Exact and Approximate Schemes for Robust Optimization Problems with Decision Dependent Information Discovery

Paradiso, Rosario; Georghiou, Angelos; Dabia, Said; Tönissen, Denise

2022

Exact and Approximate Schemes
for Robust Optimization Problems
with Decision Dependent Information Discovery

Rosario Paradiso

Department of Operations Analytics, Vrije Universiteit Amsterdam, the Netherlands, r.paradiso@vu.nl

Angelos Georghiou

Department of Business and Public Administration, University of Cyprus, Cyprus, georghiou.angelos@ucy.ac.cy

Said Dabia

Department of Operations Analytics, Vrije Universiteit Amsterdam, the Netherlands, s.dabia@vu.nl

Denise Tönissen

Department of Operations Analytics, Vrije Universiteit Amsterdam, the Netherlands, d.d.tonissen@vu.nl

Uncertain optimization problems with decision dependent information discovery allow the decision maker to control the timing of information discovery, in contrast to the classic multistage setting where uncertain parameters are revealed sequentially based on a prescribed filtration. This problem class is useful in a wide range of applications, however, its assimilation is partly limited by the lack of efficient solution schemes. In this paper we study two-stage robust optimization problems with decision dependent information discovery where uncertainty appears in the objective function. The contributions of the paper are twofold: (i) we develop an exact solution scheme based on a nested decomposition algorithm, and (ii) we improve upon the existing K-adaptability approximate by strengthening its formulation using techniques from the integer programming literature. Throughout the paper we use the orienteering problem as our working example, a challenging problem from the logistics literature which naturally fits within this framework. The complex structure of the routing recourse problem forms a challenging test bed for the proposed solution schemes, in which we show that exact solution method outperforms at times the K-adaptability approximation, however, the strengthened K-adaptability formulation can provide good quality solutions in larger instances while significantly outperforming existing approximation schemes even in the decision independent information discovery setting. We leverage the effectiveness of the proposed solution schemes and the orienteering problem in a case study from Alrijne hospital in the Netherlands, where we try to improve the collection process of empty medicine delivery crates by co-optimizing sensor placement and routing decisions.

Key words: robust optimization, decision dependent information discovery, exact method, K-adaptability, orienteering problem, sensor placement, Alrijne hospital.

1. Introduction

Robust optimization has proven to be a successful methodology to tackle optimization problems under uncertainty. In the last decades, rich and consolidated theory has been developed for the case in which all the decisions are taken *before* uncertainty is unfolded, i.e., *single stage robust optimization* with *here-and-now decisions* (Ben-Tal et al. (2009), Bertsimas et al. (2011), Gabrel et al. (2014)). When some decisions can be adjusted after observing uncertainty (*recourse* or *wait-and-see* decisions), the problem falls into the domain of *multistage robust optimization*. These problems not only can capture the dynamic nature of the process but they can also mitigate the conservatism of the static counterpart. On the other hand, they are noticeably more challenging and still resist to solution in their general form. A rich stream of research tries to tackle problems by approximating the functional form of recourse decisions, known as the decision rule approximation. Decision rule approximations have been developed for continuous (Ben-Tal et al. (2004), Chen and Zhang (2009), Georghiou et al. (2015), Bertsimas et al. (2019)) as well as binary (Bertsimas and Georghiou (2015), Postek and Hertog (2016), Bertsimas and Dunning (2016), Bertsimas and Georghiou (2018)) recourse decisions, see Delage and Iancu (2015) for a comprehensive review. When the problem features binary recourse decisions, one of the most successful methodologies is the K-adaptability approximation (Bertsimas and Caramanis (2010), Hanasusanto et al. (2015), Buchheim and Kurtz (2017), Chassein et al. (2019), Subramanyam et al. (2019)). This approximation selects K candidate constant policies before observing the realization of the uncertain parameters and then implements the best of these policies after the realization is known.

In the classic multistage setting, it is assumed that uncertain parameters are sequentially revealed to the decision-maker based on a prescribed filtration. Yet, this assumption fails to hold in a number of applications where the actions of the decision-maker can influence the timing of information discovery. For example, in *clinical trials* the success of the clinical study can only be revealed after the pharmaceutical has chosen to undertake the study, in *offshore gas-field developments* the quantity of gas reserves can only be observed if an exploratory drill is performed, and in *production planning* the production cost of a new product is uncertain and cannot be observed until the company choose to produce it (see Table 1 for more examples). This class of problems is termed as problems with *decision dependent information discovery* (Vayanos et al. (2011, 2020a)). The majority of these applications are modeled through the lens of stochastic programming

where controlling the non-anticipative structure of the decisions is translated as controlling the structure of the scenario tree using binary decisions (Goel and Grossmann (2006), Gupta and Grossmann (2011), Boland et al. (2016)). In the robust optimization setting Vayanos et al. (2011) leverages the decision rule approximation scheme to control the timing of information discovery, while Vayanos et al. (2020a) provide a more scalable and accurate approximation by extending the K-adaptability approximation to the multistage decision dependent information discovery setting.

Many applications from the field of logistics can be cast as problems with decision dependent information discovery. Yet, to the best of our knowledge, these synergies have not been explored with the exception of Macias et al. (2020) which models the monitoring and relief of a humanitarian distribution network through the use of UAVs. In this paper, we use as our working example the *orienteering problem* which is a challenging logistics problem. First introduced by Tsiligirides (1984), the deterministic orienteering problem is combination of the knapsack and the traveling salesman problem described as follows: given a set of locations each with an assigned profit, the objective is to compute the path that maximizes the total profit collected while not violating the time duration constraint. In its basic form, the orienteering problem has found many applications including distribution, maintenance, tourism, mobile crowd-sourcing and military applications, see the review papers (Vansteenwegen et al. (2011), Gunawan et al. (2016)) for further applications and variants of the problem. Stochastic or robust variants of the orienteering problem have been proposed in (Campbell et al. (2011), Evers et al. (2014a,b), Verbeeck et al. (2016)), however, all works consider the information discovery being independent of the actions of the decision maker. Yet, many logistic problems could benefit from this modeling approach. For example, the tourist trips design problem (Vansteenwegen and Van Oudheusden (2007), Gavalas et al. (2014)) can be combined with the preference elicitation problem (Bertsimas and O'Hair (2013), Vayanos et al. (2020b)) to learn the uncertain preferences of individual tourists thus being able to propose personalized sightseeing recommendations. In this two-stage robust decision dependent information discovery setting, the here-and-now discovery decisions are the questions used to "observe" the unknown "rewards" experienced by the tourist at each attraction, while the recourse actions decide the routing plan the tourist will follow to visit the attractions which is formulated as an orienteering problem. An example which we leverage in this study, and which is motivated by the delivery of medicine in the Alrijne hospital in the Netherlands (Joosten (2019), Veen (2020)), tries to improve the

collection process of empty medicine delivery crates. In this two-stage setting, the here-and-now discovery decisions are placements of sensors in different departments of the hospital to detect the uncertain number of empty crates while the recourse actions decide the collection route which we formulated as an orienteering problem.

Table 1   Literature overview on works that include applications featuring decision dependent information discovery.

Citations	Methodology	Application	Discovery Decisions
Colvin and Maravelias (2008, 2010)	Stochastic programming	Pharmaceutical clinical trial planning	Clinical trials
Jonsbråten et al. (1998)	Stochastic programming	Offshore gas field developments	Drilling process
		Production planning	Production start times
Goel and Grossmann (2004)	Stochastic programming	Offshore gas field developments	Drilling process
Vayanos et al. (2011)	Robust optimization, Stochastic programming	Offshore gas field developments	Drilling process
Solak et al. (2010)	Stochastic programming	Offshore gas field developments	Drilling process
Fragnière et al. (2010)	Stochastic programming	Workforce capacity planning	Projects start times, resources allocation
Macias et al. (2020)	Stochastic programming	Humanitarian relief distribution	UAVs to asses the state of the network
Vayanos et al. (2020a)	Robust optimization	Active preference elicitation	Questions for users

Decision dependent information discovery problems are inherently difficult to solve. This complexity stems primarily from the complexity of the recourse policy. The most successful approximation in the literature is based on K-adaptability (Vayanos et al. (2020a)) which can provide good quality solutions if K is sufficiently large, however, at the expense of significant computational time. To obtain good and meaningful solutions when the recourse problem is complex, as it is the case for the orienteering and other logistics problem, exact methods are typically useful to accurately capture the recourse policy. However, approximation schemes can be the only practical solution method for large scale problems. It is therefore also important to improve upon the scalability of the K-adaptability. The contribution of this paper can be summarized as follows:

1. We propose an exact solution scheme for two-stage robust decision dependent information discovery problems with objective uncertainty. The approach is a nested decomposition scheme that operates in two levels: (i) an outer layer optimizing over the decisions controlling the timing of information discovery in which we leverage Logic Benders decomposition, and (ii) an inner layer optimizing

over the recourse decisions and determines their worst-case cost which we solve using a column-and-constraint generation algorithm. We discuss convergence properties of the algorithm and show that the algorithm is practical and at times outperforms the K-adaptability both in solution quality and computational time.

2. We improve upon the K-adaptability formulation proposed in Vayanos et al. (2020a) by strengthening the formulation to achieve greater scalability. To this end, we propose two types of valid inequalities. The first leverages on the symmetry of the recourse problem to strengthen the McCormick inequalities used in the linearization of bilinear terms, and the second provides valid inequalities by identified a-priori "optimistic" realizations of the uncertainty parameters. We emphasize that the proposed formulation can also be used in the decision independent information discovery setting. We demonstrate the merits of the strengthened formulation numerically both in the decision dependent and decision decision independent information discovery setting.

3. We use the proposed solution methods on a real problem faced at the Alrijne hospital in the Netherlands that aims to optimize the collection of medicine carrying crates from different departments of the hospital. We formulate the problem as a two-stage robust orienteering problem with decision dependent information discovery. We solve the problem using both the exact solution scheme as well as K-adaptability approximation and discuss the interplay between degree of information discovery, the level of adaptability and the setup of the orienteering problem to provide guidelines on how the hospital should optimize its crate collection process.

The remainder of the paper is organized as follows. In Section 2 we introduce the two-stage robust decision dependent information discovery problem with objective uncertainty and motivate the connection to the orienteering problem. Section 3 presents the exact solution scheme, and Section 4 presents the strengthened K-adaptability formulation. Section 5 provides numerical evidence of the usefulness of the proposed methods and Section 6 discusses the case study of Alrijne hospital. The proofs of all statements can be found in the Electronic Companion to the paper.

Notation: Vector and matrices are represented with bold lowercase and uppercase letters, respectively. We denote be e the vector of ones and by 0 the vector of zeros. For a generic vector x, N_x denotes the size of the vector while $\mathcal{N}_x$ denotes its indices. Given two vectors $x, y \in \mathbb{R}^n$, $x \circ y$ denotes the Hadamard product. For a logical expression $\mathcal{L}$, we define the indicator function $\mathbb{I}[\mathcal{L}]$ that takes value 1 if $\mathcal{L}$ is true and 0 otherwise.

2. Problem Description

We study a two-stage robust optimization problem with *decision depended information discovery with objective uncertainty*. In the classical setting of two-stage robust optimization problems without decision dependent information discovery, after taking the first-stage (or here-and-now) decisions, the decision-maker is able to observe the whole of the uncertain vector $\boldsymbol{\xi} \in \Xi \subseteq \mathbb{R}^{N_\xi}$ and utilize this information to decide the recourse (or second-stage) decisions. In contrast, in the decision dependent information discovery setting the first-stage decisions $\boldsymbol{w} \in \mathcal{W} \subseteq \{0,1\}^{N_w}$ control the timing of information discovery. In other words, the recourse decisions $\boldsymbol{y} \in \mathcal{Y} \subseteq \{0,1\}^{N_y}$ are able to only adapt to components of the uncertain vector $\boldsymbol{\xi}$ if the decision-maker chooses to observe these components using $\boldsymbol{w}$, i.e., if $\boldsymbol{w}_i = 1$, the recourse decisions can adapt to $\boldsymbol{\xi}_i$, however, if $\boldsymbol{w}_i = 0$ the decisions cannot assimilate $\boldsymbol{\xi}_i$ which is treated as if it materializes after decisions $\boldsymbol{y}$ have been decided. Since each uncertain parameter is associated with a specific component of $\boldsymbol{w}$, we set $N_w = N_\xi$. The problem is formulated as follows:

$$\min_{\boldsymbol{w} \in \mathcal{W}} \Phi(\boldsymbol{w}), \tag{1a}$$

with

$$\Phi(\boldsymbol{w}) = \max_{\overline{\boldsymbol{\xi}} \in \Xi} \min_{\boldsymbol{y} \in \mathcal{Y}} \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \overline{\boldsymbol{\xi}})} \boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^\top \boldsymbol{P} \boldsymbol{y}, \tag{1b}$$

where we assume $\Xi := \{\boldsymbol{\xi} \in \mathbb{R}^{N_\xi} : \boldsymbol{A}\boldsymbol{\xi} \leq \boldsymbol{b}\}$ for some matrices $\boldsymbol{A} \in \mathbb{R}^{L \times N_\xi}$, $\boldsymbol{C} \in \mathbb{R}^{N_\xi \times N_w}$, $\boldsymbol{P} \in \mathbb{R}^{N_\xi \times N_y}$ and vector $\boldsymbol{b} \in \mathbb{R}^L$. Problem (1) imposes decision dependent information discovery by utilizing an auxiliary uncertain vector $\overline{\boldsymbol{\xi}} \in \Xi$ and set $\Xi(\boldsymbol{w}, \overline{\boldsymbol{\xi}}) := \{\boldsymbol{\xi} \in \Xi : \boldsymbol{w} \circ \boldsymbol{\xi} = \boldsymbol{w} \circ \overline{\boldsymbol{\xi}}\}$ which is a subset of the uncertainty set Ξ. Vector $\overline{\boldsymbol{\xi}}$ can be thought of as an independent uncertain vector that materializes before decisions $\boldsymbol{y}$ are decided, yet, since constraint $\boldsymbol{w} \circ \boldsymbol{\xi} = \boldsymbol{w} \circ \overline{\boldsymbol{\xi}}$ imposes that $\overline{\boldsymbol{\xi}}_i = \boldsymbol{\xi}_i$ if $\boldsymbol{w}_i = 1$, the recourse decisions cannot adapt to what has not been observed. The inner most maximization in (1b) accounts for the worst-case cost for the components of $\boldsymbol{\xi}$ which have not been observed, i.e., $\boldsymbol{w}_i = 0$. For each $\boldsymbol{w}$, $\Phi(\boldsymbol{w})$ represents the cost of the recourse problem. For further information, motivation and construction of problem (1b) we refer to Vayanos et al. (2020a).

Although the approaches we develop in this paper are applicable to any problem that can take the form (1), we now focus on an instance of the problem in which decisions $\boldsymbol{y} \in \mathcal{Y}$ encode feasible paths of an orienteering problem. The orienteering problem is a routing problem that aims to generate a routing path

through a set of nodes such that it maximizes the profit associated with the visited nodes, while satisfying a given maximum travel time duration (Tsiligirides (1984), Vansteenwegen et al. (2011), Gunawan et al. (2016)). The problem is defined by: (i) a complete undirected graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ where $\mathcal{N} := \{0, 1, \ldots, |\mathcal{N}|\}$ is the set of nodes and $\mathcal{E} := \{(i, j) \in \mathcal{N} \times \mathcal{N} : i < j\}$ is the set of edges, (ii) travel times $\boldsymbol{t}_e$ associated with each edge $e = (i, j) \in \mathcal{E}$ and a maximum travel time duration T, (iii) a start and destination nodes denoted by s and d, respectively and (iv) a profit $\boldsymbol{\xi}_i$ for each node $i \in \tilde{\mathcal{N}} := \mathcal{N} \setminus \{s, d\}$. The set of all feasible paths can be formally defined as follows

$$
\mathcal{Y} = \left\{
\begin{array}{c}
\boldsymbol{y} \in \{0,1\}^{|\tilde{\mathcal{N}}|} : \exists \boldsymbol{z} \in \{0,1\}^{|\mathcal{E}|} \text{ s.t. } \boldsymbol{t}^\top \boldsymbol{z} \leq T, \sum_{e \in C(s)} \boldsymbol{z}_e = \sum_{e \in \mathcal{C}(d)} \boldsymbol{z}_e = 1 \\
\sum_{e \in \mathcal{C}(i)} \boldsymbol{z}_e = 2\boldsymbol{y}_i \ \forall i \in \tilde{\mathcal{N}}, \ \sum_{e \in \mathcal{U}(\mathcal{S})} \boldsymbol{z}_e \leq \sum_{i \in \mathcal{S}} \boldsymbol{y}_i - \boldsymbol{y}_u \ u \in \mathcal{S}, \forall \mathcal{S} \subseteq \tilde{N}
\end{array}
\right\},
\tag{2}
$$

where, given a subset of nodes $\mathcal{I} \subset \mathcal{N}$ we indicate the edge cutset of set $\mathcal{I}$ as $\mathcal{C}(\mathcal{I}) := \{(i, j) \in \mathcal{E} : i \in \mathcal{I}, j \notin \mathcal{I} \lor j \in \mathcal{I}, i \notin \mathcal{I}\}$ (i.e., the set of edges (i, j) such that $i \in I, j \notin I$ or vice-versa) and with $\mathcal{U}(I)$ the set of edges such that the associated two nodes belong to I, i.e., $\mathcal{U}(I) := \{(i, j) \in \mathcal{E} : i \in \mathcal{I}, j \in \mathcal{I}\}$. For the sake of notation, we also indicate with $\mathcal{C}(i)$ the edge cutset of a single node $i \in \mathcal{N}$. In this representation of set $\mathcal{Y}$, the binary variables $\boldsymbol{y}$ indicate which nodes are visited and binary variable $\boldsymbol{z}_e$ takes value 1 if edge e is traversed and 0 otherwise. Although a compact formulation for the orienteering problem is possible by formulating subtours elimination constraints according to the Miller-Tucker-Zemlin formulation (Miller et al. (1960)), $\mathcal{Y}$ is defined using a formulation similar to the one used in Fischetti et al. (1998). Such a formulation features an exponential number of constraints and therefore, even for the deterministic case must be solved via a branch-and-cut algorithm. However, it typically results in stronger linear relaxations than its compact counterpart.

