A Bilevel Optimization Approach for a Class of Combinatorial Problems with Disruptions and Probing

Leonardo Lozano
Operations, Business Analytics & Information Systems, University of Cincinnati, Cincinnati, Ohio 45221, leolozano@uc.edu

Juan S. Borrero
School of Industrial Engineering & Management, Oklahoma State University, Stillwater, OK 74078, juan.s.borrero@okstate.edu

We consider linear combinatorial optimization problems under uncertain disruptions that increase the cost coefficients of the objective function. A decision-maker, or planner, can invest resources to probe the components (i.e., the coefficients) in order to learn their disruption status. In the proposed probing optimization problem, the planner, knowing just the disruptions' probabilities, selects which components to probe subject to a probing budget in a first decision stage. Then, the uncertainty realizes and the planner observes the disruption status of the probed components, after which the planner solves the combinatorial problem in the second stage. In contrast to standard two-stage stochastic optimization, the planner does not have access to the full uncertainty realization in the second stage. Consequently, the planner cannot directly optimize the second stage objective function, which is given by the actual cost post-disruptions, and the decisions have to be made based on an estimate of the cost. By assuming that the estimate is given by the conditional expected cost given the information revealed by probing, we reformulate the probing optimization problem as a bilevel problem with multiple followers and propose an exact algorithm based on a value function reformulation and three heuristic algorithms. We derive theoretical results that bound the value of information and the price of not having full information, and a bound on the required probing budget that attains the same performance than full information. Our extensive computational experiments suggest that probing a fraction of the components is sufficient to yield large improvements in the optimal value, that our exact algorithm is competitive for small to medium scale instances, and that the proposed heuristics find high-quality solutions in large-scale instances.

Key words: Optimization under uncertainty; Bilevel optimization; Value of information; Integer programming

---

1. Introduction

We study a class of combinatorial optimization problems for which uncertain disruptions (or failures) can affect the objective function coefficients. To describe our problem setting, we first consider the following deterministic baseline combinatorial problem:

$$\min_{x} \sum_{i \in \mathcal{N}} c_i x_i \tag{1a}$$

$$\text{s.t. } x \in \mathcal{X} \subseteq \{0,1\}^N, \tag{1b}$$

where $\mathcal{N} = \{1, \ldots, N\}$, c_i denotes the *nominal cost* corresponding to binary variable x_i, and $\mathcal{X} \neq \emptyset$ is the set of feasible solutions.

We consider the case in which uncertain disruptions (failures) affect the cost coefficients component-wise, i.e., if a disruption impacts component $i \in \mathcal{N}$, then its cost is increased from c_i to c_i', where $c_i' > c_i$. This is a common setting in the literature, for example in shortest path problems, where an arc failure results in an increased travel time (cost), or in a manufacturing setting, where a defective product results in additional cost due to reprocessing. The decision-maker, or *planner*, knows that component i is disrupted with probability p_i, $i \in \mathcal{N}$.

We consider the case in which the uncertainty about the disruptions can be mitigated by investing resources into information collection before solving the combinatorial problem. We refer to any such activity to gather additional information as *probing*. Probing a given component confirms whether the component is disrupted. Our main focus is to measure the value of information provided by probing.

In the first decision stage of our problem, the planner selects which components to probe subject to a limited budget. Then the uncertainty is realized and the planner observes whether the probed components are disrupted or not. In a second decision stage the planner solves the combinatorial problem based on the partial information collected by the probes. The objective of the planner is to decide which components to probe in order to minimize the expected cost.

Having partial, rather than full information, implies that the optimization decisions in the second stage have to be made by *estimating* any uncertainty that was not revealed by probing, and, consequently, that the optimal decisions in the second stage might not necessarily minimize the *actual* realization of the costs. Because the probing decisions are made in order to minimize actual rather than estimated costs, such a discrepancy implies that the probing problem has a two-stage *bilevel* structure. Specifically, the probing problem can be framed as a two-stage stochastic program where the second-stage is a bilevel optimization problem. Next, we present a formalization of the problem under consideration.

1.1. Problem statement

Let the occurrence of disruptions be represented by a random vector $J = (J_i\colon i \in \mathcal{N})$. We assume that the random variables J_i are independent and J_i has a Bernoulli distribution with parameter $p_i \geq 0$, $i \in \mathcal{N}$. That is, J_i takes the value 1 if a disruption impacts component i and takes the value 0, otherwise, and $P[J_i = 1] = p_i$. We denote the realizations of J by $\xi \in \{0,1\}^N$.

For each component $i \in \mathcal{N}$ let z_i be the first-stage binary decision variable that takes the value 1 if the planner decides to probe component i and 0 otherwise. If $z_i = 1$ and component i is not disrupted (i.e., if $\xi_i = 0$), then the cost at i is given by c_i. Else, if the planner probes $i \in \mathcal{N}$ and i is disrupted (i.e., $\xi_i = 1$), then the cost at i is given by c_i'. On the other hand, if the planner does not probe component $i \in \mathcal{N}$, then we assume that the planner estimates the cost of i to be $c_i(1-p_i) + c_i' p_i$. We define the *estimated cost* when the planner selects a probing plan z, if scenario ξ happens, and if the solution plan $x \in \mathcal{X}$ is executed, by

$$\begin{aligned}\hat{C}(z,\xi,x) &= \sum_{i\in\mathcal{N}\colon z_i=1} \left[c_i(1-\xi_i)x_i + c_i'\xi_i x_i\right] + \sum_{i\in\mathcal{N}\colon z_i=0} \left[c_i(1-p_i)x_i + c_i' p_i x_i\right] \\ &= \sum_{i\in\mathcal{N}} \left[c_i\big((1-\xi_i)z_i + (1-p_i)(1-z_i)\big)x_i + c_i'\big(\xi_i z_i + p_i(1-z_i)\big)x_i\right]. \end{aligned} \tag{2}$$

It can be shown that the estimate defined in (2) is precisely the conditional expected value of the cost given that the planner selects $x \in \mathcal{X}$ and that $J_i = \xi_i$ for all i with $z_i = 1$, see Appendix A.

After a solution plan $x \in \mathcal{X}$ is executed from the estimated cost, the planner observes the *actual cost* $C(\xi, x)$, corresponding to the realization of scenario $\xi \in \Xi$, which is given by

$$C(\xi,x) = \sum_{i\in\mathcal{N}} \left[c_i(1-\xi_i)x_i + c_i'\xi_i x_i\right]. \tag{3}$$

Observe that the estimated cost function $\hat{C}(z,\xi,x)$ depends on vector ξ; however, it only observes the ξ_i components for which $z_i = 1$, i.e., the occurrence or absence of disruptions for the components that are probed. On the other hand, the actual cost function $C(\xi, x)$ is independent of the probing plan because once the solution is executed, then the disruptions occur (or not) according to the realized scenario.

