$ (we have replaced $By \geq b$ with $By \leq b$ when compared to the minimization problem). The first natural maximization counterpart (studied for instance by Goetzmann et al. [2012]) considers that ξ characterizes the amount of deviation the adversary may subtract from c and thus limits that total amount using knapsack constraints, leading to the model defined as

$$\max_{w \in \mathcal{W}} \min_{\bar{\xi} \in \Xi} \max_{y \in \mathcal{Y}^{max}} \min_{\xi \in \Xi(w, \bar{\xi})} \sum_{i \in [n]} (c_i - \xi_i) y_i. \quad (37)$$

It so happens that all our results for (DDID) extend immediately to the case (37). Hence, we do not further mention (37) in what follows. However, following Paradiso et al. [2022], one can come up with an alternative model, which considers instead that ξ is a value that is added to c , thus bounded from below by knapsack constraints with reverse directions. In that context, we say that Ξ^{max} is a knapsack uncertainty polytope if it is of the form

$$\Xi^{max} = \{\xi \in \mathbb{R}^n \mid A'\xi \geq e', 0 \leq \xi \leq d\},$$

and we consider the robust counterpart

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

denoting by $\Phi^{max}(w)$ the outermost objective function

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

Following Assumption 2, we will again consider a unique constraint in the system $A'\xi \geq e'$. We detail in what follows the reformulations obtained for (DDID^{max}) , which are quite similar to those

obtained for (DDID).

C.2 Reformulations

We present first the counterpart of Proposition 2 for the maximization counterpart of (MIN-MAX),

$$\max_{y \in \mathcal{Y}} \min_{\xi \in \Xi^{max}} \sum_{i \in [n]} (c_i + \xi_i) y_i.$$

Proposition 7. *We have*

$$\min_{\xi \in \Xi^{max}} \sum_{i \in [n]} (c_i + \xi_i) y_i = \max_{\ell \in [n]_0} \left\{ \left(r - \sum_{i \in [n]} a_i d_i \right) \alpha_\ell + \sum_{i \in [n]} (c_i + \beta_i^\ell) y_i \right\},$$

where for each $\ell \in [n]$, $\alpha_\ell = 1/a_\ell$, and for each $i, \ell \in [n]$, $\beta_{\ell,i} = -d_i([a_i/a_\ell - 1]^+ - a_i/a_\ell)$, while $\alpha_0 = 0$ and $\beta_0 = 0$.

Proof. Similar to the proof of Proposition 2. □

We follow with the reformulation of $\Phi^{max}(w)$.

Proposition 8. *If $\text{conv}(\mathcal{Y}^{max}) = \mathcal{P}^{max}$, then*

$$\Phi^{max}(w) = \left\{ \begin{array}{l} \min \quad \omega \\ \text{s.t.} \quad \omega \geq \left(r - \sum_{i \in \mathcal{W}^0} a_i d_i - \sum_{i \in \mathcal{W}^1} a_i \bar{\xi}_i \right) \alpha_\ell + \sum_{i \in [n]} b_i(w) \lambda_{\ell,i}, \quad \forall \ell \in [n]_0 \\ (B_{\cdot,i})^T \lambda_\ell \geq c_i + \bar{\xi}_i, \quad \forall \ell \in [n]_0, \forall i \in \mathcal{W}^1 \\ (B_{\cdot,i})^T \lambda_\ell \geq c_i + \beta_{\ell,i}, \quad \forall \ell \in [n]_0, \forall i \in \mathcal{W}^0 \\ \bar{\xi} \in \Xi^{max} \\ \lambda \geq 0, \end{array} \right.$$

where for each $\ell \in [n]$, $\alpha_\ell = 1/a_\ell$, and for each $i, \ell \in [n]$, $\beta_{\ell,i} = -d_i([a_i/a_\ell - 1]^+ - a_i/a_\ell)$, while $\alpha_0 = 0$ and $\beta_0 = 0$.

Proof. The proof follows the exact same lines as the proofs of Theorem 1, where only Proposition 2 is replaced by Proposition 7. □

We end up with the compact reformulation for (DDID^{max}).