In the decision dependent information discovery setting, the profits $\boldsymbol{\xi}_i$ at each node $i \in \tilde{\mathcal{N}}$ are treated as uncertain and are elements of a known uncertainty set Ξ. In the first period, the decision maker is able to "place sensors", corresponding to decisions $\boldsymbol{w}$, on nodes so as to observe their associated profit. Using the obtained information, the routing decisions $\boldsymbol{y}$ are taken in view of the worst-case profits for those nodes whose profits are not observed. The decision dependent information discovery problem, which we term *sensor placement orienteering problem*, can be written as

$$
\max_{\boldsymbol{w} \in \mathcal{W}} \min_{\overline{\boldsymbol{\xi}} \in \Xi} \max_{\boldsymbol{y} \in \mathcal{Y}} \min_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \overline{\boldsymbol{\xi}})} \boldsymbol{\xi}^\top \boldsymbol{y}.
\tag{3}
$$

In the following simple example, we showcase that sensor placement can significantly impact the routing decisions, thus concluding of the need to co-optimizing sensor placements and routing decisions.

EXAMPLE 1. We consider an instance of the orienteering problem where $\mathcal{N} = \{0, 1, 2, 3\}$ (0 is both the start and destination node), $\boldsymbol{t}_{01} = \boldsymbol{t}_{03} = \boldsymbol{t}_{02} = 1$, $\boldsymbol{t}_{12} = \boldsymbol{t}_{13} = \sqrt{2}$, $\boldsymbol{t}_{13} = 2$, and the maximum travel time $T = 3.5$. In this setting, the feasible routes $\boldsymbol{y}$ implied by constraints (2) consist of all the paths visiting at most two nodes, i.e., $\mathcal{Y} = \{(0,0,0)^{\top}, (1,1,0)^{\top}, (0,1,1)^{\top}, (1,0,0)^{\top}, (0,1,0)^{\top}, (0,0,1)^{\top}\}$. The profit at each node is represented by the uncertainty vector $\boldsymbol{\xi}$ that belongs to the uncertainty set $\Xi = \{\boldsymbol{\xi} \in \mathbb{R}^3_+ : \boldsymbol{e}^{\top}\boldsymbol{\xi} = 1\}$, modeling that the total profit (100%) throughout the network is known, however, its geographical breakdown at each node is uncertain. The sensor placement problem maximizes $\Phi(\boldsymbol{w})$ over the set of admissible sensor placements $\mathcal{W} = \{\boldsymbol{w} \in \mathbb{R}^3 : \boldsymbol{e}^{\top}\boldsymbol{w} \le B\}$, where B is the maximum number of sensors allowed to be placed throughout the network. The profit $\Phi(\boldsymbol{w})$ associated with each placement is

$$\Phi(\boldsymbol{w}) = \min_{\overline{\boldsymbol{\xi}} \in \Xi} \max_{\boldsymbol{y} \in \mathcal{Y}} \min_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \overline{\boldsymbol{\xi}})} \boldsymbol{\xi}_1 \boldsymbol{y}_1 + \boldsymbol{\xi}_2 \boldsymbol{y}_2 + \boldsymbol{\xi}_3 \boldsymbol{y}_3. \tag{4}$$

For $B = 0$ the only admissible sensor placement is $\boldsymbol{w} = \boldsymbol{0}$, i.e., no sensors are placed, which implies that $\Xi(\boldsymbol{w}, \overline{\boldsymbol{\xi}}) = \Xi$ for all $\overline{\boldsymbol{\xi}} \in \Xi$. This leads to a robust orienteering problem where the decision maker decides the routing decisions in view of the worst-case realization of $\boldsymbol{\xi} \in \Xi$. As the admissible routes allow to visit at most two nodes, the worst-case allocates all profit to the non-visited node, resulting in the worst-case profit of 0 for any routing decision.

For $B = 3$, $\boldsymbol{w} = (1, 1, 1)^{\top}$ is admissible which implies placing sensors in all nodes, thus $\Xi(\boldsymbol{w}, \overline{\boldsymbol{\xi}}) = \{\overline{\boldsymbol{\xi}}\}$ for all $\overline{\boldsymbol{\xi}} \in \Xi$. Intuitively, $\boldsymbol{w} = (1, 1, 1)^{\top}$ is an optimal solution, as the routing decisions will be chosen in full knowledge of the worst-case profits in the network, i.e., more information is better for the recourse decisions. This is formalized in Lemma 1 in the following section. As demonstrated in Figure 1, the routing is a policy of the realizations of $\boldsymbol{\xi}$. If $\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 \ge \boldsymbol{\xi}_2 + \boldsymbol{\xi}_3$ (Figure 1a), nodes 1 and 2 are visited, while if $\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 \le \boldsymbol{\xi}_2 + \boldsymbol{\xi}_3$ the decision-maker select the path that visits node 2 and node 3 (Figure 1a). In this case, the adversarial nature allocates profit such that $\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 = \boldsymbol{\xi}_2 + \boldsymbol{\xi}_3$, resulting in the worst-case profit of 0.5, or 50% of the total profit.

When $B = 1$, there are four admissible sensor placements $\boldsymbol{w} = (0, 0, 0)^{\top}$, $\boldsymbol{w} = (1, 0, 0)^{\top}$, $\boldsymbol{w} = (0, 1, 0)^{\top}$ and $\boldsymbol{w} = (0, 0, 1)^{\top}$. For $\boldsymbol{w} = (0, 1, 0)^{\top}$, i.e., placing a sensor only in node 2, regardless of the optimal routing

(a) Optimal route if $\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 \geq \boldsymbol{\xi}_2 + \boldsymbol{\xi}_3$.

(b) Optimal route if $\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 \leq \boldsymbol{\xi}_2 + \boldsymbol{\xi}_3$.
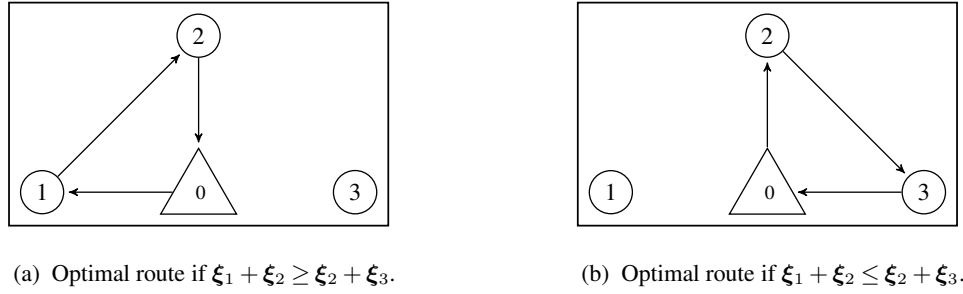
Figure 1 Routing policy for $\boldsymbol{w} = (1,1,1)^\top$, collecting in the worst-case 50% of the total profit.

policy the worst-case profit is 0. This is the case as the worst-case profit realization of node 2 is zero, which is represented by $\overline{\boldsymbol{\xi}}_2 = 0$, implying that $\Xi(\boldsymbol{w}, \overline{\boldsymbol{\xi}}) = \{\boldsymbol{\xi} \in \mathbb{R}^3_+ : \boldsymbol{\xi}_1 + \boldsymbol{\xi}_3 \leq 1, \ \boldsymbol{\xi}_2 = \overline{\boldsymbol{\xi}}_2 = 0\}$. Thus, the total profit can be placed either in node 1 or node 3, however, since the routing decisions are lacking this information and since the construction of $\mathcal{Y}$ allows only one of nodes 1 and 3 to be visited, in the worst-case none of the profit is collected as shown in Figure 2. We now consider placing a sensor on node 1, $\boldsymbol{w} = (1,0,0)^\top$. Due to the maximum time duration $T = 3.5$, the admissible routing decisions are allowed to visit either nodes 1, or nodes 2 and 3. The optimal routing policy is: if $\boldsymbol{\xi}_1 \geq 0.5$ then visit node 1, while if $\boldsymbol{\xi}_1 \leq 0.5$ then visit nodes 2 and 3, as shown in Figure 3. The corresponding worst-case realization sets $\overline{\boldsymbol{\xi}}_1 = 0.5$, implying $\Xi(\boldsymbol{w}, \overline{\boldsymbol{\xi}}) = \{\boldsymbol{\xi} \in \mathbb{R}^3_+ : \boldsymbol{\xi}_2 + \boldsymbol{\xi}_3 = 0.5, \ \boldsymbol{\xi}_1 = \overline{\boldsymbol{\xi}}_1 = 0.5\}$, which results in a worst-case profit of 0.5. Note that due to the topology of the network, the solutions for $\boldsymbol{w} = (1,0,0)^\top$ and $\boldsymbol{w} = (0,0,1)^\top$ are symmetric thus they achieve the same objective value. Since $B = 1$ and $B = 3$ both achieve an optimal profit of 0.5, $B = 2$ will result in the same worst-case profit.

(a) Worst-case scenario if route only visits node 1.

(b) Worst-case scenario if route only visits node 3.
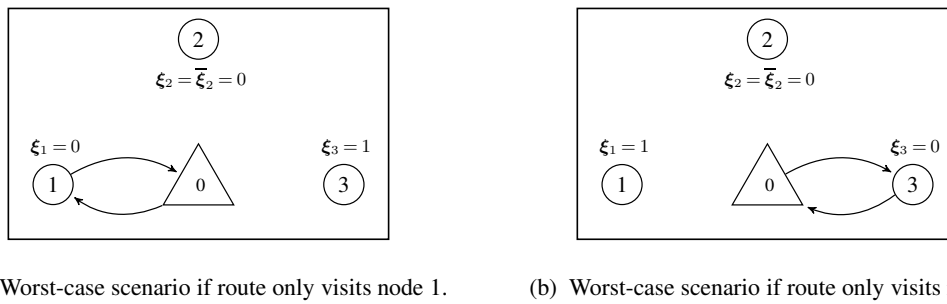
Figure 2 Worst-case scenarios for $\boldsymbol{w} = (0,1,0)^\top$, i.e., observing $\boldsymbol{\xi}_2$. All routing policies result to a worst-case profit of 0.

The example demonstrates that optimally placing sensors on only a subset of the nodes is sometimes sufficient for achieving the same optimal worst-case profit as placing sensors on all nodes. However, mis-placing sensors can lead to the same worst-case profit as having no sensors in the network. Therefore, there

(a) Optimal route if $\boldsymbol{\xi}_1 \geq 0.5$.

(b) Optimal route if $\boldsymbol{\xi}_1 \leq 0.5$.
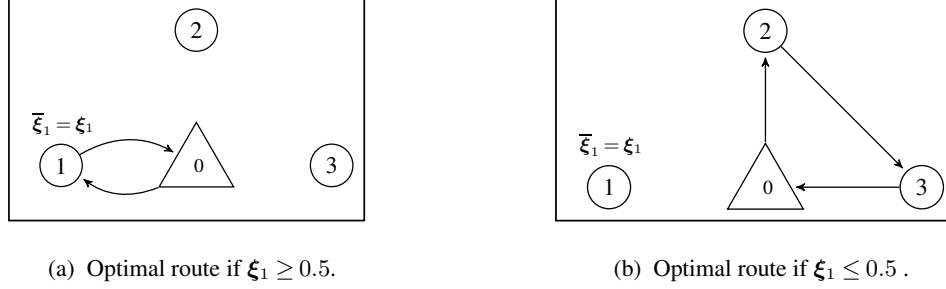
Figure 3   Routing policy for $\boldsymbol{w} = (1, 0, 0)$, i.e., observing $\boldsymbol{\xi}_1$, collecting in the worst-case 50% of the total profit.

is a need for optimizing over the sensor placement which can have a significant impact on the routing policies. In the following section we propose an exact procedure for optimally solving problem (1).

3. Exact Algorithm for Problem (1)

In this section we propose an exact algorithm for solving problem (1). Utilizing ideas from Logic Benders we iteratively underestimate $\Phi(\boldsymbol{w})$ using linear inequalities. To generate an inequality, the algorithm requires to exactly evaluate $\Phi(\boldsymbol{w})$ for each $\boldsymbol{w}$, and since (1b) is in essence a two-stage linear robust optimization we design a column-and-constraint generation algorithm to do so. In Section 3.1 we discuss the properties of $\Phi(\boldsymbol{w})$ and propose valid inequalities and in Section 3.2 we present the column-and-constraint generation algorithm for evaluating $\Phi(\boldsymbol{w})$. Section EC.2 in the Electronic Companion discusses extensions to accommodate a richer class of two-stage robust optimization problems with decision dependent information discovery.

3.1. Logic Benders Decomposition for Problem (1a)

Function $\Phi(.)$ is a real valued function which is in general not convex on $[0, 1]^{N_{\boldsymbol{w}}}$. We are however interested in the values of the function at the binary points $\{0, 1\}^{N_{\boldsymbol{w}}}$. Our decomposition leverages on the fact that the convex lower envelope of $\Phi(.)$ is a convex piece-wise linear function on $[0, 1]^{N_{\boldsymbol{w}}}$ which exactly approximate $\Phi(.)$ at all the binary points (Zou et al. 2019, Theorem 1). The convex lower envelope can be described by the following set of inequalities

$$\Phi(\boldsymbol{w}) \geq \Phi(\boldsymbol{w}') - (\Phi(\boldsymbol{w}') - \underline{\Phi}) \, \phi(\boldsymbol{w}', \boldsymbol{w}) \quad \forall \boldsymbol{w}' \in \mathcal{W}, \tag{5a}$$

where

$$\phi(\boldsymbol{w}', \boldsymbol{w}) = \sum_{i \in \mathcal{N}_{\boldsymbol{w}} : \, \boldsymbol{w}'_i = 1} (1 - \boldsymbol{w}_i) + \sum_{i \in \mathcal{N}_{\boldsymbol{w}} : \, \boldsymbol{w}'_i = 0} \boldsymbol{w}_i, \tag{5b}$$

and $\underline{\Phi} \in \mathbb{R}$ such that $\underline{\Phi} \leq \Phi(\boldsymbol{w})$ for all $\boldsymbol{w} \in \mathcal{W}$, i.e., any lower bound on $\Phi(\boldsymbol{w})$. This family of inequalities is one of the main ingredient of Logic Benders based decomposition algorithms. Logic Benders decomposition was formally introduced in Hooker and Ottosson (2003) and is usually applied to two-stage, e.g., Carøe and Tind (1998), Laporte and Louveaux (1993), and multi-stage Zou et al. (2019) stochastic optimization problems. However, they are not widely adopted within robust optimization where decomposition approaches based on column-and-constraints generation are more common, e.g., Zeng and Zhao (2013), Ayoub and Poss (2016).

PROPOSITION 1. *Inequalities* (5) *are valid and tight at each point* $\boldsymbol{w} \in \mathcal{W}$.

In line with Benders-like decomposition scheme, we leverage on Proposition 1 to define the following relaxation of problem (1a).

$$
\Theta_{\widehat{\mathcal{W}}} = \left[\begin{array}{ll} \min \varphi & \\ \text{s.t.} & \varphi \in \mathbb{R}, \ \boldsymbol{w} \in \mathcal{W} \\ & \varphi \geq \Phi(\boldsymbol{w}') - (\Phi(\boldsymbol{w}') - \underline{\Phi}) \, \phi(\boldsymbol{w}', \boldsymbol{w}) \quad \forall \, \boldsymbol{w}' \in \widehat{\mathcal{W}} \end{array} \right], \tag{6}
$$

where $\widehat{\mathcal{W}} \subseteq \mathcal{W}$. Problem (6) includes only a subset of all the possible inequalities (5), thus $\Theta_{\widehat{\mathcal{W}}} \leq \Phi(\boldsymbol{w})$ for all $\widehat{\mathcal{W}} \subseteq \mathcal{W}$. Algorithm 1 optimally solves problem (1a) by identifying which constraints (5) the progressive solution $\boldsymbol{w}^*$ violates and iteratively add them to set $\widehat{\mathcal{W}}$.

Algorithm 1: Exact Algorithm for Problem (1a)

Output: Optimal solution $\boldsymbol{w}^*$ and optimal solution value $\Phi(\boldsymbol{w}^*)$.

1 Initialize: $ub = +\infty$, $lb = -\infty$, $\widehat{\mathcal{W}} = \emptyset$;

2 Solve Problem (6) and let $\boldsymbol{w}^*$ be the optimal solution. Set $lb = \Theta_{\widehat{\mathcal{W}}}$;

3 Evaluate $\Phi(\boldsymbol{w}^*)$ by solving (1b) and set $ub = \Phi(\boldsymbol{w}^*)$;

4 if *ub - lb > 0* **then** $\widehat{\mathcal{W}} = \widehat{\mathcal{W}} \cup \{\boldsymbol{w}^*\}$ and go to Step 2 ;

5 else return $\boldsymbol{w}^*$ and $\Theta_{\widehat{\mathcal{W}}}$;

At each iteration, problem (6) is solved to identify the progressive solution $\boldsymbol{w}^*$, and the value of the recourse problem $\Phi(\boldsymbol{w}^*)$ is evaluated by solving problem (1b). The algorithm terminates when the optimal value $\Theta_{\widehat{\mathcal{W}}}$ of the relaxed problem and $\Phi(\boldsymbol{w}^*)$ coincide, otherwise $\boldsymbol{w}^*$ is added to set $\widehat{\mathcal{W}}$ and the process is repeated. The following theorem demonstrates that the algorithm terminates in finite number of steps.

THEOREM 1. *Algorithm 1 converges to the optimal solution of problem (1a) in a finite number of iterations.*

In line with modern implementations of Benders-like algorithms (see Adulyasak et al. (2015), Fischetti et al. (2017)), rather than executing Algorithm (1), we embed the resolution of the relaxed master problem (6) in a branch-and-cut algorithm. Within this algorithm, every time a feasible solution $(\varphi^*, \boldsymbol{w}^*)$ is encountered in the search tree, function $\Phi(\boldsymbol{w}^*)$ is evaluated and if $\varphi^* < \Phi(\boldsymbol{w}^*)$, the corresponding cut is added to problem (6). The branch-and-cut algorithm ends when all the nodes are pruned and an optimal solution to problem (6) is achieved. In other words, with respect to Algorithm 1, we execute Step 2 only one time while Step 3 will be execute every time a new feasible solution $\boldsymbol{w} \in \mathcal{W}$ is founded in the search tree.