Let B, $0 \le B \le N$, be the available budget for probing in the first stage. We formulate the probing problem as a two-stage stochastic problem of the form

$$\Gamma_B := \min_{z} \ \mathbb{E}[\mathcal{F}(z,J)] \tag{4a}$$

$$\text{s.t.} \ \sum_{i \in \mathcal{N}} z_i \le B \tag{4b}$$

$$z \in \{0,1\}^N, \tag{4c}$$

where

$$\mathbb{E}[\mathcal{F}(z,J)] = \sum_{\xi \in \Xi} \pi^{\xi} \mathcal{F}(z,\xi). \tag{5}$$

In equation (5) π^{ξ} is the probability of scenario ξ, which can be computed from the disruption probabilities p_i, $i \in \mathcal{N}$, by

$$\pi^{\xi} = \prod_{i \in \mathcal{N}, \xi_i = 1} p_i \prod_{i \in \mathcal{N}, \xi_i = 0} (1 - p_i) \qquad \xi \in \Xi. \tag{6}$$

For each scenario $\xi \in \Xi$, the function $\mathcal{F}(z,\xi)$ in (5) is the actual cost incurred by the planner if the probing decisions are given by z and the realized scenario is ξ:

$$\mathcal{F}(z,\xi) = \min_{x} \Big\{ C(\xi, x) \colon x \in \arg\min_{x'} \big\{ \hat{C}(z,\xi,x') \colon x' \in \mathcal{X} \big\} \Big\}, \tag{7}$$

where the outer "min" in (7) breaks ties in case that there are several plans x that minimize the estimated cost $\hat{C}$.

The first-stage problem decides the probing plan z with the objective of minimizing the expected actual cost (4a), which is computed by solving the second-stage problem. In the second stage (7), for a given probing plan z and a realization of the uncertainty, the planner first selects a solution that minimizes the estimated cost and then computes the actual cost after executing the selected solution. In contrast to standard two-stage stochastic problems in which the second stage solves a single-level problem for each realization of the uncertainty, our second stage problem is defined by "argmin" constraints resulting in a bilevel structure.

Observe that Γ_0 corresponds to the expected actual cost associated with a *limited information* (LI) approach, in which the planner is not able to perform any probing. In contrast, Γ_N is the expected actual cost under a *full information* (FI) approach, in which the planner is able to completely remove the uncertainty from the problem. For any B we refer to

$\Gamma_0 - \Gamma_B$ as the *value of information* associated with having B probes whereas $\Gamma_B - \Gamma_N$ is referred as the *price of not having full-information* associated with having B probes. The value of information measures the largest possible expected cost savings associated with being able to probe B components. The price of not having full information measures the largest possible expected cost incurred due to probing only B components.

1.2. Statement of Contribution

The main contributions of this work are as follows. We propose a special class of two-stage stochastic programs for combinatorial problems under uncertainty that can quantify the value of knowing with certainty whether disruptions impact a subset of the components of the cost function. We reformulate the two-stage stochastic problem as a bilevel problem with multiple followers and solve the problem using exact solution methods as well as heuristic approaches.

We derive bounds on the value of information and the price of not having full information. The first of these bounds does not depend on B whereas the second one does. These results suggests that there are instances where the value of information is independent on the number of probes, and that there are instances where optimally placed probes always reveal at least some information about the remaining uncertainty. We also derive an upper bound on the minimum value of B for which $\Gamma_B = \Gamma_N$, i.e., the minimum budget needed to obtain the same objective value as under full information.

In addition, we conduct computational experiments on two problem settings: a project selection problem, which is modelled as a knapsack problem, and a routing under uncertainty problem, which is modelled as a shortest path problem. Our computational results show that that:

1. Having full information often results in a considerable improvement to the optimal value.

2. Probing a small subset of the components can yield large improvements to the optimal value and in many cases get close to the value obtained under full information.

3. Our proposed upper bound on the value of B for which $\Gamma_B = \Gamma_N$, shows that often probing roughly less than 40% of the components is enough to achieve the same performance as having full information for the routing problem and probing less than 65% of the components is to enough match the performance of full information for the project selection problem.

4. Our exact solution approach for the bilevel problem with multiple followers is able to solve medium-sized problem instances for both problem settings, and the computational effort is highly dependent on the number of scenarios.
5. Our proposed heuristics are able to consistently find high-quality solutions for large problem instances for both problem settings considered and with up to 500 scenarios.

To the best of our knowledge, our bilevel formulation is the first one to provide optimal probing plans (as opposed to approximation algorithms) for these type of information discovery problems over independent uncertain events, as well as the first one to conduct an extensive computational study regarding the value of information for the two problem settings selected.

2. Literature review

Our work is related to previous work in probing problems, stochastic programming, and bilevel optimization. Regarding using probing to reduce uncertainty, Gupta et al. (2016) consider a probing setting with unlimited budget for combinatorial problems. In their model, only 'items' probed in the first stage can be included in the objective function of the second stage. The authors propose an approximation algorithm and focus on bounding an 'adaptivity gap' between optimal online and offline policies. Similar models are studied by Gupta and Nagarajan (2013) and Adamczyk et al. (2016), who also propose approximation algorithms. Guha and Munagala (2012), and particularly, Goel et al. (2010), consider a probing setting that is closer to what we do in this paper. The main differences is that their focus is on developing approximation algorithms. They show that for specific combinatorial problems there is a constant-factor approximation algorithm (based on solving the 'outlier problem') for both the online and offline versions of the problem. We note, however, that the complexity of the algorithm depends on the computational complexity of the outlier problem. This complexity, in turn, is problem-dependent, thus constant-factor approximations might not be available to all problems. More general probing problems with a similar approach to Goel et al. (2010) have also been considered in the two-stage stochastic programming literature. This problems are, as we do in this paper, formulated as two-stage stochastic problems. Their focus, is however, on the statistical and mathematical properties of the model, see Artstein and Wets (1993), Artstein (1994, 1999).

Our work can be considered to be related to stochastic optimization problems with decision-dependent uncertainty (Jonsbråten et al. 1998, Goel and Grossmann 2004, Hellemo et al. 2018), because by probing the planner updates the probability distribution in the second stage. The main differences of these models with our work is that not all uncertainty is revealed in the second stage and that the actual distribution used in the first stage remains independent of the probing decisions. These properties imply that standard stochastic optimization problems with decision-dependent uncertainty do not require a bilevel structure, as we do in this case.

The second stage problem problem in our setting is a bilevel programs with integer requirements at both levels (Gümüş and Floudas 2005, DeNegre and Ralphs 2009, Lozano and Smith 2017b, Fischetti et al. 2017, Tahernejad et al. 2020). This class of problems is Σ_2^p-hard (Caprara et al. 2013) and are normally solved by branch-and-cut methods or iterative decomposition algorithms (Kleinert et al. 2021). Standard bilevel problems with integer requirements at both levels are typically deterministic and stochastic versions are analogous to two-stage stochastic integer problems, see for example (Cormican et al. 1998, Janjarassuk and Linderoth 2008, Beck et al. 2023). On the other hand, two-stage stochastic problems where the second-stage problem is a bilevel integer program (similar to what we consider in this paper), have been studied rather scarcely in the literature, see for instance Alizadeh et al. (2013), Özaltın et al. (2018).