Proposition 9. *If $\text{conv}(\mathcal{Y}^{\max}) = \mathcal{P}^{\max}$, then (DDID^{max}) is equivalent to*

$$\begin{aligned}
& \max \sum_{\ell \in [n]_0} \left(\left[ru_\ell - \sum_{i \in [n]} a_i d_i u_{\ell,i}^0 \right] \alpha_\ell + \sum_{i \in [n]} c_i y_{\ell,i} + \sum_{i \in [n]} \beta_{\ell,i} y_{\ell,i}^0 \right) - \sum_{i \in [n]} d_i \sigma_i \\
& \text{s.t. } \sum_{\ell \in [n]_0} u_\ell = 1 \\
& \quad -\sigma_i \leq -a_i \sum_{\ell \in [n]_0} \alpha_\ell u_{\ell,i}^1 + \sum_{\ell \in [n]_0} y_{\ell,i}^1, \quad \forall i \in [n] \\
& \quad By_\ell \leq u_\ell b, \quad \forall \ell \in [n]_0 \\
& \quad u_\ell \leq 1 - w_\ell, \quad \forall \ell \in [n] \\
& \quad u_\ell = u_{\ell,i}^0 + u_{\ell,i}^1, \quad \forall \ell \in [n]_0, i \in [n], \\
& \quad \sum_{\ell \in [n]_0} u_{\ell,i}^0 \leq 1 - w_i, \quad \forall i \in [n], \\
& \quad \sum_{\ell \in [n]_0} u_{\ell,i}^1 \leq w_i, \quad \forall i \in [n], \\
& \quad Gu_\ell^1 \leq u_\ell g, \quad \forall \ell \in [n]_0, \\
& \quad y_{\ell,i} = y_{\ell,i}^0 + y_{\ell,i}^1, \quad \forall \ell \in [n]_0, i \in [n] \\
& \quad y_{\ell,i}^0 \leq u_{\ell,i}^0, \quad \forall \ell \in [n]_0, i \in [n] \\
& \quad y_{\ell,i}^1 \leq u_{\ell,i}^1, \quad \forall \ell \in [n]_0, i \in [n] \\
& \quad w \in W \\
& \quad u, v, x, y, y^0, y^1, \sigma \geq 0,
\end{aligned} \tag{38}$$

Proof. The proof follows the exact same lines as the proofs of Proposition 3, 4 and 5. \square

D. 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

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

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

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

Applying [Vayanos et al., 2022, Corollary 1] to (K -ADAPT) together with the symmetry breaking constraints detailed in Section EC.3.1. of Vayanos et al. [2022], 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 $\bar{\gamma}_i^k = w_i \gamma_i^k$ with indicator constraints to avoid the burden of computing tight big M . Furthermore, we define $\Gamma' = \Gamma + \sum_{i \in [n]} c_i/d_i$.

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

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

Table 6: Instances taken from Paradiso et al. [2022], N being equal to n .

instance	δ	root gap	$ \mathcal{Y}^{max} $		% time generating \mathcal{Y}^{max}	
			conv	CG	conv	CG
TS2N10	0.25	2.2	12	6.6	36	3.3
	0.5	1.8	12	9.2	35	2.5
	0.75	0.0	12	10	38	1.6
TS1N15	0.25	1.4	80	8.6	12	5.6
	0.5	0.73	80	15	16	6.2
	0.75	0.0	80	24	16	5.7
TS3N16	0.25	4.0	121*	19	15*	1.2
	0.5	1.0	121*	33	20*	0.98
	0.75	0.03	121*	34	16*	0.64
TS2N19	0.25	2.6	898*	13	0.37*	1.5
	0.5	2.6	898*	49	2.1*	0.23
	0.75	0.04	898*	55	2.1*	0.15
TS1N30	0.25	0.5	2132*	36	29*	1.0
	0.5	0.62	2132*	86	35*	0.69
	0.75	0.01	2132*	190	26*	0.64
TS3N31	0.25	1.7	3088*	66	4.4*	0.17
	0.5	0.58	3088*	154	5.4*	0.16
	0.75	0.0	3088*	237	6.6*	0.04

Table 7: Additional average statistics on **conv** and **CG** on solved instances, mark “*” indicates there are unsolved instances in the group.

E. Additional details for the orienteering problem

The details of the instances used in our experiments are displayed in Table 6.

Table 7, displaying the root gap of **CG**, the number of paths generated by both algorithms and the average fraction of time spent generating these paths. The table illustrates again the excellent root gap of the MILP reformulation, which decreases as the amount of possible observation increases. It also shows how small instances lead on average to sets \mathcal{Y} of small cardinality, explaining the very quick solution times of **conv** for the small instances.