Inequalities (5) are valid and tight at every $\boldsymbol{w} \in \mathcal{W}$, therefore they ensure correctness and convergence of the algorithm. However, they can be, in general, extremely weak at fractional solution, i.e., $\boldsymbol{w} \in [0,1]^{N_{\boldsymbol{w}}}$ slowing down the resolution of problem (6). This is a typical draw-back of Logic Benders solution schemes (Hooker and Ottosson (2003), Zou et al. (2019)). In the following, we leverage on the fact that $\boldsymbol{w}$ encodes the amount of information available to the recourse decisions thus we are able to better describe the behavior of $\Phi(\boldsymbol{w})$. This will allow us to build tighter inequalities , which we term *information inequalities*, achieving faster convergence rate in practice.

3.1.1. Information Inequalities We now consider a variant of problem (1) where the costs affecting discovery variables are deterministic rather than uncertainty, i.e., $\boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} = \boldsymbol{c}^\top \boldsymbol{w}$ for a given vector $\boldsymbol{c} \in \mathbb{R}^{N_{\boldsymbol{w}}}$. In this setting, problem (1a) can be written as $\min_{\boldsymbol{w} \in \mathcal{W}} \boldsymbol{c}^\top \boldsymbol{w} + \Phi(\boldsymbol{w})$, where $\Phi(\boldsymbol{w}) = \max_{\bar{\boldsymbol{\xi}} \in \Xi} \min_{\boldsymbol{y} \in \mathcal{Y}} \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \bar{\boldsymbol{\xi}})} \boldsymbol{\xi}^\top \boldsymbol{P} \boldsymbol{y}$. The following lemma captures the intuition that as the information increases, i.e., more entries of $\boldsymbol{w}$ take values equal to 1, the decision-maker can take more informed recourse decisions, thus the value of the $\Phi(\boldsymbol{w})$ is non-increasing.

LEMMA 1. *Let $\boldsymbol{w}^1, \boldsymbol{w}^2 \in \mathcal{W}$. If $\boldsymbol{w}^1 \geq \boldsymbol{w}^2$ (where the inequality holds component wise), then $\Phi(\boldsymbol{w}^1) \leq \Phi(\boldsymbol{w}^2)$.*

It is important to note that Lemma 1 does not hold when the cost associated with $\boldsymbol{w}$ is uncertain. In this setting, the value of $\Phi(\boldsymbol{w})$ can increase as more components of $\boldsymbol{w}$ take the value one. Intuitively, this can occur as the worst-case $\boldsymbol{\xi}$ is chosen to maximize $\boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{y}$, hence the it can shift more weight on $\boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w}$.

Leveraging on Lemma 1, we propose the following *information inequalities*

$$\Phi(\boldsymbol{w}) \geq \Phi(\boldsymbol{w}') - (\Phi(\boldsymbol{w}') - \underline{\Phi})\, \rho(\boldsymbol{w}', \boldsymbol{w}) \quad \forall \boldsymbol{w}' \in \mathcal{W}, \tag{7a}$$

where

$$\rho(\boldsymbol{w}', \boldsymbol{w}) = \sum_{i \in \mathcal{N}_w : \boldsymbol{w}'_i = 0} \boldsymbol{w}_i. \tag{7b}$$

The following proposition proves the validity of the inequalities.

PROPOSITION 2. *Inequalities* (7) *are valid and tight at every feasible integer solution* $\boldsymbol{w} \in \mathcal{W}$ *and at least as tight as inequalities* (5) *at every fractional solution* $\boldsymbol{w}'' \in [0,1]^{N_{\boldsymbol{w}}}$.

As we need to evaluate $\Phi(\boldsymbol{w})$ exactly at each iteration of the algorithm, solving efficiently problem (1b) is crucial for a practical implementation. In the next section we propose a column-and-constraint generation algorithm for evaluating $\Phi(\boldsymbol{w})$.

3.2. Column-and-Constraint Generation Algorithm for Problem (1b)

Problem (1b) is a two stage robust optimization problem with objective uncertainty, where the role of the first and second stage decisions are taken by $\overline{\boldsymbol{\xi}}$ and $\boldsymbol{\xi}$, respectively, and the role of the "uncertainty" is taken by the vector $\boldsymbol{y}$. To solve this problem, we develop a column-and-constraint generation algorithm. We express $\Phi(\boldsymbol{w}) = \max_{\overline{\boldsymbol{\xi}} \in \Xi} \psi(\boldsymbol{w}, \overline{\boldsymbol{\xi}})$, where

$$\psi(\boldsymbol{w}, \overline{\boldsymbol{\xi}}) = \min_{\boldsymbol{y} \in \mathcal{Y}} \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \overline{\boldsymbol{\xi}})} \boldsymbol{\xi}^{\top} \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^{\top} \boldsymbol{P} \boldsymbol{y}. \tag{8}$$

The following problem constitutes a relaxation of problem (1b)

$$\Psi_{\widehat{\mathcal{Y}}}(\boldsymbol{w}) = \max_{\overline{\boldsymbol{\xi}} \in \Xi} \min_{\boldsymbol{y} \in \widehat{\mathcal{Y}}} \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \overline{\boldsymbol{\xi}})} \boldsymbol{\xi}^{\top} \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^{\top} \boldsymbol{P} \boldsymbol{y}$$

$$= \begin{bmatrix} \max \tau \\ \text{s.t. } \tau \in \mathbb{R}, \ \overline{\boldsymbol{\xi}} \in \Xi, \ \boldsymbol{\xi} : \mathcal{Y} \mapsto \mathbb{R}^{N_{\boldsymbol{\xi}}} \\ \tau \leq \boldsymbol{\xi}(\boldsymbol{y})^{\top} \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}(\boldsymbol{y})^{\top} \boldsymbol{P} \boldsymbol{y} \ \forall \boldsymbol{y} \in \widehat{\mathcal{Y}} \\ \boldsymbol{\xi}(\boldsymbol{y}) \in \Xi(\boldsymbol{w}, \overline{\boldsymbol{\xi}}) \qquad \forall \boldsymbol{y} \in \widehat{\mathcal{Y}} \end{bmatrix}, \tag{9}$$

where $\widehat{\mathcal{Y}} \subseteq \mathcal{Y}$, and the second equality follows from (Vayanos et al. 2020a, Theorem 1). For any subset $\widehat{\mathcal{Y}}$, the optimal value $\Psi_{\widehat{\mathcal{Y}}}(\boldsymbol{w})$ is an upper bound on $\Phi(\boldsymbol{w})$.

Algorithm 2 solves problem (1b) to optimality. The algorithm operates by iteratively solving the relaxed problem (9) to obtain the upper bound $\Psi_{\widehat{\mathcal{Y}}}(\boldsymbol{w})$ and get the progressive solution $\overline{\boldsymbol{\xi}}$, whose value is subsequently evaluated by solving problem (8) to get the lower bound $\psi(\boldsymbol{w}, \overline{\boldsymbol{\xi}})$ and the value $\boldsymbol{y}^*$ that achieves

it. If the bounds coincide, then the algorithm terminates, otherwise y^* is added in set $\widehat{\mathcal{Y}}$ and the process is repeated. Notice that for given w and $\overline{\xi}$ problem (8) is a static robust optimization problem and can be either reformulated as a single mixed-integer linear program by dualizing the inner maximization, or can be solve through a constraint generation algorithm, see Bertsimas et al. (2016) for a comparison between the two approaches. In our work, we follow the former approach. The algorithm is termed column-and-constraint generation as for every y^* added in set $\widehat{\mathcal{Y}}$ new constraints are added in problem (9), as well as new decision variables ("columns") through $\xi(y)$. The following theorem demonstrates that the algorithm terminates in finite number of steps.

Algorithm 2: Exact Column-and-Constraint Generation Algorithm for Problem (1b)

Input : A vector $w \in \mathcal{W}$.

Output: Value of $\Phi(w)$.

1 **Initialize:** $lb = -\infty$, $ub = \infty$, $\widehat{\mathcal{Y}} = \emptyset$;

2 Solve (9) and let $\overline{\xi}^*$ be the optimal solution. Set $ub = \Psi_{\widehat{\mathcal{Y}}}(w)$;

3 Evaluate $\psi(w, \overline{\xi}^*)$ by solving problem (8) and let y^* the optimal solution. Set $lb = \psi(w, \overline{\xi}^*)$;

4 **if** *ub - lb > 0* **then** $\widehat{\mathcal{Y}} = \widehat{\mathcal{Y}} \cup \{y^*\}$ and go to Step 2 ;

5 **else return** $\Psi_{\widehat{\mathcal{Y}}}(w)$;

THEOREM 2. *Algorithm 2 converges after a finite number of iterations.*

4. Improving the Performance of the K-adaptability Approximation

The two-stage robust optimization problem with decision-dependent information discovery is known to be NP-hard, thus exact algorithms are expected to struggle as the size of the problem instance increases. To gain tractability an alternative approach is to approximate problem (1). First introduced by Bertsimas and Caramanis (2010), K-adaptability is a popular approximation for the class of problems with binary recourse decisions. The approximation reduces the complexity by approximating the recourse decisions y with K constant decisions and treating them as first-stage (here-and-now) decisions. After uncertainty materializes (or partly materializes in the decision-dependent discovery setting) the best amongst the K constant decisions is used so as to minimize the worst-case cost. For $w = e$, i.e., the decision-independent information discovery, Hanasusanto et al. (2015) and Subramanyam et al. (2019) provide efficient solution techniques based on a monolith mixed integer linear formulation and an iterative solution

scheme involving disjunctive programming, respectively. For the decision dependent information discovery case, Vayanos et al. (2020a) also provides a monolithic formulation which is similar in spirit to the formulation provided in Hanasusanto et al. (2015). In this section, we improve upon the formulation proposed in Vayanos et al. (2020a) by strengthening the formulation thus providing significantly improvements in terms of scalability and computational time. Our results can also be applied in the decision-independent case which is subsumed by the decision depended case. In the following, we first briefly present the K-adaptability formulation and then we discuss additional valid inequalities that strengthen the formulation.

The K-adaptability approximation of problem (1a), as introduced in Vayanos et al. (2020a), is stated as follows:

$$\min \max_{\overline{\boldsymbol{\xi}} \in \Xi} \min_{k \in \mathcal{K}} \left\{ \max_{\xi \in \Xi(\boldsymbol{w}, \overline{\boldsymbol{\xi}})} \boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^\top \boldsymbol{P} \boldsymbol{y}^k \right\}$$

$$\text{s.t.} \quad \boldsymbol{w} \in \mathcal{W}, \boldsymbol{y}^k \in \mathcal{Y}, \ \forall k \in \mathcal{K},$$

where $\mathcal{K} = \{1, 2, \ldots, K\}$. The here-and-now variables $\boldsymbol{y}^k, k \in \mathcal{K}$ encode the K constant decisions that define the recourse policy. In view of vector $\overline{\boldsymbol{\xi}}$ the inner minimization calls for selecting the best piece among the K constant decisions. The higher the value of K, the better the K-adaptability approximation, while setting $K = |\mathcal{Y}|$ ensures the optimal solution of (1) is recovered. In practice, it turns out that for some problem relatively small values of K, can provide good quality solutions. If we contrast K-adaptability and the exact algorithm presented in Section 3, K-adaptability only uses K constant decisions while Algorithm 2 dynamically increases the number of constant decisions in $\widehat{\mathcal{Y}}$ until a set that guarantees optimality is achieved. The K-adaptability counterpart can be reformulated as the following mixed integer bilinear programming problem (Vayanos et al. 2020a, Theorem 2)

$$\min \quad \boldsymbol{b}^\top \left(\boldsymbol{\beta} + \sum_{k \in \mathcal{K}} \boldsymbol{\beta}^k \right) \tag{10a}$$

$$\text{s.t.} \quad \boldsymbol{w} \in \mathcal{W}, \ \boldsymbol{y}^k \in \mathcal{Y}, k \in \mathcal{K} \tag{10b}$$

$$\boldsymbol{\alpha} \in \mathbb{R}_+^K, \ \boldsymbol{\beta} \in \mathbb{R}_+^L, \ \boldsymbol{\beta}^k \in \mathbb{R}_+^L, \ \boldsymbol{\gamma}^k \in \mathbb{R}^{N_w}, \ k \in \mathcal{K} \tag{10c}$$

$$\boldsymbol{e}^\top \boldsymbol{\alpha} = 1 \tag{10d}$$

$$\boldsymbol{A}^\top \boldsymbol{\beta} = \sum_{k \in \mathcal{K}} \boldsymbol{\gamma}^k \circ \boldsymbol{w} \tag{10e}$$

$$\boldsymbol{A}^\top \boldsymbol{\beta}^k + \boldsymbol{\gamma}^k \circ \boldsymbol{w} = \alpha_k \left(\boldsymbol{C} \boldsymbol{w} + \boldsymbol{P} \boldsymbol{y}^k \right) \qquad \forall k \in \mathcal{K}. \tag{10f}$$

As shown in Vayanos et al. (2020a), the bilinear terms $\boldsymbol{\gamma}^k \circ \boldsymbol{w}$, $\boldsymbol{\alpha}_k \boldsymbol{y}^k$ and $\boldsymbol{\alpha}_k \boldsymbol{w}$, for each $k \in \mathcal{K}$, can be linearized using the well known McCormick inequalities (McCormick (1983)). Unfortunately, the resulting linearization is in general characterized by a weak linear relaxation. On top of that, K-adaptability reformulation presents large amount of symmetry since the K candidate policies can be permuted to achieve equivalent solutions. Consequently, the K-adaptability problem is already challenging to solve for $K \in \{3, 4\}$ (Hanasusanto et al. (2015)). To overcome these issues, we propose several valid inequalities to both strengthen the formulation and decrease the degree of symmetry. In Section 4.1 we propose a new family of inequalities to reduce the symmetry of the problem and in Section 4.2, we show how we can strengthen the McCormick inequalities by restricting the domain of variables involved in the linearization. Furthermore, in Section 4.3, we propose the *optimistic inequalities* defined on optimistic realizations of the uncertainty vector $\boldsymbol{\xi}$. Finally, in Section 4.4 we discuss additional valid inequalities that can be added by exploiting the Reformulation-Linearization-Technique.

4.1. Addressing Symmetry

A drawback of the K-adaptability formulation is the symmetry with respect to the policies $\boldsymbol{y}^k$. Indeed, in problem (10) the index of variables $(\boldsymbol{\alpha}_k, \boldsymbol{\beta}^k, \boldsymbol{\gamma}^k, \boldsymbol{y}^k)$ can be permuted without changing the optimal value of the problem. To reduce symmetry in the K-adaptability reformulation, we need to ensure that permutating values of variables associated to the K policies, does not lead to equivalent solutions. We enforce this condition by imposing that $\boldsymbol{\alpha}$ belongs to set $\mathcal{A}_S$ defined as follows

$$\mathcal{A}_S := \left\{ \boldsymbol{\alpha} \in \mathbb{R}_+^K : \boldsymbol{e}^\top \boldsymbol{\alpha} = 1, \, \boldsymbol{\alpha}_k \geq \boldsymbol{\alpha}_{k+1} \, \forall k \in \mathcal{K} \setminus \{K\} \right\}. \tag{11}$$

The following theorem ensures that the addition of constraints defining $\mathcal{A}_S$ does not result in a conservative approximation.

THEOREM 3. *At least one optimal solution of problem (10) satisfies $\boldsymbol{\alpha} \in \mathcal{A}_S$.*

In Vayanos et al. (2020a) a different family of symmetry breaking constraints is introduced which breaks symmetry by imposing that $\boldsymbol{y}^k$, $k \in \mathcal{K}$ satisfy a lexicographic decreasing order. In the next subsection, we will use the properties of set $\mathcal{A}_S$ to further strengthen the McCormick inequalities used to linearize the bilinear terms in problem (10). Unfortunately, the symmetry breaking constraints from Vayanos et al. (2020a) cannot be used together with the strengthened McCormick inequalities.

4.2. Strengthening McCormick Inequalities

A common approach to linearize products of binary and continuous variables is through the so-called Mc-Cormik inequalities (McCormick (1983)). For every product of two original variables, a new continuous variable and four constraints are introduced. Consider bilinearity $\boldsymbol{\alpha}_k \boldsymbol{y}^k$ in problem (10). If $\ell_{\boldsymbol{\alpha}_k}$ and $u_{\boldsymbol{\alpha}_k}$ are the lower and upper bounds for variables $\boldsymbol{\alpha}_k$ for each $k \in \mathcal{K}$, respectively, and if we set $\tilde{\boldsymbol{y}}_i^k = \boldsymbol{\alpha}_k \boldsymbol{y}_i^k$, for each $i \in \mathcal{N}_{\boldsymbol{y}}$, the corresponding McCormick inequalities state as follows

$$
\begin{aligned}
\tilde{\boldsymbol{y}}_i^k \geq \ell_{\boldsymbol{\alpha}_k} \boldsymbol{y}_i^k, \ \ \tilde{\boldsymbol{y}}_i^k \leq u_{\boldsymbol{\alpha}_k} \boldsymbol{y}_i^k & \quad\quad\quad\quad \forall k \in \mathcal{K}, \forall i \in \mathcal{N}_{\boldsymbol{y}} \\
\tilde{\boldsymbol{y}}_i^k \leq \ell_{\boldsymbol{\alpha}_k} \boldsymbol{y}_i^k + \boldsymbol{\alpha}_k - \ell_{\boldsymbol{\alpha}_k}, \ \ \tilde{\boldsymbol{y}}_i^k \geq u_{\boldsymbol{\alpha}_k} \boldsymbol{y}_i^k + \boldsymbol{\alpha}_k - u_{\boldsymbol{\alpha}_k} & \quad\quad \forall k \in \mathcal{K}, \forall i \in \mathcal{N}_{\boldsymbol{y}}.
\end{aligned}
\tag{12}
$$

Inequalities (12) are tight when at least one of the two variables involved takes value equal to its upper or lower bound, otherwise they represent, in general, a relaxation. However, since variables $\boldsymbol{y}^k$ are binary, the McCormick inequalities are tight at every feasible solution, for each valid choice of $\ell_{\boldsymbol{\alpha}_k}$ and $u_{\boldsymbol{\alpha}_k}$, $k \in \mathcal{K}$. Since $\boldsymbol{\alpha} \in \mathbb{R}_+^K$ and $\boldsymbol{e}^\top \boldsymbol{\alpha} = 1$, as in Hanasusanto et al. (2015), Vayanos et al. (2020a), $\ell_{\boldsymbol{\alpha}_k}$ and $u_{\boldsymbol{\alpha}_k}$, $k \in \mathcal{K}$ can be set equal to 0 and 1, respectively. Nevertheless, the resulting linear relaxation may be extremely weak, slowing down the convergence of branch-and-cut algorithms. This is shown in the following example in which we illustrate how at fractional solutions the McCormick inequalities do not significantly constrain variables $\tilde{\boldsymbol{y}}^k$, $k \in \mathcal{K}$.