3. Exact solution approach

We first reformulate our problem as a bilevel problem with multiple followers and analyze two limiting cases. We then propose exact solution approaches to solve the proposed reformulation.

3.1. A bilevel reformulation of Problem (4)

A standard approach to solve two-stage stochastic programs is to use a deterministic equivalent monolithic formulation, in which the expectation is represented by a weighted sum over all possible scenarios and copies of the second-stage variables are introduced for each scenario. We generate such monolithic formulation for problem (4) as follows. For scenario $\xi \in \Xi$, let $x^\xi \in \mathcal{X}$ be the solution plan under scenario ξ and let all solution plans be represented by $x^\Xi = \left\{x^\xi\right\}_{\xi\in\Xi}$, that is, x^ξ is an optimal solution to the inner problem in (7). Then

$$\Gamma_B = \min_{z,x^\Xi} \sum_{\xi\in\Xi} \pi^\xi \sum_{i\in\mathcal{N}} \left[c_i(1-\xi_i)x_i^\xi + c_i'\xi_i x_i^\xi\right] \tag{8a}$$

$$\text{s.t.} \sum_{i\in\mathcal{N}} z_i \leq B \tag{8b}$$

$$x^\xi \in \arg\min_x \Big\{ \sum_{i\in\mathcal{N}} \big[c_i\big((1-\xi_i)z_i + (1-p_i)(1-z_i)\big)x_i + c_i'\big(\xi_i z_i + p_i(1-z_i)\big)x_i\big] : x \in \mathcal{X} \Big\} \quad \forall \xi \in \Xi \tag{8c}$$

$$z \in \{0,1\}^N. \tag{8d}$$

Formulation (8) describes a bilevel problem with multiple followers (one per scenario) under the so-called *optimistic assumption* (Dempe 2002), that is, it breaks possible ties in the arg min by selecting the second-stage solutions that minimize the objective function in (8a). The probing stage corresponds to the *leader's* problem, while the original combinatorial problem corresponds to the *follower's* problem. The objective is to minimize the expected actual cost while ensuring that the second-stage solutions minimize the estimated cost for each scenario.

We first analyze the two limiting cases of problem (8) that can be readily solved as single-stage problems. For the first limiting case, suppose that there is no probing budget, that is, $B = 0$, which implies that $z_i = 0$ for all $i \in \mathcal{N}$. In this case, the estimated cost function $\hat{C}$ simplifies to

$$\hat{C}(z,\xi,x) = \sum_{i\in\mathcal{N}} \big[c_i(1-p_i)x_i + c_i' p_i x_i\big] \tag{9}$$

and evaluates to the same value under all scenarios for a given x-solution. As a result, optimal decisions for all scenarios coincide and all constraints in (8c) can be replaced by the single constraint

$$x \in \arg\min_{x'} \Big\{ \sum_{i\in\mathcal{N}} \big[c_i(1-p_i)x_i' + c_i' p_i x_i'\big] : x' \in \mathcal{X} \Big\}. \tag{10}$$

On the other hand, since we select a single solution x for all the scenarios, the objective function of (8) simplifies to

$$\sum_{\xi\in\Xi} \pi^\xi \sum_{i\in\mathcal{N}} \big[c_i(1-\xi_i)x_i + c_i'\xi_i x_i\big] = \sum_{i\in\mathcal{N}} \Big(\sum_{\xi\in\Xi:\,\xi_i=0} c_i x_i + \sum_{\xi\in\Xi:\,\xi_i=1} c_i' x_i \Big) \tag{11a}$$

$$= \sum_{i\in N} \big[c_i(1-p_i)x_i + c_i' p_i x_i\big], \tag{11b}$$

that is, the leader's objective (8a) of minimizing the expected actual cost coincides with the follower's objective (10) of minimizing the estimated cost. As a result, if there is no probing ($B = 0$), then the two-stage problem (8) simplifies to

$$\Gamma_0 = \min \Big\{ \sum_{i\in\mathcal{N}} \big[c_i(1-p_i)x_i + c_i' p_i x_i\big] : x \in \mathcal{X} \Big\}, \tag{12}$$

which is a single-stage combinatorial problem where an optimal solution is selected entirely based on the estimated probabilities p_i, $i \in \mathcal{N}$. An alternative interpretation of (12) is that in the absence of probing, both the actual cost and estimated cost functions simplify to a naive expected cost function based on the probabilities p_i.

For the second limiting case consider that all nodes can be probed, i.e., $B = N$. We have that $z_i = 1$ for all $i \in N$ and the estimated cost becomes

$$\hat{C}(z, \xi, x) = \sum_{i \in \mathcal{N}} \left[c_i (1 - \xi_i) x_i + c_i' \xi x_i \right]. \tag{13}$$

In other words, the estimated cost for each scenario is equal to the actual cost since there is full information and it is readily seen that problem (8) becomes the single-stage problem

$$\Gamma_N = \min \Big\{ \sum_{\xi \in \Xi} \pi^\xi \sum_{i \in \mathcal{N}} \left[c_i (1 - \xi_i) x_i^\xi + c_i' \xi x_i^\xi \right] : x^\xi \in \mathcal{X} \quad \forall \xi \in \Xi \Big\}. \tag{14}$$

Moreover, as in (14) there are no coupling requirements between solutions x^ξ, we have that

$$\Gamma_N = \sum_{\xi \in \Xi} \pi^\xi \min \Big\{ \sum_{i \in \mathcal{N}} \left[c_i (1 - \xi_i) x_i^\xi + c_i' \xi x_i^\xi \right] : x^\xi \in \mathcal{X} \Big\}. \tag{15}$$

As a result, Γ_N can be computed by decomposing problem (14) and solving $|\Xi|$ single-stage combinatorial problems independently (and potentially in parallel).

3.2. Value function approach

One of the major challenges in solving discrete bilevel problems is the construction of valid relaxations. A common relaxation from the bilevel literature is known as the *High-point relaxation*, which is obtained by dropping the requirement of optimality in the lower-level problem enforced by constraints (8c). After removing these constraints in problem (8), the probing variables become irrelevant because they do not appear in the objective function, and the high-point relaxation reduces to

$$\Gamma^H = \sum_{\xi \in \Xi} \pi^\xi \min \Big\{ \sum_{i \in \mathcal{N}} \left[c_i (1 - \xi_i) x_i^\xi + c_i' \xi x_i^\xi \right] : x^\xi \in \mathcal{X} \Big\} = \Gamma_N, \tag{16}$$

i.e., the high-point relaxation is precisely the full information problem Γ_N (see Equation (15)), which generally yields weak lower bounds on the optimal value.

Value function approaches have been successfully used in the literature for bilevel problems with one follower (Mitsos 2010, Lozano and Smith 2017b). For our multi-follower problem observe that formulation (8) can be equivalently posed as

$$\Gamma_B = \min \ \sum_{\xi \in \Xi} \pi^\xi \sum_{i \in \mathcal{N}} \left[c_i (1 - \xi_i) x_i^\xi + c_i' \xi x_i^\xi \right] \tag{17a}$$

$$\text{s.t.} \sum_{i\in\mathcal{N}} z_i \leq B \tag{17b}$$

$$\sum_{i\in\mathcal{N}} \left[c_i\big((1-\xi_i)z_i+(1-p_i)(1-z_i)\big)x_i^{\xi}+c_i'\big(\xi_i z_i+p_i(1-z_i)\big)x_i^{\xi}\right] \leq$$

$$\sum_{i\in\mathcal{N}} \left[c_i\big((1-\xi_i)z_i+(1-p_i)(1-z_i)\big)x_i+c_i'\big(\xi_i z_i+p_i(1-z_i)\big)x_i\right] \quad \forall x\in\mathcal{X}, \forall\xi\in\Xi \tag{17c}$$

$$z\in\{0,1\}^N;\ x^{\xi}\in\mathcal{X} \quad \forall\xi\in\Xi, \tag{17d}$$

which is a single-level non-linear binary optimization problem with potentially exponentially many constraints. Observe that the nonlinearities are products between binary variables ($z_i x_i^{\xi}$) and thus readily linearized.

We define a relaxed value function problem (RVF) by considering a subset of second-stage solutions $\hat{\mathcal{X}} \subseteq \mathcal{X}$. Formally, RFV($\hat{\mathcal{X}}$) is defined as (17), except that $\mathcal{X}$ is replaced by $\hat{\mathcal{X}}$ in (17c). Let $\Gamma_B(\hat{\mathcal{X}})$ be the optimal objective function value of RFV($\hat{\mathcal{X}}$) and note that for any $\hat{\mathcal{X}} \subseteq \mathcal{X}$, it holds that $\Gamma_B(\hat{\mathcal{X}}) \leq \Gamma_B$.

We propose a cutting-plane algorithm that iteratively explores second-stage solutions and adds them to $\hat{\mathcal{X}}$. Solving RFV($\hat{\mathcal{X}}$) for each $\hat{\mathcal{X}}$ provides a sequence of non-decreasing lower bounds on Γ_B. Upper bounds are obtained by solving the lower-level problem for fixed probing plans stemming from the relaxed problems. Algorithm 1 formalizes our proposed cutting-plane approach. Line 1 initializes the lower and upper bounds, sets $\hat{\mathcal{X}} = \emptyset$, and creates a trivial probing plan $\bar{z} = 0$. Line 2 computes a lower bound by solving RVF for the solutions obtained thus far in set $\hat{\mathcal{X}}$. Line 3 solves the combinatorial problem to minimize the estimated cost for the probing plan $\hat{z}$ found in Line 2. Line 4 computes the expected actual cost and updates the upper bound if necessary. Line 5 stops the execution of the algorithm if the lower bound is equal to the upper bound. Otherwise, it updates the set of second-stage solutions by adding all the solutions discovered in Line 3 and goes back to Line 2 to continue with the cut-generation algorithm.

Algorithm 1 terminates finitely with an optimal solution because the set of all possible second-stage solutions $\mathcal{X}$ is finite. Note that all the problems solved are feasible (setting all the variables to zero gives a trivial feasible solution) and bounded (all the variables are binary) and as a result there is no need to check for unboundedness or feasibility. On the other hand, because of the optimistic assumption that the follower breaks ties among alternative optimal solutions by selecting the one that minimizes the leader's objective,

Algorithm 1: Cutting-plane Algorithm

1: Set $LB = -\infty$, $UB = \infty$, $\hat{\mathcal{X}} = \emptyset$, and incumbent solution $\bar{z} = 0$.

2: Solve RFV($\hat{\mathcal{X}}$). Set $LB = \Gamma_B(\hat{\mathcal{X}})$ and record the optimal probing plan found $\hat{z}$.

3: For each scenario $\xi \in \Xi$, solve lower-level problem $\min\left\{\hat{C}(\hat{z}, \xi, x) \colon x \in \mathcal{X}\right\}$ and record the optimal solutions found $\hat{x}^\xi$.

4: If $\sum_{\xi \in \Xi} \pi^\xi C(\xi, \hat{x}^\xi) < UB$, then update $UB = \sum_{\xi \in \Xi} \pi^\xi C(\xi, \hat{x}^\xi)$ and $\bar{z} = \hat{z}$.

5: If $LB = UB$, terminate with an optimal solution given by $\bar{z}$. Otherwise, update $\hat{\mathcal{X}} := \hat{\mathcal{X}} \cup \left\{\bigcup_{\xi \in \Xi} \hat{x}^\xi\right\}$ and return to Line 2.

we need to be careful when recording an optimal solution $\hat{x}^\xi$ at Line 3 to account for the case in which there exist alternative optimal x-solutions (see Appendix B).

The major computational challenge for Algorithm 1 is solving RFV($\hat{\mathcal{X}}$), because enforcing $x^\xi \in \mathcal{X}$ requires making copies of all the constraints needed to describe $\mathcal{X}$ for each scenario. As the number of scenarios grows, RFV($\hat{\mathcal{X}}$) becomes prohibitively large and considerably challenging to solve.

4. Theoretical bounds on the performance of probing

In this section we derive theoretical bounds on the difference of objective value that can be attained with more probing resources (Section 4.1). We also provide an scheme to find an upper-bound on the budget B^* that is required to attain the same performance as full information, i.e., to attain that $\Gamma_{B^*} = \Gamma_N$ (Section 4.2). The quality of some of these bounds are evaluated empirically in Section 6.3.

4.1. Bounds on the value of information and on the price of not having full information

We derive bounds on the value of information $\Gamma_0 - \Gamma_B$ and on the price of not having full information $\Gamma_B - \Gamma_N$. We first provide results for the effect of additional probing on the optimal estimated cost for a given scenario and then construct bounds for the value of probing in terms of the expected actual cost.

For any given subset of components $\mathcal{P} \subseteq \mathcal{N}$ and any scenario $\xi \in \Xi$, let $\phi^\xi_{\mathcal{P}}$ be the optimal estimated cost for scenario ξ when the components in $\mathcal{P}$ are probed. That is,

$$\phi^\xi_{\mathcal{P}} = \min\left\{\sum_{i \in \mathcal{P}} \left[c_i(1-\xi_i) + c'_i \xi_i\right] x_i + \sum_{i \in \mathcal{N} \setminus \mathcal{P}} \left[c_i(1-p_i) + c'_i p_i\right] x_i \colon x \in \mathcal{X}\right\}. \tag{18}$$

Let $x^{\xi,\mathcal{P}}$ be an optimal solution associated with $\phi^\xi_{\mathcal{P}}$. Lemma 1 presents upper and lower bounds on the difference in optimal estimated cost for two nested probing plans, that is, two plans such that one probes a subset of the components probed by the other plan.