EXAMPLE 2. Consider a 2-adaptability instance of problem (10) with one recourse decision $\boldsymbol{y} \in \{0,1\}$ at a fractional solution $\boldsymbol{y}^1 = \boldsymbol{y}^2 = 0.5$ and $\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_2 = 0.5$. Using the bounds $\boldsymbol{\alpha}_k \in [\ell_{\boldsymbol{\alpha}_k}, u_{\boldsymbol{\alpha}_k}] = [0,1]$, $k \in \mathcal{K} = \{1,2\}$ the McCormick inequalities (12) imply that $\tilde{\boldsymbol{y}}^1 \in [0, \boldsymbol{y}^1] = [0, 0.5]$ and $\tilde{\boldsymbol{y}}^2 \in [0, \boldsymbol{y}^2] = [0, 0.5]$. These bounds are somewhat trivial and provide a very loose approximation to constraints $\tilde{\boldsymbol{y}}^1 = \boldsymbol{\alpha}_1 \boldsymbol{y}^1 = 0.025$ and $\tilde{\boldsymbol{y}}^2 = \boldsymbol{\alpha}_2 \boldsymbol{y}^2 = 0.025$.

Example 2 motivates us to examine how to reduce the bounds on $\boldsymbol{\alpha}$ as these can significantly strengthen the McCormick inequalities and drastically impact the convergence rate of the branch-and-cut algorithm. To this end, we define set $\mathcal{A}_B$ which provides strengthened bounds on $\boldsymbol{\alpha}$.

$$
\mathcal{A}_B := \left\{ \boldsymbol{\alpha} \in \mathbb{R}_+^K : \ \boldsymbol{\alpha}_k \in [\ell_{\boldsymbol{\alpha}_k}, u_{\boldsymbol{\alpha}_k}] \ \forall k \in \mathcal{K}, \ \boldsymbol{e}^\top \boldsymbol{\alpha} = 1 \right\}
$$

where

$$
\ell_{\boldsymbol{\alpha}_k} = \begin{cases} \frac{1}{K}, & \text{if } k = 1 \\[2mm] 0, & \text{otherwise} \end{cases} \quad \forall k \in \mathcal{K}, \quad\quad u_{\boldsymbol{\alpha}_k} = \frac{1}{k} \quad \forall k \in \mathcal{K}.
$$

The following theorem ensures that the addition of constraints $\mathcal{A}_B$ does not result in a conservative approximation.

THEOREM 4. *At least one optimal solution of problem (10) satisfies* $\boldsymbol{\alpha} \in \mathcal{A}_B$.

EXAMPLE 2 (CONTINUED). We now consider $\boldsymbol{\alpha} \in \mathcal{A}_S \cap \mathcal{A}_B$, i.e., $\boldsymbol{\alpha}_1 \in [0.5, 1]$ and $\boldsymbol{\alpha}_2 \in [0, 0.5]$. We can immediately verify that inequalities (12) are tight, i.e., they ensure $\tilde{\boldsymbol{y}}^1 = \boldsymbol{\alpha}_1 \boldsymbol{y}^1 = 0.025$ and $\tilde{\boldsymbol{y}}^2 = \boldsymbol{\alpha}_2 \boldsymbol{y}^2 = 0.025$.

The McCormick inequalities can also be applied to the bilinear term $\boldsymbol{\gamma}^k \circ \boldsymbol{w}$ by introducing extra variables $\tilde{\boldsymbol{\gamma}}^k_{\,i} = \boldsymbol{\gamma}^k_i \boldsymbol{w}_i$ for each $k \in \mathcal{K}$ and $i \in \mathcal{N}_{\boldsymbol{w}}$. However, since the $\boldsymbol{\gamma}^k$ always appears in problem (10) multiplied by variables $\boldsymbol{w}$, one can linearize the bilinear terms by imposing the following constraint

$$\boldsymbol{\gamma}^k \leq M\boldsymbol{w}, \ \boldsymbol{\gamma}^k \geq -M\boldsymbol{w} \quad \forall k \in \mathcal{K},$$

where M is a sufficiently large constant and replacing $\boldsymbol{\gamma}^k \circ \boldsymbol{w}$ with $\boldsymbol{\gamma}^k$, for each $k \in \mathcal{K}$ in constraints (10e)-(10f).

The linearized counterpart of problem (10) can be expressed as

$$
\begin{aligned}
\min \quad & \boldsymbol{b}^\top \left(\boldsymbol{\beta} + \sum_{k \in \mathcal{K}} \boldsymbol{\beta}^k \right) \\
\text{s.t.} \quad & \boldsymbol{w} \in \mathcal{W}, \ \tilde{\boldsymbol{y}}^k \in \mathcal{Y}, \ \tilde{\boldsymbol{w}}^k \in \left[0, \frac{1}{k}\right]^{N_{\boldsymbol{w}}}, \tilde{\boldsymbol{y}}^k \in \left[0, \frac{1}{k}\right]^{N_{\boldsymbol{y}}}, k \in \mathcal{K} \\
& \boldsymbol{\alpha} \in \mathcal{A}_S \cap \mathcal{A}_B, \ \boldsymbol{\beta} \in \mathbb{R}^L_+, \ \boldsymbol{\beta}^k \in \mathbb{R}^L_+, \ \boldsymbol{\gamma}^k \in \mathbb{R}^{N_{\boldsymbol{w}}}, \ k \in \mathcal{K} \\
& \boldsymbol{A}^\top \boldsymbol{\beta} = \sum_{k \in \mathcal{K}} \boldsymbol{\gamma}^k \\
& \boldsymbol{A}^\top \boldsymbol{\beta}^k + \boldsymbol{\gamma}^k = \boldsymbol{C}\tilde{\boldsymbol{w}}^k + \boldsymbol{P}\tilde{\boldsymbol{y}}^k \\
& \boldsymbol{\gamma}^k \leq M\boldsymbol{w}, \ \boldsymbol{\gamma}^k \geq -M\boldsymbol{w} \\
& \tilde{\boldsymbol{y}}^k \geq \frac{\mathbb{I}[k=1]\boldsymbol{y}^k}{K}, \ \tilde{\boldsymbol{y}}^k \geq \frac{\boldsymbol{y}^k}{k} + \left(\boldsymbol{\alpha}_k - \frac{1}{k}\right)\boldsymbol{e} \\
& \tilde{\boldsymbol{y}}^k \leq \frac{\mathbb{I}[k=1]\boldsymbol{y}^k}{K} + \left(\boldsymbol{\alpha}_k - \frac{\mathbb{I}[k=1]}{K}\right)\boldsymbol{e}, \ \tilde{\boldsymbol{y}}^k \leq \frac{\boldsymbol{y}^k}{k} \\
& \tilde{\boldsymbol{w}}^k \geq \frac{\mathbb{I}[k=1]\boldsymbol{w}}{K}, \ \tilde{\boldsymbol{w}}^k \geq \frac{\boldsymbol{w}}{k} + \left(\boldsymbol{\alpha}_k - \frac{1}{k}\right)\boldsymbol{e} \\
& \tilde{\boldsymbol{w}}^k \leq \frac{\mathbb{I}[k=1]\boldsymbol{w}}{K} + \left(\boldsymbol{\alpha}_k - \frac{\mathbb{I}[k=1]}{K}\right)\boldsymbol{e}, \ \tilde{\boldsymbol{w}}^k \leq \frac{\boldsymbol{w}}{k}
\end{aligned}
\right\} \ \forall\, k \in \mathcal{K}. \qquad (13)
$$

For the remainder of the paper we will refer to problem (13) as the strengthened K-adaptability problem.

4.3. Optimistic Inequalities

In this section, we introduce a family of valid inequalities based on optimistic realizations of the uncertainty parameters. These so called optimistic inequalities leverage the following observation, which sheds light on the relationship between $\boldsymbol{\alpha}$ and $\left\{\boldsymbol{y}^k\right\}_{k\in\mathcal{K}}$ in problem (10).

OBSERVATION 1   *For fixed $(\boldsymbol{w}, \{\boldsymbol{y}^k\}_{k\in\mathcal{K}})$ we have*

$$\max_{\bar{\boldsymbol{\xi}}\in\Xi} \min_{k\in\mathcal{K}} \max_{\boldsymbol{\xi}\in\Xi(\boldsymbol{w},\bar{\boldsymbol{\xi}})} \boldsymbol{\xi}^\top\boldsymbol{C}\boldsymbol{w} + \boldsymbol{\xi}^\top\boldsymbol{P}\boldsymbol{y}^k$$

$$= \begin{bmatrix} \min\ \tau \\ s.t. \quad \tau\in\mathbb{R},\ \bar{\boldsymbol{\xi}}\in\Xi,\ \boldsymbol{\xi}^k\in\Xi(\boldsymbol{w},\bar{\boldsymbol{\xi}})\ \forall k\in\mathcal{K} \\ \qquad \tau \le \boldsymbol{\xi}^\top\boldsymbol{C}\boldsymbol{w} + \boldsymbol{\xi}^\top\boldsymbol{P}\boldsymbol{y}^k \qquad \forall k\in\mathcal{K} \end{bmatrix} \tag{14}$$

where the equality follows from (Vayanos et al. 2020a, Lemma 1). Dualizing (14) and combining with the minimization over $(\boldsymbol{w}, \{\boldsymbol{y}^k\}_{k\in\mathcal{K}})$ results in problem (10). For each $k\in\mathcal{K}$, variable $\boldsymbol{\alpha}_k$ in problem (10) correspond to the dual multipliers associated with constraint $\tau \le \boldsymbol{\xi}^\top\boldsymbol{C}\boldsymbol{w} + \boldsymbol{\xi}^\top\boldsymbol{P}\boldsymbol{y}^k$. At optimality $\tau = \boldsymbol{\xi}^\top\boldsymbol{C}\boldsymbol{w} + \boldsymbol{\xi}^\top\boldsymbol{P}\boldsymbol{y}^k$ for some $k\in\mathcal{K}$, with $\boldsymbol{\alpha}_k > 0$ indicating that the constraint is binding hence $\boldsymbol{y}^k$ influences the optimal value of problem (10), while $\boldsymbol{\alpha}_k = 0$ indicates the contrary. This is also evident in problem (10) where if $\boldsymbol{\alpha}_k = 0$ for some $k\in\mathcal{K}$, then $\boldsymbol{y}^k$ does not influence the optimal value since term $\boldsymbol{\alpha}_k\boldsymbol{y}^k$ in constraint (10f) is responsible for the interaction of $\boldsymbol{y}^k$ with the rest of the variables. Hence for $k\in\mathcal{K}$ with $\boldsymbol{\alpha}_k = 0$ setting $\boldsymbol{y}^k := \boldsymbol{y}'^k$ to any $\boldsymbol{y}'^k \in \mathcal{Y}$, then $(\boldsymbol{w}, \{\boldsymbol{y}_k\}_{k\in\mathcal{K}})$ achieves the same optimal value in (10).

We now propose the following inequalities which are based on an optimistic view of uncertainty. Let $\boldsymbol{\zeta}^{\boldsymbol{w}} \in \mathbb{R}^{N_{\boldsymbol{w}}}$, $\boldsymbol{\zeta}^{\boldsymbol{y}} \in \mathbb{R}^{N_{\boldsymbol{y}}}$ be two vectors such that $\zeta^{\boldsymbol{w}}_i \le \boldsymbol{\xi}^\top\boldsymbol{C}_i$ for $i\in\mathcal{N}_{\boldsymbol{w}}$ and $\zeta^{\boldsymbol{y}}_j \le \boldsymbol{\xi}^\top\boldsymbol{P}_j$ for $j\in\mathcal{N}_{\boldsymbol{y}}$ for all $\boldsymbol{\xi}\in\Xi$, where $\boldsymbol{C}_i$ and $\boldsymbol{P}_j$ represent the i-th and j-th columns of matrices $\boldsymbol{C}$ and $\boldsymbol{P}$, respectively. Then the following family of inequalities

$$\boldsymbol{b}^\top\left(\boldsymbol{\beta} + \sum_{k\in\mathcal{K}}\boldsymbol{\beta}^k\right) \ge \boldsymbol{\zeta}^{\boldsymbol{w}\top}\boldsymbol{w} + \boldsymbol{\zeta}^{\boldsymbol{y}\top}\boldsymbol{y}^k \quad \forall k\in\mathcal{K}, \tag{15}$$

can be added to problem (13) without loss of optimality, as stated in the following theorem.

THEOREM 5.   *At least one optimal solution of problem (13) satisfies inequalities (15).*

In general, there may be many possible vectors $\boldsymbol{\zeta}^{\boldsymbol{w}}$, $\boldsymbol{\zeta}^{\boldsymbol{y}}$, however, the tightest vectors for which inequalities are still valid can be generated by solving $N_{\boldsymbol{w}} + N_{\boldsymbol{y}}$ linear optimization problems

$$\boldsymbol{\zeta}_i^{\boldsymbol{w}} = \min_{\boldsymbol{\xi} \in \Xi} \boldsymbol{\xi}^\top \boldsymbol{C}_i \quad \forall i \in \mathcal{N}^{\boldsymbol{w}}, \qquad \boldsymbol{\zeta}_j^{\boldsymbol{y}} = \min_{\boldsymbol{\xi} \in \Xi} \boldsymbol{\xi}^\top \boldsymbol{P}_j \quad \forall j \in \mathcal{N}^{\boldsymbol{y}}$$

The way vectors $\boldsymbol{\zeta}^{\boldsymbol{w}}$ and $\boldsymbol{\zeta}^{\boldsymbol{y}}$ are generated suggests that the effectiveness of (15) strongly depends on the considered uncertainty set. Note that a similar inequality can be added to problem (1a). In this case, the inequality only involves $\boldsymbol{w}$ and take the form $\Phi(\boldsymbol{w}) \geq \boldsymbol{\zeta}^{\boldsymbol{w}^\top} \boldsymbol{w}$.

4.4. Reformulation-Linearization-Technique Inequalities

The *Reformulation-Linearization-Technique* (RLT), introduced in Sherali and Alameddine (1992), is a procedure used to derive tight relaxations for discrete and continuous non-convex problems. It consists of two phases: reformulation and linearization. In the reformulation phase, valid quadratic constraints are generated using pairwise product operations between constraints and/or variables. In the linearization phase, the products resulting from the reformulation phase are linearized through additional variables and McCormick inequalities.

We apply the RLT scheme to problem (13) to obtain tighter linear relaxations that can assist the convergence of the branch-and-cut algorithm. In our implementation, the reformulation phase consists in generating valid inequalities by multiplying each constraints in the definition of $\mathcal{Y}$ with $\boldsymbol{\alpha}_k$ for each $k \in \mathcal{K}$. Notice that if some constraints in the definition of $\mathcal{Y}$ are added on the fly, additional RLT inequalities can be generated within the branch-and-cut procedure. This is the case for the orienteering problem whose constraints are described in (2). In this case the exponential number of subtours elimination constraints are dynamically added when needed together with the corresponding RLT inequalities.

5. Computational Experiments

In this section, we present experiments to assess the performance of the exact algorithm presented in Section 3 using the information inequalities (7), and the strengthened K-adaptability problem (13) in conjunction with optimistic inequalities and the RLT inequalities presented in Section 4. Section 5.1 compares the performance for the two solution schemes on the sensor placement orienteering problem. As the strengthened K-adaptability formulation can also be applied to problem instances without decision dependent information discovery, in Section 5.2 we benchmark its performance to the K-adaptability formulation proposed

by Hanasusanto et al. (2015) and the iterative solution scheme proposed by Subramanyam et al. (2019) using the shortest path problem, as both publications present results on this problem instance. All of our experiments are performed in single core on a machine equipped with a 4.0GHz Intel i7-600K processor and with 24GB of RAM, using CPLEX 12.10. A time of 7200 seconds is allowed to solve each instance.

5.1. Computational Experiments for the Sensor Placement Orienteering Problem

In the following we use the orienteering problem networks presented in Tsiligirides (1984) which consists of three network structures which we denote by TS1, TS2 and TS3, involving 30, 19 and 31 nodes, respectively. For all networks, the start and destination nodes are placed in the center of the of the network. We also construct smaller versions of the same networks where we remove half of the nodes closest to the start/destination node, resulting in networks with 15, 10 and 16 nodes, respectively. The original networks are denoted by TS1N30, TS2N19 and TS3N31, and their smaller variants by TS1N15, TS2N10 and TS1N16, respectively. Similar to Tsiligirides (1984), due to the different topology of each network, different values of the maximum time duration T are considered, and are summarized in Table EC.1. All instances can be found in Paradiso.

We consider the sensor placement orienteering problem (3) where $\mathcal{Y}$ is given in (2) and the uncertainty set is given by $\Xi = \{\boldsymbol{\xi} \in [0, U]^N : \boldsymbol{e}^\top \boldsymbol{\xi} = 1\}$ which expresses the view that the total profit in the network is known and equals to 1 (or 100%), however, the geographical breakdown of the profit in each node is uncertain. The upper bound U denotes the maximum profit of each node, and its value for each network is given in Table EC.1. To assess the impact of the number of sensors on the total profit collect, we denote by δ the fraction of the total nodes where sensors can be placed, and for each $\delta \in \{0.25, 0.5, 0.75\}$ the admissible sensor placements are described by $\mathcal{W} = \{\boldsymbol{w} \in \{0, 1\}^{N_{\boldsymbol{w}}} : \boldsymbol{e}^\top \boldsymbol{w} \leq \lceil \delta N_{\boldsymbol{w}} \rceil \}$. We next compare the relative performance of the exact method and the strengthened K-adaptability formulation (Section 5.1.1), and examine the behavior of the strengthened K-adaptability formulation (Section 5.1.2).

5.1.1. Comparing the Exact Algorithm and K-adaptability
In our first experiment, we compare the performance of the exact solution method and of the strengthened K-adaptability formulation for $K \in \{2, 3, 4\}$ in terms of their optimality gap. For each $\delta = \{0.25, 0.5, 0.75\}$ and network topology, we solve using both methods problem (3) for all duration times T listed in Table EC.1. Notice that the optimal value

of problem (6), which we denote by "best progressive bound", provides a progressive bound to the optimal value of problem (3). Hence, the optimality gap after 7200 seconds of computational time, is calculated as $\frac{\text{"best conservative bound" - "best progressive bound"}}{\text{"best progressive bound"}}$ where the "best conservative bound" is the best objective value achieved by each method within the time limit. Table 2 reports the average gap with respect to the different values of T achieved for each method. Tables 3 and 4 (next section) presents further details about the behavior of exact method and strengthened K-adaptability, respectively.

Table 2   Optimality gap for the exact solution method and the K-adaptability approximation.

Network	δ	2-Adapt	3-Adapt	4-Adapt	Exact	Network	δ	2-Adapt	3-Adapt	4-Adapt	Exact
	0.25	8.9%	7.1%	7.1%	0.0%		0.25	10.0%	8.0%	11.2%	9.5%
TS1N15	0.50	16.2%	8.8%	7.4%	0.0%	TS1N30	0.50	8.6%	4.5%	4.8%	3.2%
	0.75	17.7%	10.2%	7.6%	0.0%		0.75	8.6%	3.7%	2.8%	0.3%
	0.25	11.1%	0.0%	0.0%	0.0%		0.25	8.6%	7.6%	7.6%	7.5%
TS2N10	0.50	16.8%	1.8%	0.0%	0.0%	TS2N19	0.50	21.0%	8.3%	2.4%	0.6%
	0.75	17.0%	4.4%	0.2%	0.0%		0.75	22.0%	10.5%	5.1%	0.0%
	0.25	7.1%	2.4%	0.0%	0.0%		0.25	9.1%	7.8%	11.2%	10.8%
TS3N16	0.50	12.9%	5.4%	2.9%	0.0%	TS3N31	0.50	8.2%	4.4%	3.5%	3.3%
	0.75	13.9%	6.4%	3.9%	0.0%		0.75	8.2%	3.8%	2.0%	0.8%

Table 3   Additional details on the performance of the exact algorithm. We denote by *Opt (#)* the number of instances solved to optimality, by *Time (s)* the average computational time (in seconds) for instances solved to optimality and by *Gap* the average optimality gap for the instances not solved within the time limit. Overall, 182 instances out of 258 are solved to optimality with an average computational time of 411 seconds. The average gap for the 76 instances not solved to optimality is equal to 7.94%.

Network	δ	Opt (#)	Time (s)	Gap	Network	δ	Opt (#)	Time (s)	Gap
	0.25	14/14	80.4	-		0.25	6/18	71.2	14.0%
TS1N15	0.50	14/14	89.5	-	TS1N30	0.50	6/18	103.0	4.8%
	0.75	14/14	12.4	-		0.75	10/18	193.2	0.7%
	0.25	9/9	3.8	-		0.25	7/11	2364.4	17.1%
TS2N10	0.50	9/9	4.1	-	TS2N19	0.50	11/11	2985.1	-
	0.75	9/9	1.6	-		0.75	11/11	1155.9	-
	0.25	14/14	43.1	-		0.25	6/20	154.6	14.5%
TS3N16	0.50	14/14	156.7	-	TS3N31	0.50	7/20	321.2	6.6%
	0.75	14/14	15.4	-		0.75	7/20	120.6	1.2%

From our first experiment we observed the following. First notice that the exact method optimally solves the smaller instances TS1N15, TS2N10 and TS2N16 for all values of δ and T. Overall, it solves to optimality 70% of the total instances. Interestingly, for the larger more challenging instances, we observe that the exact method is more efficient in finding near optimal solutions as δ increases. The strengthened K-adaptabililty approximation is solved to optimality for the majority of the smaller instances as well, while across all values of $K \in \{2, 3, 4\}$ around 65% are solved to optimality, see Table 4. Hence, for the smaller instances, the gap endured from the K-adaptability approximation results primarily from the lack of adaptivity from the recourse decisions $\boldsymbol{y}$, which gets worst as the δ increases. Still, as expected, the gap improves as the level of adaptability K increases and in some cases K-adaptability achieves the optimal solution of problem (3), e.g., for network TS2N10 and $\delta = 0.25$, $K = 3$ achieves the optimal solution for all values of T. We conclude the following: (i) the exact method provides on average better quality solutions compared to K-adaptability, however, in two cases (i.e., TS1N30 and TS3N31 with $\delta = 0.25$) K-adaptability can provide better solutions with small values of K regardless if the optimization problems are not solved to optimality, (ii) if adaptability is limited, the extra information provided by increasing the number of sensors, i.e., increasing δ, will not be able to be exploited by K-adaptability. We will see further evidence of this in the following and in Section 6.

Note that our results do not report 1-adaptability as this case is not interesting. This is the case as there is essentially no adaptability, thus increasing δ will not improve the optimal solution. In fact all sensor placements will achieve the same optimal value as the case where no sensors are used. Similar to Example 1, due to the structure of uncertainty set Ξ, the worst-case $\boldsymbol{\xi}$ will move the profit away from the visited nodes. Hence, for small values of T, 1-adaptability will achieve zero profit regardless of which nodes are visited, while for larger values of T it will achieve the minimum profit allowable by the uncertainty set. In terms of computational time, 1-adaptability can be optimally solved within a few seconds.

Next we discuss the sensor placement positions resulting from K-adaptability for $K \in \{3, 4\}$ and the exact solution method. We consider the network TS3N16 with $\delta = 0.5$ and time duration $T = 20$, i.e., the network has 16 nodes and at most 8 sensors can be placed throughout the network. All problems are solved to optimality. Figure 4 presents the nodes of the network (white circles) and the optimal sensor positions for each method (black squares). The start/destination node is denoted by a black triangle. For

K-adaptability the graph also presents the resulting K constant decisions $\boldsymbol{y}$. Note that we do not report results for 2-adaptability as the optimal profit is zero. This implies that regardless of the sensor placement, due to not enough adaptability from the recourse decisions, nature is able to adversary place the worst-case profit in other nodes. In contrast, 3- and 4-adaptability produce positive worst-case profit which amounts to 3% and 5% of the total profit in the network, see Figures 4a and 4b, respectively. Similar to 2-adaptability, 3-adaptability is not able to fully utilize the information from all 8 sensors, as the optimal solution places only 7 out of the 8 sensors. In contrast, 4-adaptability places all 8 sensors and achieves a higher collected profit. The sensor placement for both 3- and 4-adaptability is significantly different to the sensor placement of the exact solution, Figure 4c, which places the sensors almost uniformly around the start/destination node, and is able to achieve a worst-case profit of 7.7%.
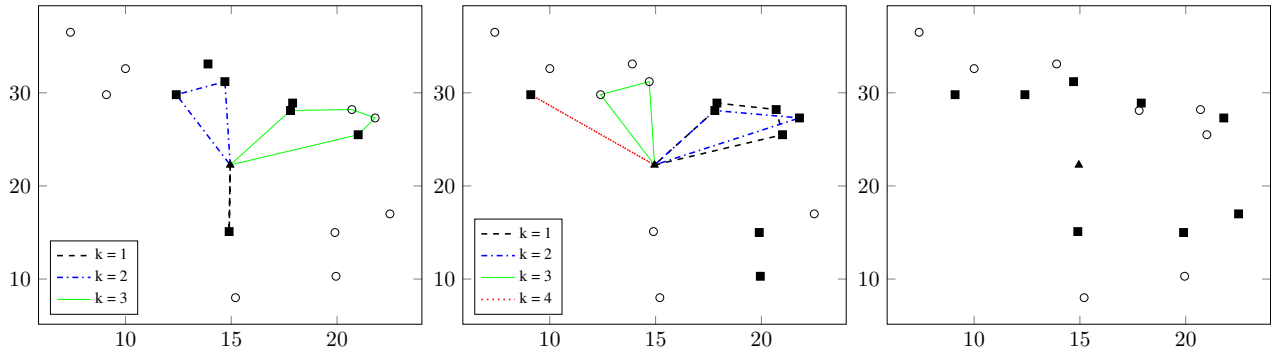
(a) 3-adaptability: worst-case profit 3%.   (b) 4-adaptability: worst-case profit 5%.   (c) Optimal: worst-case profit 7.7%.

Figure 4   Sensor placements for TS3N16 with $\delta = 0.50$ and $T = 20$ generated by K-adaptability ($K = 3, 4$) and the exact method. Nodes are plotted in cartesian coordinates using black squares (sensor placed) or circles (sensor not placed).

To better understand the difference in exact and K-adaptability approximation, and to distinguish if the suboptimality is due to the suboptimal placement of the sensors or the lack of adaptivity from the K-adaptable solution, we next evaluate $\Phi(\boldsymbol{w}^{*K})$ using Algorithm 2, where $\boldsymbol{w}^{*3}$ and $\boldsymbol{w}^{*4}$ are the optimal sensor placement resulting from 3- and 4-adaptability, respectively. We observe that $\Phi(\boldsymbol{w}^{*3})$ results in a 5% profit, and $\Phi(\boldsymbol{w}^{*4})$ results in in 6.67% profit, as opposed to the 3% and 5% indicated by the optimal value of problem (13), respectively. We conclude that indeed, part of the suboptimality K-adaptability is due to the lack of adaptability, and part is due suboptimal placement of the sensors.

5.1.2. Further Details on the Performance of the Strengthened K-adaptability Approximation

We next take a closed look at the behavior of the strengthened K-adaptability formulation and compare its performance to the K-adaptability formulation proposed in Vayanos et al. (2020a). Results are reported in Table 4. *Str K-adapt* denotes the strengthened formulation introduced in Section 4 and *K-adapt* refer to the formulation proposed in Vayanos et al. (2020a). The last column labeled *RN* reports the average root node relaxation improvement of the strengthened formulation with respect to original one (calculated as $\frac{\text{Root Node strengthened } K\text{-adapt - Root Node } K\text{-adapt}}{\text{Root Node strengthened } K\text{-adapt}}$). The rest of the table is interpreted as Table 3. We clarify that in Table 4, *Gap* refers to the optimality gap of the mixed-integer solver when the time limit is reached. All the average values are calculated with regards to the different values of the maximum time durations T. Overall, the strengthened formulation consistently outperforms the formulation of Vayanos et al. (2020a), solving to optimality 65% of all instances as opposed to 52% of the Vayanos et al. (2020a) formulation. Beside that, optimality gaps are consistently reduced when instances are not solved to optimality. The benefit of the improved bounds and valid inequalities can be also noticed by observing the bound at the root node relaxation of the two formulation. As shown in the last column, the improvements of the root node relaxation ranges between 45% and 80%.

In the majority of cases, the stronger lower bounds assist the branch-and-cut procedure to converge faster. However, there is a small number of exceptions, e.g., one instance of TS1N15 with $\delta = 0.25$ and $K = 2$, where the strengthened formulation has an adverse effects. Nevertheless, we can conclude that for the sensor placement orienteering problem, the strengthened K-adaptability formulation improves significantly the computational time.

5.2. Assessing the Strengthened K-adaptability Formulation: The Shortest Path Problem

The strengthened K-adaptability formulation is also valid for problem instances without decision dependent information discovery. To assess its relative merits compared to other approaches in the literature, we consider the robust shortest path problem solved in Hanasusanto et al. (2015) and in Subramanyam et al. (2019). Note that this problem is not characterized by decision dependent information discovery: uncertainty is always revealed before the selection of the recourse decisions. In our formulation, this is achieved by fixing $\boldsymbol{w} = \boldsymbol{e}$. As in Hanasusanto et al. (2015) and in Subramanyam et al. (2019) we consider a shortest path problem defined on a directed weighted graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N} := \{1, \dots N\}$ represents the set

Table 4   Summary of the results for the K-adaptability formulations.

Network	K	δ	Opt (#) Str K-Adapt	K-Adapt	Time (s) Str K-Adapt	K-Adapt	Gap Str K-Adapt	K-Adapt	RN
		0.25	13/14	14/14	324.3	114.8	7.1%	-	58.5%
	2	0.50	14/14	14/14	15.4	50.5	-	-	56.3%
		0.75	14/14	14/14	14.4	44.4	-	-	53.7%
		0.25	9/14	8/14	289.9	167.9	19.2%	24.1%	64.5%
	3	0.50	14/14	10/14	213.4	426.9	-	11.2%	64.5%
TS1N15		0.75	14/14	10/14	94.1	505.7	-	11.0%	64.8%
		0.25	5/14	6/14	246.0	462.4	21.4%	38.0%	76.1%
	4	0.50	9/14	7/14	552.4	316.0	4.8%	27.6%	76.2%
		0.75	11/14	6/14	1792.3	114.3	4.3%	23.6%	75.1%
		0.25	9/9	9/9	0.9	0.3	-	-	42.6%
	2	0.50	9/9	9/9	0.5	0.2	-	-	43.1%
		0.75	9/9	9/9	0.6	0.2	-	-	42.3%
		0.25	9/9	9/9	30.3	6.1	-	-	53.5%
	3	0.50	9/9	9/9	3.6	3.8	-	-	52.9%
TS2N10		0.75	9/9	9/9	4.6	3.7	-	-	53.4%
		0.25	7/9	9/9	357.8	789.8	7.7%	-	59.2%
	4	0.50	9/9	9/9	33.7	131.4	-	-	63.3%
		0.75	9/9	9/9	43.2	73.2	-	-	62.4%
		0.25	14/14	14/14	21.5	8.8	-	-	45.9%
	2	0.50	14/14	14/14	6.3	9.5	-	-	46.9%
		0.75	14/14	14/14	6.2	9.7	-	-	46.3%
		0.25	13/14	11/14	392.2	715.9	12.0%	12.4%	61.1%
	3	0.50	14/14	12/14	274.8	1164.3	-	10.2%	60.9%
TS3N16		0.75	14/14	13/14	197.5	592.4	-	13.8%	60.9%
		0.25	7/14	4/14	312.5	402.4	16.1%	22.4%	67.1%
	4	0.50	10/14	6/14	1068.2	1460.0	7.1%	20.3%	66.5%
		0.75	11/14	6/14	968.6	552.6	6.8%	16.7%	68.3%
		0.25	12/18	6/18	675.1	95.6	22.2%	21.5%	49.4%
	2	0.50	15/18	6/18	415.7	101.7	2.0%	17.3%	49.2%
		0.75	16/18	8/18	404.3	516.7	1.7%	17.8%	49.4%
		0.25	6/18	4/18	1232.2	168.1	13.5%	36.3%	71.3%
	3	0.50	5/18	4/18	1275.7	455.9	5.9%	36.5%	70.2%
TS1N30		0.75	6/18	4/18	665.9	411.3	4.6%	35.3%	71.2%
		0.25	4/18	3/18	481.3	0.1	20.3%	43.2%	77.5%
	4	0.50	3/18	3/18	0.1	0.5	11.5%	41.4%	80.4%
		0.75	3/18	3/18	0.2	0.3	10.0%	42.0%	79.9%
		0.25	9/11	11/11	219.6	391.7	3.7%	-	49.6%
	2	0.50	11/11	11/11	162.1	179.2	-	-	50.1%
		0.75	11/11	11/11	151.1	324.8	-	-	49.4%
		0.25	1/11	3/11	4222.3	1024.4	21.3%	35.3%	60.8%
	3	0.50	8/11	6/11	649.4	666.1	2.7%	13.4%	60.8%
TS2N19		0.75	8/11	7/11	998.5	792.2	1.4%	16.1%	60.8%
		0.25	1/11	1/11	0.3	0.0	32.9%	42.4%	65.9%
	4	0.50	2/11	2/11	1895.1	273.8	5.9%	28.0%	65.5%
		0.75	3/11	2/11	3115.4	614.6	2.1%	24.1%	66.7%
		0.25	15/20	7/20	547.4	38.4	2.9%	21.8%	48.0%
	2	0.50	18/20	8/20	338.6	209.5	4.4%	18.9%	48.1%
		0.75	18/20	8/20	487.0	60.8	2.5%	19.2%	48.5%
		0.25	4/20	4/20	2000.7	34.5	10.5%	34.3%	67.3%
	3	0.50	10/20	4/20	2048.8	79.8	7.2 %	32.6%	67.1%
TS3N31		0.75	6/20	4/20	1513.3	37.8	4.0 %	32.5%	68.6%
		0.25	5/20	3/20	1197.6	0.2	20.8%	39.9%	74.2%
	4	0.50	4/20	4/20	328.3	438.1	9.9%	38.1%	74.0%
		0.75	5/20	4/20	927.0	393.2	8.4%	37.3%	73.1%
Total			502/774	405/774					
Average					359.4	275.4	11.5%	30.0%	61.8%

of the N nodes, $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of arcs and $\boldsymbol{c}_{ij}(\boldsymbol{\xi}) = (1 + \boldsymbol{\xi}_{ij}/2)\tilde{\boldsymbol{c}}_{ij}$ is the cost associated to arc (i,j), function of the Euclidean distance $\tilde{c}_{ij}$ and of the uncertain vector $\boldsymbol{\xi}$ which belongs to the uncertainty set $\Xi \subseteq \mathbb{R}^{|\mathcal{A}|}$. The objective is to determine K paths starting from a start node $s \in \mathcal{N}$ and terminating at a terminal node $t \in \mathcal{N}$, $t \neq s$ before observing $\boldsymbol{\xi}$, such that the worst-case length of the best path among the K defined is minimized. The problem can be formulated as follows.

$$\min \quad \max_{\boldsymbol{\xi} \in \Xi} \min_{k \in \mathcal{K}} \sum_{(i,j) \in A} \boldsymbol{c}_{ij}(\boldsymbol{\xi}) \boldsymbol{y}_{ij}^{k}$$

$$\text{s.t.} \quad \boldsymbol{y}_{ij}^{k} \in \{0,1\}, \ \forall (i,j) \in A, \forall k \in \mathcal{K}$$

$$\sum_{(j,l) \in A} \boldsymbol{y}_{jl}^{k} \geq \sum_{(i,j) \in A} \boldsymbol{y}_{ij}^{k} + \mathbb{I}[j=s] - \mathbb{I}[j=t] \quad \forall j \in V, \ \forall k \in \mathcal{K}$$

We consider instances with $N \in \{30, 40, 50\}$ using the same generation procedure as in Hanasusanto et al. (2015) and in Subramanyam et al. (2019). In each problem, the location of each node is chosen uniformly at random from the interval $[0,10]^2$. Start and terminal nodes are chosen as the pair of nodes with largest Euclidean distance. The set of arc $\mathcal{A}$ is generated by removing from the set $\mathcal{N} \times \mathcal{N}$ the $\lfloor 0.7|\mathcal{N} \times \mathcal{N}| \rfloor$ arcs with the largest nominal weight. A budget uncertainty set $\Xi = \{\boldsymbol{\xi} \in [0,1]^A : \sum_{(i,j) \in A} \boldsymbol{\xi}_{ij} \leq B\}$ set is considered. For each graph size, 10 problems are generated and solved for B equal to 3 and 6.