Lemma 1. *Let $\xi \in \Xi$ and $\mathcal{Q} \subseteq \mathcal{P} \subseteq \mathcal{N}$ be given. Then*

$$\sum_{i\in\mathcal{P}\setminus\mathcal{Q}} (c'_i - c_i)(p_i - \xi_i)x_i^{\xi,\mathcal{Q}} \le \phi^\xi_{\mathcal{Q}} - \phi^\xi_{\mathcal{P}} \le \sum_{i\in\mathcal{P}\setminus\mathcal{Q}} (c'_i - c_i)(p_i - \xi_i)x_i^{\xi,\mathcal{P}}. \tag{19}$$

Proof. For convenience, let us denote

$$C_i^\xi = c_i(1-\xi_i) + c'_i\xi_i \quad i \in \mathcal{N}, \xi \in \Xi \text{ and } \hat{C}_i = c_i(1-p_i) + c'_i p_i \quad i \in \mathcal{N},$$

then $\phi^\xi_{\mathcal{Q}} = \sum_{i\in\mathcal{Q}} C_i^\xi x_i^{\xi,\mathcal{Q}} + \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} \hat{C}_i x_i^{\xi,\mathcal{Q}}$. We have that

$$\phi^\xi_{\mathcal{Q}} = \sum_{i\in\mathcal{Q}} C_i^\xi x_i^{\xi,\mathcal{Q}} + \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} \hat{C}_i x_i^{\xi,\mathcal{Q}} \tag{20a}$$

$$\le \sum_{i\in\mathcal{Q}} C_i^\xi x_i^{\xi,\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} \hat{C}_i x_i^{\xi,\mathcal{P}} \tag{20b}$$

$$\le \sum_{i\in\mathcal{P}} C_i^\xi x_i^{\xi,\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} \hat{C}_i x_i^{\xi,\mathcal{P}} - \sum_{i\in\mathcal{P}\setminus\mathcal{Q}} C_i^\xi x_i^{\xi,\mathcal{P}} \tag{20c}$$

$$\le \sum_{i\in\mathcal{P}} C_i^\xi x_i^{\xi,\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{P}} \hat{C}_i x_i^{\xi,\mathcal{P}} + \sum_{i\in\mathcal{P}\setminus\mathcal{Q}} \hat{C}_i x_i^{\xi,\mathcal{P}} - \sum_{i\in\mathcal{P}\setminus\mathcal{Q}} C_i^\xi x_i^{\xi,\mathcal{P}} \tag{20d}$$

$$\le \phi^\xi_{\mathcal{P}} + \sum_{i\in\mathcal{P}\setminus\mathcal{Q}} (c'_i - c_i)(p_i - \xi_i)x_i^{\xi,\mathcal{P}}, \tag{20e}$$

where the second line follows from the optimality of $x^{\xi,\mathcal{Q}}$, the third by rearranging the terms, and the last one from the definition of $\phi^\xi_{\mathcal{P}}$. By doing similar steps, starting from $\phi^\xi_{\mathcal{P}}$ rather than $\phi^\xi_{\mathcal{Q}}$ it can be shown that $\phi^\xi_{\mathcal{P}} \le \phi^\xi_{\mathcal{Q}} + \sum_{i\in\mathcal{P}\setminus\mathcal{Q}} (c'_i - c_i)(\xi_i - p_i)x_i^{\xi,\mathcal{Q}}$. The combination of both inequalities gives the result. ∎

Now we turn our attention to the effects of probing on the expected actual cost. For a given $\mathcal{P} \subset \mathcal{N}$, let $\gamma(\mathcal{P})$ be the expected actual cost corresponding to nodes in $\mathcal{P}$ being probed, i.e., $\gamma(\mathcal{P})$ is the objective function in (8) for the probing plan $z_i = 1$ for any $i \in \mathcal{P}$ and $z_i = 0$ $i \notin \mathcal{P}$. From the definition of $x^{\xi,\mathcal{P}}$ and constraint (8c) it follows that,

$$\gamma(\mathcal{P}) = \sum_{\xi\in\Xi} \pi^\xi \sum_{i\in\mathcal{N}} \left[c_i(1-\xi_i) + c'_i\xi_i\right] x_i^{\xi,\mathcal{P}}. \tag{21}$$

Note that the relationship between $\gamma(\mathcal{P})$ and Γ_B is given by

$$\Gamma_B = \min\{\gamma(\mathcal{P}) \colon |\mathcal{P}| = B, \mathcal{P} \subseteq \mathcal{N}\}. \tag{22}$$

Theorem 1 presents upper and lower bounds for the change in optimal expected actual cost corresponding to two nested probing plans.

Theorem 1. *Let $\mathcal{Q} \subseteq \mathcal{P} \subseteq \mathcal{N}$ be given. Then,*

$$\sum_{\xi\in\Xi}\pi^{\xi}\Big[\sum_{i\in\mathcal{N}\setminus\mathcal{P}}(c_i'-c_i)(p_i-\xi_i)(x_i^{\xi,\mathcal{P}}-x_i^{\xi,\mathcal{Q}})\Big] \le \gamma(\mathcal{Q})-\gamma(\mathcal{P}) \le \sum_{\xi\in\Xi}\pi^{\xi}\Big[\sum_{i\in\mathcal{N}\setminus\mathcal{Q}}(c_i'-c_i)(p_i-\xi_i)(x_i^{\xi,\mathcal{P}}-x_i^{\xi,\mathcal{Q}})\Big]. \tag{23}$$

Proof. Note that

$$\sum_{i\in\mathcal{N}}C_i^{\xi}x_i^{\xi,\mathcal{P}} = \sum_{i\in\mathcal{P}}C_i^{\xi}x_i^{\xi,\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{P}}\hat{C}_i x_i^{\xi,\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{P}}(C_i^{\xi}-\hat{C}_i)x_i^{\xi,\mathcal{P}} \tag{24a}$$

$$= \phi_{\mathcal{P}}^{\xi} + \sum_{i\in\mathcal{N}\setminus\mathcal{P}}(c_i'-c_i)(\xi_i-p_i)x_i^{\xi,\mathcal{P}}. \tag{24b}$$

and therefore

$$\gamma(\mathcal{P}) = \sum_{\xi\in\Xi}\pi^{\xi}\Big(\phi_{\mathcal{P}}^{\xi} + \sum_{i\in\mathcal{N}\setminus\mathcal{P}}(c_i'-c_i)(\xi_i-p_i)x_i^{\xi,\mathcal{P}}\Big). \tag{25}$$