Results are reported in Table 5 and the interpretation of its columns is the same as in Tables 3 and 4. The results clearly demonstrate the merits of the strengthened formulation which provides significant computational benefits compared to the existing methods.

6. Case Study: Collecting Medicine Crates at the Alrijne Hospital

The Alrijne hospital in the Netherlands is a general hospital with more than 3700 staff and has a capacity of 500 beds across several departments, treating yearly more than 190,000 patients. In this study we consider the hospital in Leiderdorp which is a single building with six floors. The medications are supplied by a pharmacy operating from a depot at the basement of the hospital. To ensure the quality of the medicines, the deliveries are done in specialized crates equipped with sensors that record temperature, light and shock variations to make sure that medication is not damaged and that the crates are not opened by unauthorized personnel. The crates are expensive equipment that need to be recovered to be reused in future deliveries.

The pharmacy is facing the problem that although the crates are delivered without problems to their destinations, their collection process is highly problematic as a large number of these crates are lost. Consequently, to achieve timely delivery of the medicine, the pharmacy is forced to acquire more of these

Table 5   Computational results for the shortest path problem

	K	N	Hanasusanto et al. (2015)			Subramanyam et al. (2019)			Str K-Adapt		
			Opt (#)	Time (s)	Gap	Opt (#)	Time (s)	Gap	Opt (#)	Time (s)	Gap
$B=3$	2	30	0/10	-	49.2%	7/10	852.0	6.7%	10/10	2.7	-
		40	0/10	-	63.7%	1/10	1034.0	7.0%	10/10	20.0	-
		50	0/10	-	69.5%	0/10	-	9.3%	10/10	158.1	-
	3	30	0/10	-	58.8%	2/10	987.2	3.8%	10/10	8.4	-
		40	0/10	-	70.0%	0/10	-	6.5%	10/10	60.3	-
		50	0/10	-	74.8%	0/10	-	8.4%	10/10	1207.0	-
	4	30	0/10	-	70.4%	2/10	1875.7	4.7%	10/10	51.0	-
		40	0/10	-	76.1%	0/10	-	6.1%	10/10	232.7	-
		50	0/10	-	82.2%	0/10	-	7.3%	10/10	2451.4	-
$B=6$	2	30	0/10	-	50.4%	0/10	-	10.8%	10/10	27.5%	-
		40	0/10	-	67.1%	0/10	-	15.4%	10/10	321.6	-
		50	0/10	-	68.4%	0/10	-	17.2%	9/10	1915.2	0.4%
	3	30	0/10	-	60.0%	0/10	-	10.8%	10/10	113.9	-
		40	0/10	-	72.7%	0/10	-	14.1%	9/10	939.2	0.6%
		50	0/10	-	77.0%	0/10	-	15.4%	4/10	2714.0	3.4%
	4	30	0/10	-	70.7%	0/10	-	9.5%	10/10	529.3	-
		40	0/10	-	79.2%	0/10	-	12.3%	6/10	2324.9	1.3%
		50	0/10	-	86.4%	0/10	-	13.3%	3/10	2528.5	3.6%

specialized crates, thus increasing significantly their operating costs. In an attempt to understand this issue, in collaboration with the Alrijne hospital, we conducted a pilot study from February 11 to April 30 in 2019 (Joosten (2019), Veen (2020)). The primary purpose of the study was to understand where the crates ended up. To this end, transmitters were retrofitted in the crates and their location was tracked by a network of sensors placed in a part of the hospital. The study disputed the original belief that the crates where stolen and moved out of the hospital, and in fact showed that the crates were ending up in different departments within the hospital.

In this follow up study, we integrate the placing of sensors with the crate collection process. Following the guidelines set by the hospital, the aim is to be able to efficiently collect the crates by placing a limited number of sensors throughout the hospital, thus limiting the infrastructural as well as the amount of sensory data that need to be collected, stored and analyzed. The aim is to understand the trade-off between the number of sensors, the flexibility of adapting the collection routes and the maximum available duration of the collection routes.

6.1. Data Description

We model the problem as a sensor placement orienteering problem (3). The hospital is represented via a graph where the 27 departments of the hospital are clustered into 14 nodes that cover the 6 floors of the building. The clustering was based on the floor plan of the hospital and ensures that all departments within each node have short walking distances, see Table EC.2 and Figure 5. The associated travel time $t_{(i,j)}$ between nodes i and j is calculated as the estimated shortest path walking time plus the time need to collect the crates in node j. If two nodes are located in different floors of the hospital, the travel time is calculated as the walking time from the departing node to the elevator, the time spent in the elevator and the walking time from the elevator to the destination node plus the time to collect the crates. The resulting distance matrix is available at Paradiso. We model by $\boldsymbol{\xi}_i$ as the uncertain percentage of crates located in node i. As before, the uncertainty set is given by $\Xi_1 = \{\boldsymbol{\xi} \in [0,U]^{14} : \boldsymbol{e}^\top \boldsymbol{\xi} = 1\}$ with $U = 0.2$ and reflects that the total number of crates circulating in the hospital is known through constraints $\boldsymbol{e}^\top \boldsymbol{\xi} = 1$, however, their geographical breakdown is uncertainty, with at most 20% of the total crates can be found at any node of the network.

6.2. Adaptability and Value of Information

In our first experiment, we assess the impact of the number of sensors placed in the network on the percentage of crates collected, using the exact algorithm and K-adaptability with $K \in \{2,3,4\}$. As before, we denote use $\mathcal{W} = \{\boldsymbol{w} \in \{0,1\}^{N_w} : \boldsymbol{e}^\top \boldsymbol{w} \le \lceil \delta N_w \rceil\}$ and examine the impact of the number of sensors by changing $\delta \in \{0,\ 0.25,\ 0.50,\ 0.75,\ 1\}$. We set the maximum time duration to two hours, i.e., $T = 2$ hours. All problem instances were solved to optimality. Figure 6 reports the optimal value achieved by both methods as a function of δ. As expected, the exact algorithm achieves higher worst-case profit, and is able to better utilize the information from the increased number of sensors. We again observe, that K-adaptability can only utilize a proportion of the information provided. Indeed, 2-adaptability approximation seems extremely restrictive and achieves the same solution as $\delta = 0$, while 3-adaptability and 4-adaptability can effectively utilize the information from 25% and 50% of the sensors, respectively. Despite this shortcoming, a salient feature of K-adaptability is that the optimal solution provides K alternative routing plans from which one can be selected once measurements from the sensors has been obtained. This could have a practical interest if the personnel needs to train for standardized routing. However, due to the relatively small size
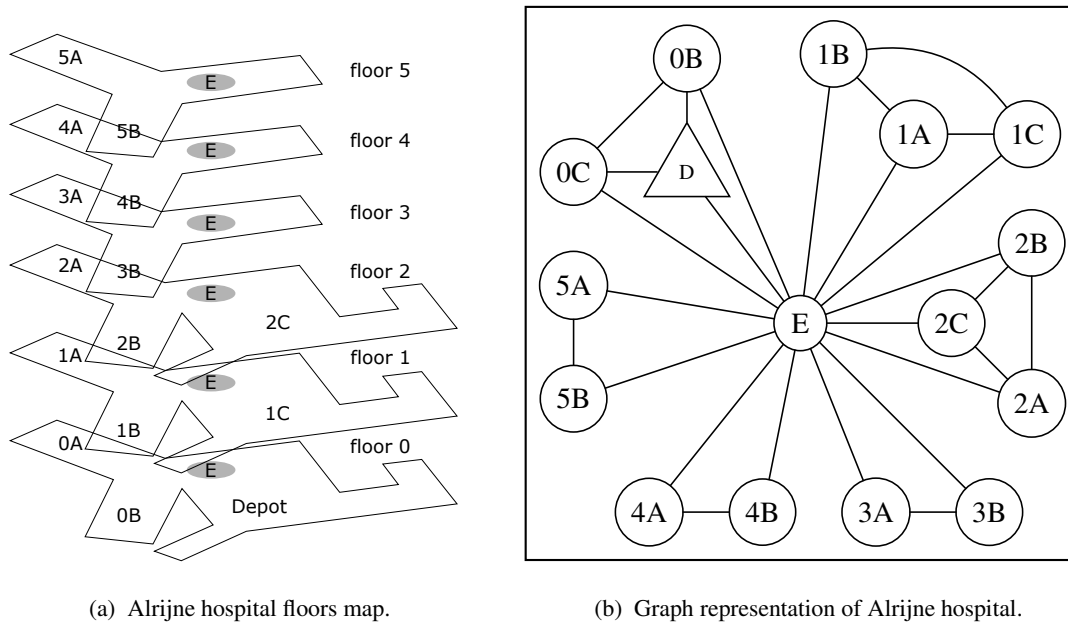
(a) Alrijne hospital floors map.

(b) Graph representation of Alrijne hospital.

Figure 5   The Alrijne hospital floor map and the graph representing connections between the different areas and floors of the hospital. The 27 departments are clustered into 14 areas (Figure 5a). The departments in each cluster are physically very close to each other. As illustrated in Figure 5a, each floor is divided in at most three areas with area A and B corresponding to the north and south wing of the hospital respectively and area C covers the right wing and the south zone. Node D (triangle) represents the pharmacy depot and node E the elevators. Arcs between nodes indicates the direct path between two areas.

of the hospital, once the $\overline{\boldsymbol{\xi}}$ associated to node equipped with sensor has been observed, the optimal worst-case routing problem can be achieved by solving $\max\limits_{\boldsymbol{y}\in\mathcal{Y}} \min\limits_{\boldsymbol{\xi}\in\Xi(\boldsymbol{w},\overline{\boldsymbol{\xi}})} \boldsymbol{\xi}^{\top}\boldsymbol{y}$. From the experiment we conclude that extra information can only be assimilated if the recourse decisions have adequate adaptability.

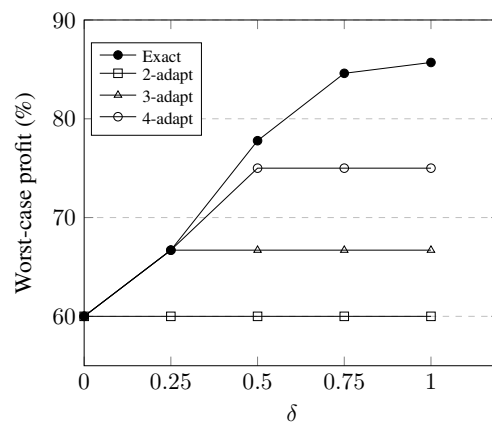**Figure 6**   Worst-case profit of the 2-, 3-, 4-adaptable and exact solution method for increasing values of δ and for $T = 2$ hours.

6.2.1. Impact of the Maximum Route Duration and Uncertainty Set   In our second experiment, we examine the impact of the maximum time duration T on the quality of the solution. To this end, we again evaluate the solution of the exact algorithm for different the values of δ by changing the time duration $T \in \{0.5, 1, 1.5, 2\}$ hours and considering the same uncertainty set adopted in the first experiment. The results are presented in Figure 7a. As expected, we observe that as the time duration increases, the collection process improves as the routing plan is able to visit more nodes in the network. As a result the impact of uncertainty diminishes. In contrast, the information provided by sensors is more impactful for short time durations. We conclude that as the hospital is trying to limit its investment in sensor infrastructure, they should consider the trade-off between number of sensors and the duration of the collecting process.
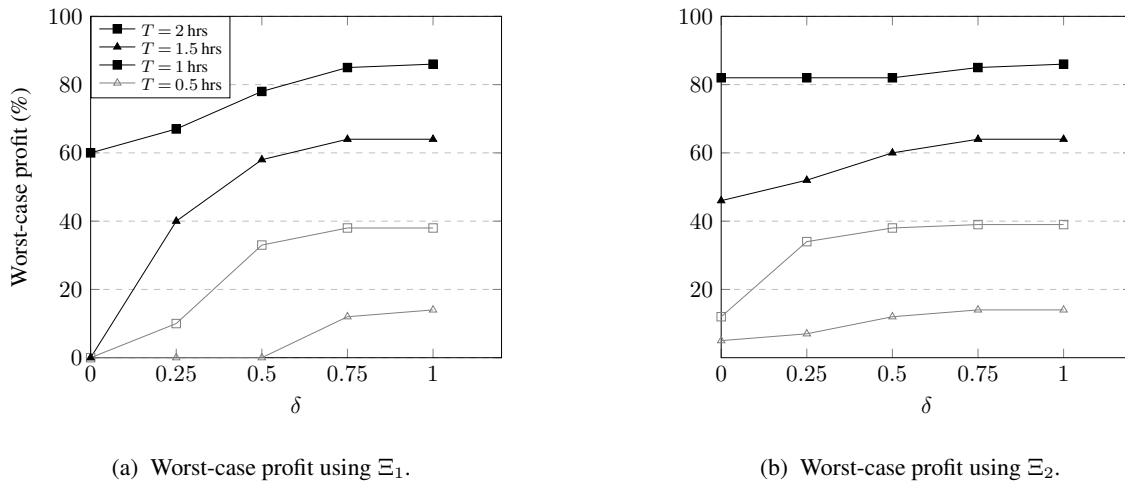

(a) Worst-case profit using Ξ_1.   (b) Worst-case profit using Ξ_2.

Figure 7   Comparison between optimal worst-case profits obtained for different values of δ and T using different uncertainty sets.

In our last experiment, we examine the situation where additional information is known on the number of crates at each node. In particular, we use data collected during the pilot to estimate the expected fraction of crates $\widehat{U}_i$ at each node i based on the orders of each department. We consider the uncertainty set $\Xi_2 = \left\{ \boldsymbol{\xi} \in \mathbb{R}_+^{14} : \boldsymbol{\xi}_i \in \left[\widehat{U}_i (1 - \theta), \widehat{U}_i (1 + \theta) \right], i \in \{1, \dots 14\}, \boldsymbol{e}^\top \boldsymbol{\xi} = 1 \right\}$ with $\theta = 0.75$. Table EC.2 summarizes the values of $\widehat{U}_i$ for each node. We solve the problem using the exact algorithm for different values of δ and T. The results are presented in Figure 7b. We observe that the sensor information does provide additional benefits for all values of T, however, their impact is less significant compared to the case where uncertainty set Ξ_1 is considered. This can be explained by the fact that Ξ_2 ensures that a minimum amount of crates is present at each node and that for some nodes the maximum value of $\widehat{U}_i$ is quite low, e.g., $\widehat{U}_3 = 0.05$,

therefore due to the presence of constraint $e^\top \xi = 1$, the orienteering routing plan will focus on nodes which have higher potential of collecting crates and avoid visiting nodes which have smaller values of $\widehat{U}_i$. There are two take away from this experiment. First, if additional information is available which limits the amount of uncertainty in the model, then the impact of the number of sensors diminishes compared to the case where uncertainty is uniform across all nodes, making the nodes somewhat "indistinguishable". Second, from a practical point of view, since problem (3) tries to model effectively an infinite horizon problem where if crates are not collected at a given node then they could potentially accumulate over time, having a constant U across all nodes ensures that the sensors are not biasedly placed in the network.

Acknowledgments

References

Y. Adulyasak, J.-F. Cordeau, and R. Jans. Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867, 2015.

J. Ayoub and M. Poss. Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Computational Management Science*, 13(2):219–239, 2016.

A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical programming*, 99(2):351–376, 2004.

A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton university press, 2009.

D. Bertsimas and C. Caramanis. Finite adaptability in multistage linear optimization. *IEEE Transactions on Automatic Control*, 55(12):2751–2766, 2010.

D. Bertsimas and I. Dunning. Multistage robust mixed-integer optimization with adaptive partitions. *Operations Research*, 64(4):980–998, 2016.

D. Bertsimas and A. Georghiou. Design of near optimal decision rules in multistage adaptive mixed-integer optimization. *Operations Research*, 63(3):610–627, 2015.

D. Bertsimas and A. Georghiou. Binary decision rules for multistage adaptive mixed-integer optimization. *Mathematical Programming*, 167(2):395–433, 2018.

D. Bertsimas and A. O'Hair. Learning preferences under noise and loss aversion: An optimization approach. *Operations Research*, 61(5):1190–1199, 2013.