Because an analogous result holds for $\mathcal{Q}$, we conclude that

$$\gamma(\mathcal{Q})-\gamma(\mathcal{P}) = \sum_{\xi\in\Xi}\pi^{\xi}\Big(\phi_{\mathcal{Q}}^{\xi}-\phi_{\mathcal{P}}^{\xi} + \sum_{i\in\mathcal{N}\setminus\mathcal{Q}}(c_i'-c_i)(\xi_i-p_i)x_i^{\xi,\mathcal{Q}} - \sum_{i\in\mathcal{N}\setminus\mathcal{P}}(c_i'-c_i)(\xi_i-p_i)x_i^{\xi,\mathcal{P}}\Big) \tag{26a}$$

$$\le \sum_{\xi\in\Xi}\pi^{\xi}\Big(\sum_{i\in\mathcal{P}\setminus\mathcal{Q}}(c_i'-c_i)(p_i-\xi_i)x_i^{\xi,\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{Q}}(c_i'-c_i)(\xi_i-p_i)x_i^{\xi,\mathcal{Q}} - \sum_{i\in\mathcal{N}\setminus\mathcal{P}}(c_i'-c_i)(\xi_i-p_i)x_i^{\xi,\mathcal{P}}\Big) \tag{26b}$$

$$\le \sum_{\xi\in\Xi}\pi^{\xi}\Big(\sum_{i\in\mathcal{N}\setminus\mathcal{Q}}(c_i'-c_i)(\xi_i-p_i)x_i^{\xi,\mathcal{Q}} - \sum_{i\in\mathcal{N}\setminus\mathcal{Q}}(c_i'-c_i)(\xi_i-p_i)x_i^{\xi,\mathcal{P}}\Big), \tag{26c}$$

where (26b) follows from the upper-bound in Lemma 1, and thus the upper-bound in (23) follows. The lower-bound in (23) follow from a similar procedure, using the lower-bound in Lemma 1. ∎

Theorem 1 can be used to provide non-trivial bounds on $\Gamma_B - \Gamma_{B'}$ for any $B' > B \ge 0$. These bounds, however, require to know in advance optimal solutions to the second-stage problem. Corollary 1, shown next, removes some of these limitation and provides upper-bounds that only depend on the cost coefficients and the 'inner-most' optimal plan.

Corollary 1. *Let $\mathcal{Q} \subseteq \mathcal{P} \subseteq \mathcal{N}$ be given. Then,*

$$-2\sum_{i\in\mathcal{N}\setminus\mathcal{P}}(c_i'-c_i)p_i(1-p_i) \le \gamma(\mathcal{Q})-\gamma(\mathcal{P}) \le 2\sum_{i\in\mathcal{N}\setminus\mathcal{Q}}(c_i'-c_i)p_i(1-p_i). \tag{27}$$

Moreover,

$$-\sum_{i\in\mathcal{N}\setminus\mathcal{P}}(c_i'-c_i)p_i(1-p_i) \le \gamma(\emptyset)-\gamma(\mathcal{P}) \le \sum_{i\in\mathcal{N}}(c_i'-c_i)p_i(1-p_i). \tag{28}$$

In particular

$$-\frac{1}{2}\sum_{i\in\mathcal{N}\setminus\mathcal{P}}(c_i'-c_i) \le \gamma(\mathcal{Q})-\gamma(\mathcal{P}) \le \frac{1}{2}\sum_{i\in\mathcal{N}\setminus\mathcal{Q}}(c_i'-c_i), \tag{29}$$

and

$$-\frac{1}{4}\sum_{i\in\mathcal{N}\setminus\mathcal{P}}(c_i'-c_i) \le \gamma(\emptyset)-\gamma(\mathcal{P}) \le \frac{1}{4}\sum_{i\in\mathcal{N}}(c_i'-c_i). \tag{30}$$

Proof. By Theorem (1) we have that,

$$\gamma(\mathcal{Q})-\gamma(\mathcal{P}) \le \sum_{\xi\in\Xi}\pi^\xi\Big[\sum_{i\in\mathcal{N}\setminus\mathcal{Q}}(c_i'-c_i)(p_i-\xi_i)(x_i^{\xi,\mathcal{P}}-x_i^{\xi,\mathcal{Q}})\Big] \tag{31a}$$

$$= \sum_{i\in\mathcal{N}\setminus\mathcal{Q}}(c_i'-c_i)\sum_{\xi\in\Xi}\pi^\xi(p_i-\xi_i)(x_i^{\xi,\mathcal{P}}-x_i^{\xi,\mathcal{Q}}) \tag{31b}$$

$$= \sum_{i\in\mathcal{N}\setminus\mathcal{Q}}(c_i'-c_i)\Big[\sum_{\xi\in\Xi,\xi_i=0}\pi^\xi(p_i-\xi_i)(x_i^{\xi,\mathcal{P}}-x_i^{\xi,\mathcal{Q}}) + \sum_{\xi\in\Xi,\xi_i=1}\pi^\xi(\xi_i-p_i)(x_i^{\xi,\mathcal{Q}}-x_i^{\xi,\mathcal{P}})\Big]. \tag{31c}$$

Fix $i\in\mathcal{N}$ and note that

$$\sum_{\xi\in\Xi,\xi_i=0}\pi^\xi(p_i-\xi_i)(x_i^{\xi,\mathcal{P}}-x_i^{\xi,\mathcal{Q}}) = \sum_{\xi\in\Xi,\xi_i=0}\pi^\xi p_i(x_i^{\xi,\mathcal{P}}-x_i^{\xi,\mathcal{Q}}) \tag{32}$$

$$\le \sum_{\xi\in\Xi,\xi_i=0}\pi^\xi p_i \tag{33}$$

$$\le p_i(1-p_i), \tag{34}$$

where the first inequality follows because $\pi^\xi p_i \ge 0$ and $x_i^{\xi,\mathcal{P}}-x_i^{\xi,\mathcal{Q}} \le 1$ and the final inequality because $\sum_{\xi\in\Xi,\xi_i=0}\pi^\xi = 1-p_i$. Analogously,

$$\sum_{\xi\in\Xi,\xi_i=1}\pi^\xi(\xi_i-p_i)(x_i^{\xi,\mathcal{Q}}-x_i^{\xi,\mathcal{P}}) = \sum_{\xi\in\Xi,\xi_i=1}\pi^\xi(1-p_i)(x_i^{\xi,\mathcal{Q}}-x_i^{\xi,\mathcal{P}}) \tag{35}$$

$$\le \sum_{\xi\in\Xi,\xi_i=1}\pi^\xi(1-p_i) \tag{36}$$

$$\leq (1-p_i)p_i. \tag{37}$$

Thus, it can be concluded that

$$\gamma(\mathcal{Q}) - \gamma(\mathcal{P}) \leq \sum_{i \in \mathcal{N} \setminus \mathcal{Q}} 2(c_i' - c_i)p_i(1-p_i).$$

The final inequality follows since $p_i(1-p_i) \leq 1/4$ as $p_i \in [0,1]$ for all $i \in \mathcal{N}$.