D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3): 464–501, 2011.

D. Bertsimas, I. Dunning, and M. Lubin. Reformulation versus cutting-planes for robust optimization. *Computational Management Science*, 13(2):195–217, 2016.

D. Bertsimas, M. Sim, and M. Zhang. Adaptive distributionally robust optimization. *Management Science*, 65(2): 604–618, 2019.

N. Boland, I. Dumitrescu, G. Froyland, and T. Kalinowski. Minimum cardinality non-anticipativity constraint sets for multistage stochastic programming. *Mathematical Programming*, 157(1):69–93, 2016.

C. Buchheim and J. Kurtz. Min–max–min robust combinatorial optimization. *Mathematical Programming*, 163(1-2): 1–23, 2017.

A. M. Campbell, M. Gendreau, and B. W. Thomas. The orienteering problem with stochastic travel and service times. *Annals of Operations Research*, 186(1):61–81, 2011.

C. C. Carøe and J. Tind. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83(1):451–464, 1998.

A. Chassein, M. Goerigk, J. Kurtz, and M. Poss. Faster algorithms for min-max-min robustness for combinatorial problems with budgeted uncertainty. *European Journal of Operational Research*, 279(2):308–319, 2019.

X. Chen and Y. Zhang. Uncertain linear programs: Extended affinely adjustable robust counterparts. *Operations Research*, 57(6):1469–1482, 2009.

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.

M. Colvin and C. T. Maravelias. Modeling methods and a branch and cut algorithm for pharmaceutical clinical trial planning using stochastic programming. *European Journal of Operational Research*, 203(1):205–215, 2010.

E. Delage and D. Iancu. Robust multistage decision making. *INFORMS Tutorials in Operations Research*, pages 20–46, 2015.

L. Evers, T. Dollevoet, A. I. Barros, and H. Monsuur. Robust UAV mission planning. *Annals of Operations Research*, 222(1):293–315, 2014a.

L. Evers, K. Glorie, S. Van Der Ster, A. I. Barros, and H. Monsuur. A two-stage approach to the orienteering problem with stochastic weights. *Computers & Operations Research*, 43:248–260, 2014b.

M. Fischetti, J. J. S. Gonzalez, and P. Toth. Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, 10(2):133–148, 1998.

M. Fischetti, I. Ljubić, and M. Sinnl. Redesigning Benders decomposition for large-scale facility location. *Management Science*, 63(7):2146–2162, 2017.

E. Fragnière, J. Gondzio, and X. Yang. Operations risk management by optimally planning the qualified workforce capacity. *European Journal of Operational Research*, 202(2):518–527, 2010.

V. Gabrel, C. Murat, and A. Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014.

D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou. A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics*, 20(3):291–328, 2014.

A. Georghiou, W. Wiesemann, and D. Kuhn. Generalized decision rule approximations for stochastic programming via liftings. *Mathematical Programming*, 152(1):301–338, 2015.

V. Goel and I. E. Grossmann. A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves. *Computers & Chemical Engineering*, 28(8):1409–1429, 2004.

V. Goel and I. E. Grossmann. A class of stochastic programs with decision dependent uncertainty. *Mathematical programming*, 108(2-3):355–394, 2006.

A. Gunawan, H. C. Lau, and P. Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332, 2016.

V. Gupta and I. E. Grossmann. Solution strategies for multistage stochastic programming with endogenous uncertainties. *Computers & Chemical Engineering*, 35(11):2235–2247, 2011.

G. A. Hanasusanto, D. Kuhn, and W. Wiesemann. K-adaptability in two-stage robust binary programming. *Operations Research*, 63(4):877–891, 2015.

J. N. Hooker and G. Ottosson. Logic-based Benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.

T. W. Jonsbråten, R. J. Wets, and D. L. Woodruff. A class of stochastic programs withdecision dependent random elements. *Annals of Operations Research*, 82:83–106, 1998.

M. Joosten. Tracking medical crates in the Alrijne hospital through sensors. Master's thesis, Schoolof Business and Economics, Vrije Universiteit Amsterdam, 2019. Available at https://www.ubvu.vu.nl/pub/fulltext/scripties/27_2634607_0.pdf.

G. Laporte and F. V. Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142, 1993.

J. E. Macias, N. Goldbeck, P.-Y. Hsu, P. Angeloudis, and W. Ochieng. Endogenous stochastic optimisation for relief distribution assisted with unmanned aerial vehicles. *OR Spectrum*, 42(4):1089–1125, 2020.

G. P. McCormick. Nonlinear programming; theory, algorithms, and applications. Technical report, 1983.

C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4):326–329, 1960.

R. Paradiso. Numerical experiments. https://github.com/rospa/Exact-and-Approximate-Schemes-for-Robust-Optimization-Problems-with-DDID. Accessed July 23, 2021.

K. Postek and D. d. Hertog. Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing*, 28(3):553–574, 2016.

H. D. Sherali and A. Alameddine. A new reformulation-linearization technique for bilinear programming problems. *Journal of Global optimization*, 2(4):379–410, 1992.

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.

A. Subramanyam, C. E. Gounaris, and W. Wiesemann. K-adaptability in two-stage mixed-integer robust optimization. *Mathematical Programming Computation*, pages 1–32, 2019.

T. Tsiligirides. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35(9):797–809, 1984.

P. Vansteenwegen and D. Van Oudheusden. The mobile tourist guide: an OR opportunity. *OR insight*, 20(3):21–27, 2007.

P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, 2011.

P. Vayanos, D. Kuhn, and B. Rustem. Decision rules for information discovery in multi-stage stochastic programming. In *Proceedings of the 50th IEEE Conference on Decision and Control*, pages 7368–7373, 2011.

P. Vayanos, A. Georghiou, and H. Yu. Robust optimization with decision-dependent information discovery, 2020a. Available at arXiv:2004.08490.

P. Vayanos, D. McElfresh, Y. Ye, J. Dickerson, and E. Rice. Active preference elicitation via adjustable robust optimization, 2020b. Available at arXiv:2003.01899.

A. L. Veen. Tracking medical crates in the alrijne hospital through sensors: Initial data analysis of the sensory data of the alrijne hospital. Bachelor's thesis, Schoolof Business and Economics, Vrije Universiteit Amsterdam, 2020. Available at https://www.ubvu.vu.nl/pub/fulltext/scripties/27_2616957_0.pdf.

C. Verbeeck, P. Vansteenwegen, and E.-H. Aghezzaf. Solving the stochastic time-dependent orienteering problem with time windows. *European Journal of Operational Research*, 255(3):699–718, 2016.

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. Zou, S. Ahmed, and X. A. Sun. Stochastic dual dynamic integer programming. *Mathematical Programming*, 175 (1):461–502, 2019.

Electronic Companion

EC.1. Proofs for Section 3

PROOF OF PROPOSITION 1. To prove the proposition, we need to show that the two following properties hold.

(i) $\Phi(\boldsymbol{w}) = \Phi(\boldsymbol{w}) - (\Phi(\boldsymbol{w}) - \underline{\Phi}) \, \phi(\boldsymbol{w}, \boldsymbol{w}),$

(ii) $\Phi(\boldsymbol{w}) \geq \Phi(\boldsymbol{w}') - (\Phi(\boldsymbol{w}') - \underline{\Phi}) \, \phi(\boldsymbol{w}', \boldsymbol{w}) \quad \forall \boldsymbol{w}' \in \mathcal{W}.$

Property (i) holds since $\phi(\boldsymbol{w}, \boldsymbol{w}) = 0$. This property ensures tightness (i.e., the right hand side of the inequality coincides with the value of $\Phi(\boldsymbol{w})$ at point $\boldsymbol{w}$). Moreover, for $\boldsymbol{w}' \in \mathcal{W} \setminus \{\boldsymbol{w}\}$, we observe that $\phi(\boldsymbol{w}', \boldsymbol{w}) \geq 1$. As $\underline{\Phi} \leq \Phi(\boldsymbol{w})$ for all $\boldsymbol{w} \in \mathcal{W}$, property (ii) ensures that the inequalities never over estimate the value of $\Phi(\boldsymbol{w})$, thus the inequalities are valid. $\square$

PROOF OF THEOREM 1. The finite converge of Algorithm 1 relies on $\mathcal{W}$ being a finite set and on the convexity of the lower envelope of $\Phi(\boldsymbol{w})$. Since by Proposition 1 the inequalities (5) are valid and tight, in the worst-case the algorithm will generate all possible $\boldsymbol{w} \in \mathcal{W}$ thus terminating in $|\mathcal{W}|$ iterations. $\square$

PROOF OF LEMMA 1. Assume that there exist at least one $i \in \mathcal{N}_w$ such that $\boldsymbol{w}_i^1 > \boldsymbol{w}_i^2$. By definition, $\Xi(\boldsymbol{w}^1, \overline{\boldsymbol{\xi}}) \subseteq \Xi(\boldsymbol{w}^2, \overline{\boldsymbol{\xi}})$ and therefore $\max_{\overline{\boldsymbol{\xi}} \in \Xi} \min_{\boldsymbol{y} \in \mathcal{Y}} \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}^1, \overline{\boldsymbol{\xi}})} \boldsymbol{\xi}^\top \boldsymbol{P} \boldsymbol{y} \leq \max_{\overline{\boldsymbol{\xi}} \in \Xi} \min_{\boldsymbol{y} \in \mathcal{Y}} \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}^2, \overline{\boldsymbol{\xi}})} \boldsymbol{\xi}^\top \boldsymbol{P} \boldsymbol{y}$ which implies $\Phi(\boldsymbol{w}^1) \leq \Phi(\boldsymbol{w}^2)$, as desired. $\square$

PROOF OF PROPOSITION 2. To prove the proposition, we need to show the following conditions hold:

(i) $\Phi(\boldsymbol{w}) = \Phi(\boldsymbol{w}) - (\Phi(\boldsymbol{w}) - \underline{\Phi}) \, \rho(\boldsymbol{w}, \boldsymbol{w}),$

(ii) $\Phi(\boldsymbol{w}) \geq \Phi(\boldsymbol{w}') - (\Phi(\boldsymbol{w}') - \underline{\Phi}) \, \rho(\boldsymbol{w}', \boldsymbol{w}) \quad \forall \boldsymbol{w}' \in \mathcal{W},$

(iii) $\Phi(\boldsymbol{w}') - (\Phi(\boldsymbol{w}') - \underline{\Phi}) \, \rho(\boldsymbol{w}', \boldsymbol{w}'') \geq \Phi(\boldsymbol{w}') - (\Phi(\boldsymbol{w}') - \underline{\Phi}) \, \phi(\boldsymbol{w}', \boldsymbol{w}'') \quad \forall \boldsymbol{w}' \in \mathcal{W}.$

Condition (i) ensures tightness at integer solutions and holds since $\rho(\boldsymbol{w}, \boldsymbol{w}) = 0$. To show that condition (ii) holds we need to distinguish two cases. If $\boldsymbol{w} > \boldsymbol{w}'$, the cut is valid since $\rho(\boldsymbol{w}', \boldsymbol{w}) \geq 1$. When $\boldsymbol{w} \leq \boldsymbol{w}'$, $\rho(\boldsymbol{w}', \boldsymbol{w}) = 0$ that implies $\Phi(\boldsymbol{w}) \geq \Phi(\boldsymbol{w}')$ which follows from Lemma 1. Condition (iii) ensures that the information inequalities are stronger than inequalities (5) at fractional solutions. As $\Phi(\boldsymbol{w}) \geq \underline{\Phi}$, it is sufficient to prove that $\rho(\boldsymbol{w}, \boldsymbol{w}') \leq \phi(\boldsymbol{w}, \boldsymbol{w}')$. We have $\phi(\boldsymbol{w}, \boldsymbol{w}') - \rho(\boldsymbol{w}, \boldsymbol{w}') = \sum_{i \in \mathcal{N}_w : \boldsymbol{w}_i = 1} (1 - \boldsymbol{w}_i')$. Hence, if $\boldsymbol{w}_i' = 1$ for all $i \in \{\mathcal{N}_w : \boldsymbol{w}_i = 1\}$, then $\sum_{i \in \mathcal{N}_w : \boldsymbol{w}_i = 1} (1 - \boldsymbol{w}_i') = 0$ and $\phi(\boldsymbol{w}, \boldsymbol{w}') = \rho(\boldsymbol{w}, \boldsymbol{w}')$. In all the other cases, $\sum_{i \in \mathcal{N}_w : \boldsymbol{w}_i = 1} (1 - \boldsymbol{w}_i') > 0$ and therefore $\rho(\boldsymbol{w}, \boldsymbol{w}') < \phi(\boldsymbol{w}, \boldsymbol{w}')$, as desired. $\square$

PROOF OF THEOREM 2. Since $\mathcal{Y}$ is finite, in the worst-case $|\mathcal{Y}|$ steps are needed. In fact, after $|\mathcal{Y}|$ steps, set $\widehat{\mathcal{Y}}$ will be equivalent to set $\mathcal{Y}$ since in each iteration (8) generates a new variable $\boldsymbol{y} \notin \widehat{\mathcal{Y}}$ if $ub - lb > 0$. $\square$

EC.2. Extensions of the Exact Algorithm

Algorithms 1 and 2 can be further extended or slightly modified to tackle a richer class of problems. Below we list some of the possible extensions.

Finite Ξ and mixed integer $\mathcal{Y}$: If problem (1b) involves a finite uncertainty set Ξ and mixed integer set $\mathcal{Y}$, Algorithm 2 will produce the optimal solution in no more than $|\Xi|$ iterations. Intuitively, this is the case as there are $|\Xi|$ inner $\min_{\boldsymbol{y} \in \mathcal{Y}} \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \bar{\boldsymbol{\xi}})} \boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^\top \boldsymbol{P} \boldsymbol{y}$ problems, one for each $\bar{\boldsymbol{\xi}} \in \Xi$. Therefore, in the worst-case the column-and-constraint generation algorithm will have to evaluate the worst-case cost of all $|\Xi|$ values of $\bar{\boldsymbol{\xi}}$ to find the optimal solution.

Continuous Ξ and mixed integer $\mathcal{Y}$: If problem (1b) involves a continuous uncertainty set Ξ and mixed integer set $\mathcal{Y}$, Algorithm 2 will produce the optimal solution in a finite number of iterations. In particular, consider the case where $\mathcal{Y} := \{\boldsymbol{y}_B \in \{0,1\}^{N_{\boldsymbol{y}_B}}, \boldsymbol{y}_C \in \mathbb{R}^{N_{\boldsymbol{y}_C}} : \boldsymbol{F} \boldsymbol{y}_B + \boldsymbol{G} \boldsymbol{y}_C \leq \boldsymbol{h}\}$ is a bounded set for some matrices $\boldsymbol{F} \in \mathbb{R}^{m \times N_{\boldsymbol{y}_B}}$, $\boldsymbol{G} \in \mathbb{R}^{m \times N_{\boldsymbol{y}_C}}$ and vector $\boldsymbol{h} \in \mathbb{R}^m$. We can define sets $\mathcal{Y}_B := \{\boldsymbol{y}_B \in \{0,1\}^{N_{\boldsymbol{y}_B}} : \exists \boldsymbol{y}_C \in \mathbb{R}^{N_{\boldsymbol{y}_C}}$ s.t. $\boldsymbol{F} \boldsymbol{y}_B + \boldsymbol{G} \boldsymbol{y}_C \leq \boldsymbol{h}\}$ and $\mathcal{Y}_C(\boldsymbol{y}_B) := \{\boldsymbol{y}_C \in \mathbb{R}^{N_{\boldsymbol{y}_C}} : \boldsymbol{F} \boldsymbol{y}_B + \boldsymbol{G} \boldsymbol{y}_C \leq \boldsymbol{h}\}$. Problem (1b) can be written as

$$
\begin{aligned}
\Phi(\boldsymbol{w}) &= \max_{\bar{\boldsymbol{\xi}} \in \Xi} \min_{\boldsymbol{y}_B \in \mathcal{Y}_B} \min_{\boldsymbol{y}_C \in \mathcal{Y}_C(\boldsymbol{y}_B)} \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \bar{\boldsymbol{\xi}})} \boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^\top \boldsymbol{P}_B \boldsymbol{y}_B + \boldsymbol{\xi}^\top \boldsymbol{P}_C \boldsymbol{y}_c \\
&= \max_{\bar{\boldsymbol{\xi}} \in \Xi} \min_{\boldsymbol{y}_B \in \mathcal{Y}_B} \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \bar{\boldsymbol{\xi}})} \min_{\boldsymbol{y}_C \in \mathcal{Y}_C(\boldsymbol{y}_I)} \boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^\top \boldsymbol{P}_B \boldsymbol{y}_B + \boldsymbol{\xi}^\top \boldsymbol{P}_C \boldsymbol{y}_c \\
&= \max_{\bar{\boldsymbol{\xi}} \in \Xi} \min_{\boldsymbol{y}_B \in \mathcal{Y}_B} \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \bar{\boldsymbol{\xi}})} \max_{\boldsymbol{\lambda} \in \Lambda(\boldsymbol{\xi})} \boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^\top \boldsymbol{P}_B \boldsymbol{y}_B + \boldsymbol{\lambda}^\top \boldsymbol{F} \boldsymbol{y}_B - \boldsymbol{\lambda}^\top \boldsymbol{h},
\end{aligned}
$$

where $\Lambda(\boldsymbol{\xi}) = \{\boldsymbol{\lambda} \in \mathbb{R}_+^m : (\boldsymbol{P}_C)^\top \boldsymbol{\xi} + \boldsymbol{G}^\top \boldsymbol{\lambda} = 0\}$. The second equality follows from the min-max theorem since both sets $\Xi(\boldsymbol{w}, \bar{\boldsymbol{\xi}})$ and $\mathcal{Y}_C(\boldsymbol{y}_B)$ are convex and $\mathcal{Y}_C(\boldsymbol{y}_B)$ is bounded by construction, while the last equality results from the dualization of $\min_{\boldsymbol{y}_C \in \mathcal{Y}_C(\boldsymbol{y}_B)} \boldsymbol{\xi}^\top \boldsymbol{P}_C \boldsymbol{y}_C$. The last problem has a similar structure as (1b) and Algorithm 2 can be directly applied, producing the optimal solution in no more than $|\mathcal{Y}_B|$ iterations.