Now, suppose $\mathcal{Q} = \emptyset$. In this case, optimal solution $x^{\xi,\emptyset}$ do not depend on ξ, and therefore $x^{\xi,\emptyset} = x^0$ for all $\xi \in \Xi$. In this case, Equation (31c) becomes

$$\gamma(\emptyset) - \gamma(\mathcal{P}) \leq \sum_{i \in \mathcal{N}} (c_i' - c_i) \Big[\sum_{\xi \in \Xi, \xi_i = 0} \pi^\xi p_i (x_i^{\xi,\mathcal{P}} - x_i^0) + \sum_{\xi \in \Xi, \xi_i = 1} \pi^\xi (1-p_i)(x_i^0 - x_i^{\xi,\mathcal{P}}) \Big] \tag{38}$$

$$\leq \sum_{i \in \mathcal{N}} (c_i' - c_i) \Big[\sum_{\xi \in \Xi, \xi_i = 0} \pi^\xi p_i x_i^{\xi,\mathcal{P}} - \sum_{\xi \in \Xi, \xi_i = 1} \pi^\xi (1-p_i) x_i^{\xi,\mathcal{P}} - p_i(1-p_i)x_i^0 + p_i(1-p_i)x_i^0 \Big] \tag{39}$$

$$\leq \sum_{i \in \mathcal{N}} (c_i' - c_i) \Big[\sum_{\xi \in \Xi, \xi_i = 0} \pi^\xi p_i x_i^{\xi,\mathcal{P}} - \sum_{\xi \in \Xi, \xi_i = 1} \pi^\xi (1-p_i) x_i^{\xi,\mathcal{P}} \Big] \tag{40}$$

$$\leq \sum_{i \in \mathcal{N}} (c_i' - c_i) p_i (1-p_i), \tag{41}$$

where the last inequality follows as $0 \leq x_i^{\xi,\mathcal{P}} \leq 1$ for all $\xi \in \Xi$. ∎

Let $\mathcal{B} \subseteq \mathcal{N}$ denote the components probed in an optimal probing plan associated with Γ_B. Corollary 1 implies that the value of information, $\Gamma_0 - \Gamma_B$, is upper-bounded by

$$\Gamma_0 - \Gamma_B \leq \sum_{i \in \mathcal{N}} (c_i' - c_i) p_i (1-p_i) \leq \frac{1}{4} \sum_{i \in \mathcal{N}} (c_i' - c_i), \tag{42}$$

for any $1 \leq B \leq N$. Similarly, for the price of not having full information, $\Gamma_B - \Gamma_N$, we have

$$\Gamma_B - \Gamma_N \leq 2 \sum_{i \in \mathcal{N} \setminus \mathcal{B}} (c_i' - c_i) p_i (1-p_i) \leq \frac{1}{2} \sum_{i \in \mathcal{N} \setminus \mathcal{B}} (c_i' - c_i). \tag{43}$$

Observe that there is a remarkable asymmetry in these bounds. On the one hand, the bound on the value of information does not depend on the number of components that are probed, whereas the bound on the price of not having full information does. The bound on the value of information might be explained from the observation that, in general, one might gather all the relevant uncertain information of the system by just probing one component (see for example, Remark 1). The bound for the price of not having full information might be interpreted as implying that there exist instances where optimally placed probes always reveal at least some information about the remaining uncertainty. Next, we provide an example that show that where some of these bounds are tight.

Remark 1. *Consider the shortest path problem in Figure 1 with source node 1 and sink node 3 with $\epsilon > 0$ arbitrary. On each arc there is a failure with probability 1/2. If there is no probing resources available, i.e., if $B = 0$, then the estimate for both paths is the same (both are estimated to be 1) and both have the same actual expected cost of 1, therefore $\Gamma_0 = 1$. Now consider $B = 1$ and that arc $(1,3)$ is probed. Then, in all scenarios ξ with $\xi_{(1,3)} = 0$ the optimal path is 1-3, with a cost $1-\epsilon$. By contrast, in all scenarios ξ with $\xi_{(1,3)} = 1$, the optimal path is 1-2-3, with a cost of 1. The actual expected cost in this case would be $1/2 \times (1-\epsilon) + 1/2 \times (1) = 1 - \epsilon/2$. Moreover, it can be verified that if $B = 1$ then probing $(1,3)$ is optimal, therefore, $\Gamma_1 = 1 - \epsilon/2$ and $\Gamma_0 - \Gamma_1 = \epsilon/2$. Evaluating the upper-bound proposed we get that $\sum_{i \in \mathcal{N}} (c_i' - c_i)/4 = \epsilon/2$, which is exactly the same as $\Gamma_0 - \Gamma_1$, and thus the upper bound in Corollary 1, particularly Equation (42), is tight.*

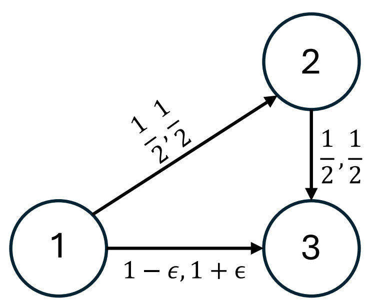

Figure 1 **Over each arc we denote the costs by c_i, c_i'.**

4.2. Upper-bound on probing budget

Next, we derive a bound on the amount of budget needed to achieve the same objective value as when solving the problem under perfect information. Formally, we are interested in finding the minimum budget value B^* such that $\Gamma_{B^*} = \Gamma_N$. The problem of finding B^* is at least as challenging as solving the original problem since for any candidate budget value B' we must solve the original problem to optimality in order to obtain $\Gamma_{B'}$. As a result, we propose a simpler approach to obtain an upper bound on B^* described in Algorithm 2.

Line 1 starts the procedure by solving the problem under perfect information and recording the optimal solutions found for each scenario denoted by $\hat{x}^\xi$. The intuition behind our proposed approach is that a probing plan the probes all the components $i \in \mathcal{N}$ for which at least one solution $\hat{x}_i^\xi = 1$ is likely to achieve an objective value close to Γ_N. For this reason, Line 1 defines set $\mathcal{Q}$ to contain all the components described above. Line 2 sets a probing plan $\hat{z}$ to probe all the components included in set $\mathcal{Q}$ and solves the follower problems of minimizing the estimated cost for each scenario. Line 3 checks if, for each scenario, the optimal estimated cost obtained is equal to the estimated cost function evaluated at $\hat{x}^\xi$. If this is the case, then probing plan $\hat{z}$ achieves an objective function equal to Γ_N since each $\hat{x}^\xi$ is an alternative optimal solution to the follower problem of minimizing the estimated cost and $\hat{B} = |\mathcal{Q}|$ establishes an upper bound on B^*. Otherwise, there must exist at least one scenario $\xi \in \Xi$ and one component $j \in \mathcal{N} \setminus \mathcal{Q}$ for which the solution that minimizes the estimated cost function does not match $\hat{x}_j^\xi$. Line 4 identifies this component, adds it to set $\mathcal{Q}$, and returns to Line 2 to update the probing plan and continue executing the algorithm.

Algorithm 2: Budget Upper Bound Algorithm

1: Solve the limiting case in which $B = N$ and record the optimal solutions found for each scenario $\hat{x}^\xi$. Define set $\mathcal{Q} = \{i \in \mathcal{N} : \exists \xi \text{ such that } \hat{x}_i^\xi = 1\}$.

2: Set probing plan $\hat{z}_i = 1$ for all $i \in \mathcal{Q}$ and $\hat{z}_i = 0$ otherwise. For each scenario $\xi \in \Xi$, solve lower-level problem $\min\left\{\hat{C}(\hat{z}, \xi, x) : x \in \mathcal{X}\right\}$ and record the optimal solutions found $\tilde{x}^\xi$.

3: If $\hat{C}(\hat{z}, \xi, \tilde{x}^\xi) = \hat{C}(\hat{z}, \xi, \hat{x}^\xi)$ for all $\xi \in \Xi$, then go to Line 5. Otherwise, go to Line 4.