Binary first stage decisions $\boldsymbol{x} \in \mathcal{X}$: Binary first stage decisions $\boldsymbol{x} \in \mathcal{X} \subseteq \{0,1\}^{N_x}$ can be incorporated into the problem in two ways which will lead to two variants of the algorithm. The first way is to express the problem as

$$
\min_{\boldsymbol{w} \in \mathcal{W}, \boldsymbol{x} \in \mathcal{X}} \Phi(\boldsymbol{w}, \boldsymbol{x}), \tag{EC.1a}
$$

where

$$\Phi(\boldsymbol{w}, \boldsymbol{x}) = \max_{\bar{\boldsymbol{\xi}} \in \Xi} \; \min_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} \; \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \bar{\boldsymbol{\xi}})} \; \boldsymbol{\xi}^\top \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^\top \boldsymbol{P} \boldsymbol{y}. \tag{EC.1b}$$

Since both $\boldsymbol{w}$ and $\boldsymbol{x}$ are binary vectors, Algorithm 1 can still be applied in conjunction with the Logic Benders inequalities (5). In this case, in the worst-case Algorithm 1 will converge in $|\mathcal{W}||\mathcal{X}|$ iterations. Notice that the feasible region of the recourse decisions $\mathcal{Y}(\boldsymbol{x})$ can be affected by the first stage decisions. However, this does not impact Algorithm 2 which for each $\boldsymbol{x}$ will produce the optimal solution in $|\mathcal{Y}(\boldsymbol{x})|$ iterations in the worst-case. If for a given $\boldsymbol{x}$ the corresponding feasible region $\mathcal{Y}(\boldsymbol{x})$ results being empty, a feasibility cut must be added to problem (EC.1a) in order to exclude the solution from the problem. Note that due to the binary nature of variables $\boldsymbol{w}$, this can be easily extended to the case in which $\mathcal{Y}(\boldsymbol{w}, \boldsymbol{x})$.

An alternative modelling approach is to include the first stage decision in the definition of $\Phi(\boldsymbol{w})$ and express the problem as

$$\min_{\boldsymbol{w} \in \mathcal{W}} \; \Phi(\boldsymbol{w}), \tag{EC.2a}$$

where

$$\Phi(\boldsymbol{w}) = \min_{\boldsymbol{x} \in \mathcal{X}} \; \max_{\bar{\boldsymbol{\xi}} \in \Xi} \; \min_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} \; \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \bar{\boldsymbol{\xi}})} \; \boldsymbol{\xi}^\top \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^\top \boldsymbol{P} \boldsymbol{y}. \tag{EC.2b}$$

In this setting, evaluating $\Phi(\boldsymbol{w})$ requires to have an outer column-and-constraint generation algorithm to optimize over $\boldsymbol{x}$. This additional algorithm will iteratively solve the following relaxation of problem (EC.2b)

$$\min \; \tau$$

$$\text{s.t. } \boldsymbol{x} \in \mathcal{X}, \; \tau \in \mathbb{R}, \; \boldsymbol{y} : \widehat{\Xi} \mapsto \mathbb{R}^{N_{\boldsymbol{y}}}$$

$$\max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \bar{\boldsymbol{\xi}})} \boldsymbol{\xi}^\top \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^\top \boldsymbol{P} \boldsymbol{y}(\bar{\boldsymbol{\xi}}) \le \tau \quad \forall \bar{\boldsymbol{\xi}} \in \widehat{\Xi}$$

$$\boldsymbol{y}(\bar{\boldsymbol{\xi}}) \in \mathcal{Y}(\boldsymbol{x}) \qquad\qquad\qquad \forall \bar{\boldsymbol{\xi}} \in \widehat{\Xi},$$

where $\widehat{\Xi}$ is a finite subset of Ξ, to obtain a progressive solution $\boldsymbol{x}^*$ and a lower bound on $\Phi(\boldsymbol{w})$. Using $\boldsymbol{x}^*$ we subsequently evaluate $\max_{\bar{\boldsymbol{\xi}} \in \Xi} \min_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x}^*)} \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \bar{\boldsymbol{\xi}})} \boldsymbol{\xi}^\top \boldsymbol{Q} \boldsymbol{x}^* + \boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^\top \boldsymbol{P} \boldsymbol{y}$ using Algorithm 2 to generate an upper bound on the value of $\Phi(\boldsymbol{w})$ and the next $\bar{\boldsymbol{\xi}}$ to be added in set $\widehat{\Xi}$. The algorithm terminates when the values of the upper and lower bounds coincide. Similar to the previous algorithms, in the worst-case, this extra algorithmic layer will produce the optimal solution in $|\mathcal{X}|$ iterations.

Mixed integer first stage decisions $\boldsymbol{x} \in \mathcal{X}$: In the case where the first stage decisions $\boldsymbol{x} = (\boldsymbol{x}_B, \boldsymbol{x}_C)$ involve both binary $\boldsymbol{x}_B$ and continuous $\boldsymbol{x}_C$ decisions, $\Phi(\boldsymbol{w}, \boldsymbol{x}_B, \cdot)$ in problem (EC.1) will in general be non-convex.

Since Algorithm 1 approximates $\Phi(\cdot)$ using a convex piecewise linear underestimator, it cannot be used to approximate $\Phi(\boldsymbol{w}, \boldsymbol{x}_B, \cdot)$. We can, however, use formulation (EC.2) and the column-and-constraint generation algorithm outlined above. Unfortunately, in the worst-case the algorithm will convergence to the optimal solution asymptotically, unlike when $\boldsymbol{x}$ is purely binary where we have finite worst-case convergence.

Stochastic and distributionally robust optimization problems: It is interesting to note that Algorithm 1 can be used for any two-stage optimization problem (stochastic or robust) with decision-dependent information discovery. In all cases, $\boldsymbol{w}$ controls the timing of information and $\Phi(\boldsymbol{w})$ represents the value of the two-stage problem for the information structure prescribed by $\boldsymbol{w}$. The only requirements are (i) to be able to exactly evaluate $\Phi(\boldsymbol{w})$ so as Algorithm 1 can produce the optimal solution, and (ii) perform the evaluation efficiently for the algorithm to be practical.

EC.3. Proofs for Section 4

PROOF OF THEOREM 3. We show that for each feasible solution of problem (10) an equivalent solution in which $\boldsymbol{\alpha}$ belongs to set $\mathcal{A}_S$ exists. Let $(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\boldsymbol{y}^k, \boldsymbol{\beta}^k, \boldsymbol{\gamma}^k\}_{k \in \mathcal{K}})$ be a feasible solution of problem (10) such that for some $k \in \mathcal{K} \setminus \{K\}$, $\boldsymbol{\alpha}_k < \boldsymbol{\alpha}_{k+1}$. We prove that an equivalent solution such that $\boldsymbol{\alpha}_k \geq \boldsymbol{\alpha}_{k+1}$ always exists, and show how to generate such a solution. If the values of variables $(\boldsymbol{\alpha}_k,\ \boldsymbol{\beta}^k,\ \boldsymbol{\gamma}^k,\ \boldsymbol{y}^k)$ are interchanged with the values of variables $(\boldsymbol{\alpha}_{k+1},\ \boldsymbol{\beta}^{k+1}\,\boldsymbol{\gamma}^{k+1}, \boldsymbol{y}^{k+1})$, we obtain a solution that satisfy (11). We need now to show that this solution is feasible if $(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\boldsymbol{y}^k, \boldsymbol{\beta}^k, \boldsymbol{\gamma}^k\}_{k \in \mathcal{K}})$ is feasible and that has the same cost. Obviously, the solution satisfy constraints (10f) and (10e). Furthermore, the values of variables $\boldsymbol{\beta}$ remain unchanged, and hence the objective value does not change as well. We conclude that the two solutions are equivalent, as desired. $\square$

PROOF OF THEOREM 4. We prove the theorem by demonstrating that $\mathcal{A}_S \subseteq \mathcal{A}_B$, thus leveraging on Theorem 3 to demonstrate that at least one optimal solution of problem (10) satisfies $\boldsymbol{\alpha} \in \mathcal{A}_B$. For $K = 1$ the statement is trivial. Consider some $K > 1$. We prove the statement by contradiction. Consider $k = 1$ and assume that there exists $\boldsymbol{\alpha} \in \mathcal{A}_S$ with $\boldsymbol{\alpha}_1 < \frac{1}{K}$. Since $\boldsymbol{\alpha}_1 > 0$, there exists $\epsilon \in (0, \frac{1}{K})$ such that $\boldsymbol{\alpha}_1 = \frac{1}{K} - \epsilon$. Constraint $\boldsymbol{e}^\top \boldsymbol{\alpha} = 1$ implies that

$$\sum_{k \in \mathcal{K} \setminus \{1\}} \boldsymbol{\alpha}_k = 1 - \frac{1}{K} + \epsilon, \tag{EC.3a}$$

while constraints $\boldsymbol{\alpha}_k \geq \boldsymbol{\alpha}_{k+1}$ for all $k \in \mathcal{K} \setminus \{K\}$ imply

$$\boldsymbol{\alpha}_1 \geq \max_{k \in \mathcal{K} \setminus \{1\}} \boldsymbol{\alpha}_k. \tag{EC.3b}$$

We show that none of the $\boldsymbol{\alpha} \in \mathbb{R}_+^K$ such that (EC.3a) is satisfied satisfies (EC.3b), since

$$\boldsymbol{\alpha}_1 \geq \min_{\boldsymbol{\alpha}_{-1} \geq 0:(EC.3a)} \max_{k \in \mathcal{K} \setminus \{1\}} \boldsymbol{\alpha}_k = \frac{1 + \epsilon - \frac{1}{K}}{K - 1} \implies \epsilon \leq 0. \tag{EC.3c}$$

Therefore, we reach the contradiction that $\epsilon \in (0, \frac{1}{K})$. Consider now the generic component $k > 1$ and assume that $\boldsymbol{\alpha}_k > \frac{1}{k}$. This implies that there exists $\epsilon \in \left(0, 1 - \frac{1}{k}\right)$ such that $\boldsymbol{\alpha}_k = \frac{1}{k} + \epsilon$. Constraint $\boldsymbol{e}^\top \boldsymbol{\alpha} = 1$ implies

$$\sum_{\tilde{k} \in \mathcal{K} \setminus \{k\}} \boldsymbol{\alpha}_{\tilde{k}} = 1 - \frac{1}{k} - \epsilon, \tag{EC.4a}$$

while constraint constraints $\boldsymbol{\alpha}_k \geq \boldsymbol{\alpha}_{k+1}$ for all $k \in \mathcal{K} \setminus \{K\}$ imply

$$\min_{\left\{\tilde{k} \in \mathcal{K}: \tilde{k} < k\right\}} \boldsymbol{\alpha}_{\tilde{k}} \geq \boldsymbol{\alpha}_k \geq \max_{\left\{\tilde{k} \in \mathcal{K}: \tilde{k} > k\right\}} \boldsymbol{\alpha}_{\tilde{k}}. \tag{EC.4b}$$

We show that none of the $\boldsymbol{\alpha}$ that satisfies (EC.4a) will satisfy (EC.4b), since

$$\max_{\boldsymbol{\alpha}_{-k} \geq 0:(EC.4a)} \min_{\tilde{k} \in \{\mathcal{K}: \tilde{k} < k\}} \boldsymbol{\alpha}_{\tilde{k}} \geq \boldsymbol{\alpha}_k \implies \frac{1}{k} + \epsilon \leq \frac{1 - \epsilon - \frac{1}{k}}{k - 1} \implies \epsilon \leq 0.$$

Therefore, we reach the contradiction that $\epsilon \in (0, 1 - \frac{1}{k})$. $\square$

PROOF OF THEOREM 5. Let $(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \{\boldsymbol{y}^k, \boldsymbol{\beta}^k, \boldsymbol{\gamma}^k\}_{k \in \mathcal{K}})$ be an optimal solution to problem (13). For all $k' \in \mathcal{K}$ for which $\boldsymbol{\alpha}_{k'} = 0$, we construct the solution where we set $\boldsymbol{y}^{k'} := \boldsymbol{y}^k$ for some $k \in \mathcal{K}$ whose corresponding $\boldsymbol{\alpha}_k > 0$, which from Observation 1 we know is also an optimal solution. Using this modified optimal solution, we have that

$$\max_{\bar{\boldsymbol{\xi}} \in \Xi} \min_{k \in \mathcal{K}} \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{w}, \bar{\boldsymbol{\xi}})} \boldsymbol{\xi}^\top \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^\top \boldsymbol{P} \boldsymbol{y}^k$$

$$= \max_{\bar{\boldsymbol{\xi}} \in \Xi} \max_{\boldsymbol{\xi}^k \in \Xi(\boldsymbol{w}, \bar{\boldsymbol{\xi}})} \min_{k \in \mathcal{K}} \boldsymbol{\xi}^{k^\top} \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^{k^\top} \boldsymbol{P} \boldsymbol{y}^k \tag{EC.5a}$$

$$= \min_{k \in \mathcal{K}} \boldsymbol{\xi}^{k*^\top} \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^{k*^\top} \boldsymbol{P} \boldsymbol{y}^k \tag{EC.5b}$$

$$= \boldsymbol{\xi}^{k*^\top} \boldsymbol{C} \boldsymbol{w} + \boldsymbol{\xi}^{k*^\top} \boldsymbol{P} \boldsymbol{y}^k \quad \forall k \in \mathcal{K} \tag{EC.5c}$$

$$\geq \boldsymbol{\zeta}^{\boldsymbol{w}^\top} \boldsymbol{w} + \boldsymbol{\zeta}^{\boldsymbol{y}^\top} \boldsymbol{y}^k \quad \forall k \in \mathcal{K}, \tag{EC.5d}$$

where (EC.5a) follows from (Vayanos et al. 2020a, Lemma 1). Vectors $\{\boldsymbol{\xi}^{k*}\}_{k\in\mathcal{K}}$ indicate the optimal solution of (EC.5a), which we substitute resulting in (EC.5b). Equality (EC.5c) follows since by construction of the optimal solution $\boldsymbol{\xi}^{k*\top}\boldsymbol{C}\boldsymbol{w} + \boldsymbol{\xi}^{k*\top}\boldsymbol{P}\boldsymbol{y}^k$ is equal for all $k \in \mathcal{K}$, while the last inequality results from the definition of $\boldsymbol{\zeta}^w$ and $\boldsymbol{\zeta}^y$. From strong duality we also have that

$$\max_{\bar{\boldsymbol{\xi}}\in\Xi}\ \min_{k\in\mathcal{K}}\ \max_{\boldsymbol{\xi}\in\Xi(\boldsymbol{w},\bar{\boldsymbol{\xi}})}\ \boldsymbol{\xi}^\top\boldsymbol{C}\boldsymbol{w} + \boldsymbol{\xi}^\top\boldsymbol{P}\boldsymbol{y}^k$$

$$= \left[\begin{array}{ll} \min & \boldsymbol{b}^\top\left(\boldsymbol{\beta} + \displaystyle\sum_{k\in\mathcal{K}}\boldsymbol{\beta}^k\right) \\[2ex] \text{s.t.} & \boldsymbol{\alpha}\in\mathbb{R}_+^K,\ \boldsymbol{\beta}\in\mathbb{R}_+^L,\ \boldsymbol{\beta}^k\in\mathbb{R}_+^L,\ \boldsymbol{\gamma}^k\in\mathbb{R}^{N_\xi},\ k\in\mathcal{K} \\[2ex] & \boldsymbol{e}^\top\boldsymbol{\alpha} = 1 \\[2ex] & \boldsymbol{A}^\top\boldsymbol{\beta} = \displaystyle\sum_{k\in\mathcal{K}}\boldsymbol{w}\circ\boldsymbol{\gamma}^k \\[2ex] & \boldsymbol{A}^\top\boldsymbol{\beta}^k + \boldsymbol{w}\circ\boldsymbol{\gamma}^k = \boldsymbol{\alpha}_k\left(\boldsymbol{C}\boldsymbol{w} + \boldsymbol{P}\boldsymbol{y}^k\right) \quad \forall k\in\mathcal{K} \end{array}\right]$$

hence the result follows. $\quad\square$

EC.4. Supporting Tables for Numerical Examples and Case Study

Table EC.1   Instance description

Network	U	T	Network	U	T
TS1N15	0.10	5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70	TS1N30	0.05	5, 10, 15, 20, 25, 30, 35, 40, 46, 50, 55, 60, 65, 70, 73, 75, 80, 85
TS2N10	0.20	15, 20, 23, 25, 27, 30, 32, 35, 38	TS2N19	0.15	15, 20, 23, 25, 27, 30, 32, 35, 38, 40, 45
TS3N16	0.10	15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80	TS3N31	0.05	15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 110, 105, 110

Table EC.2   Location areas and departments of the Alrijne hospital clustered based on the proximity of the departments on each floor. Column Node and Department denote the cluster in which each department is assigned, for example node 1A covers Nuclear Medicine and Radiology. The column Expected Crates reports the expected percentages of crates at each node derived from data used in Veen (2020).

Node	Department	Expected Crates	Node	Department	Expected Crates
0B	Gynecology outpatient clinic	5%	2A	Urology ward	10%
0C	Orthopedist	10%	2B	Day care/short stay & orthopedics	5%
	Ophthalmology outpatient clinic			Surgery ward	
	Ear, nose and throat outpatient clinic		2C	Anesthesia	6%
	Orthopedics outpatient clinic			Operating theatres	
	Wound center		3A	Geriatric trauma unit	7%
1A	Nuclear medicine	10%		Neurology	
	Radiology		3B	Cardiac care unit	5%
1B	Poli plastic surgery	7%		Cardiology II	
1C	Cardiology I	5%	4A	Mother and child center	7%
	Lung diseases outpatient clinic		4B	Digestive and liver diseases	8%
	Endoscopy			Oncology ward	
	Dermatology outpatient clinic		5A	Child and youth department	10%
			5B	Dialysis	5%