4: Identify a component $j \in \mathcal{N} \setminus \mathcal{Q}$ for which $\tilde{x}_j^\xi \neq \hat{x}_j^\xi$ for some $\xi \in \Xi$. Update set $\mathcal{Q} := \mathcal{Q} \cup \{j\}$ and return to Line 2.

5: Terminate with an upper bound on B^* given by $\hat{B} = |\mathcal{Q}|$.

The algorithm terminates in a finite number of steps because in the worst case it obtains $\mathcal{Q} = \mathcal{N}$ and returns the trivial bound $\hat{B} = N$.

5. Heuristic approaches

We propose three heuristic solution approaches for the problem. Our first heuristic is a simple randomized greedy constructive approach that prioritizes probing components with low cost coefficients. Algorithm 3 describes our proposed heuristic approach.

Algorithm 3: Randomized Greedy Heuristic

1: Sort the components $i \in \mathcal{N}$ according to their cost coefficients. Let π be the ordering of the components, where π_k denotes the index for the kth component in ordering π.

2: For $k = 1, \ldots, N$ set probing plan $\hat{z}_{\pi_k} = 1$ with probability λ. Stop once B components have been selected or when $k = N$.

3: For each scenario $\xi \in \Xi$, solve lower-level problem $\min\left\{\hat{C}(\hat{z}, \xi, x) \colon x \in \mathcal{X}\right\}$ and record the optimal solutions found $\hat{x}^{\xi}$. Store the objective value given by $\sum_{\xi \in \Xi} \pi^{\xi} C(\xi, \hat{x}^{\xi})$.

4: Repeat lines 2 and 3 for a given number of iterations. Return the best solution found.

Line 2 introduces randomization, depending on the parameter $\lambda \in [0, 1]$, in order to add diversity to the pool of solutions explored. We consider this first approach as the most naive way in which a practitioner could quickly obtain probing plans for their problem. Thus, we use this heuristic as a baseline to measure the performance of our more advanced heuristic approaches described below.

Our second heuristic approach follows the same intuition as our procedure to find an upper bound on B^*. That is, the components for which x-solutions to the perfect information setting are equal to 1 could be good candidates to be probed. Algorithm 4 describes our second heuristic approach.

Algorithm 4: Heuristic Based on Perfect Information

1: Solve the limiting case in which $B = N$ and record the optimal solutions found for each scenario $x^{\xi, N}$. For each $i \in \mathcal{N}$ record the number of scenarios in which $x_i^{\xi, N} = 1$. Denote this number by s_i.

2: Sort the components $i \in \mathcal{N}$ in non-increasing order according to s_i. Let π be the ordering of the components, where π_k denotes the index for the kth component in ordering π.

3: For $k = 1, \ldots, N$ set probing plan $\hat{z}_{\pi_k} = 1$ with probability λ. Stop once B components have been selected or when $k = N$.

4: For each scenario $\xi \in \Xi$, solve lower-level problem $\min\left\{\hat{C}(\hat{z}, \xi, x) \colon x \in \mathcal{X}\right\}$ and record the optimal solutions found $\hat{x}^{\xi}$. Store the objective value given by $\sum_{\xi \in \Xi} \pi^{\xi} C(\xi, \hat{x}^{\xi})$.

5: Repeat lines 2 to 4 for a given number of iterations. Return the best solution found.

In this case the ordering prioritizes components for which $x_i^{\xi, N} = 1$ across multiple scenarios. As with the first heuristic, line 3 introduces randomization in order to add diversity to the pool of solutions explored.

Our third heuristic approach defines probing plans according to the estimated cost function. We consider the problem of minimizing the expected estimated cost:

$$\min_{z,x^{\Xi}} \sum_{i\in\mathcal{N}} \left[c_i\big((1-\xi_i)z_i+(1-p_i)(1-z_i)\big)x_i + c_i'\big(\xi_i z_i + p_i(1-z_i)\big)x_i\right] \tag{44a}$$

$$\text{s.t.} \sum_{i\in\mathcal{N}} z_i \leq B \tag{44b}$$

$$x^{\xi}\in\mathcal{X} \quad \forall \xi\in\Xi \tag{44c}$$

$$z\in\{0,1\}^N, \tag{44d}$$

which is a single-level problem that can be readily solved with an off-the-shelf optimization solver. The intuition behind our third heuristic approach is that (near) optimal solutions to (44) have a high chance of performing well in the original problem. Algorithm 5 describes our third heuristic approach.

Algorithm 5: Heuristic Based on Minimizing the Estimated Cost Function

1: Solve problem (44) and record the optimal probing plan $\hat{z}$ and the optimal solutions found for each scenario $\hat{x}^{\xi}$.
2: For each scenario $\xi\in\Xi$, compute the expected actual cost given by $\sum_{\xi\in\Xi}\pi^{\xi}C(\xi,\hat{x}^{\xi})$.
3: Add a no-good cut $\sum_{i\in N:\hat{z}_i=1} z_i \leq \sum_{i\in N}\hat{z}_i - 1$ to formulation (44). Repeat lines 1 and 2 for a given number of iterations. Return the best solution found.

6. Computational results

We conduct a computational study to compare the performance of the proposed algorithms. We measure the value of information and the price of not having full information over a set of instances from the literature for a shortest path problem, and over a set of synthetic instances for a project selection problem. We code our algorithms in Java using Eclipse SDK version 4.7.1 and all optimization problems are solved using CPLEX 20.1 with a time limit of one hour (3600s). All experiments are conducted on an Intel(R) Xeon(R) CPU E5-1650 v4 at 3.60GHz with 32GB of memory. The source code and problem instances will be publicly available at GitHub.

We use sample average approximation (SAA) to estimate the second-stage expected value since $|\Xi|$ grows exponentially as N increases. We note that SAA is a common approach to estimate expectations in two-stage settings. SAA has an exponentially fast convergence rate in terms of the number of scenarios used (Kleywegt et al. 2002) and has been shown to be highly accurate in routing problems (Verweij et al. 2003).

6.1. Results for a shortest path problem with disruptions and probing

We first study a shortest path problem in which arcs are susceptible to uncertain disruptions that increase their cost and the decision maker is able to probe a limited set of arcs, revealing if a disruption impacts (or not) each arc probed, before planning their route/path.

We consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of arcs. The nominal cost for arc $a \in \mathcal{A}$ is denoted by c_a and when a disruption impacts arc a there is a cost increase of d_a units. Let x be a vector of variables corresponding to the flow on arcs in $\mathcal{A}$, let s be the source node, t be the destination node, and let $\gamma^+(u)/\gamma^-(u)$ be the arcs directed out of/into node u, respectively. The set $\mathcal{X}$ of feasible solutions for this problem is given by:

$$\sum_{a \in \gamma^+(s)} x_a = \sum_{a \in \gamma^-(t)} x_a = 1 \tag{45a}$$

$$\sum_{a \in \gamma^+(u)} x_a = \sum_{a \in \gamma^-(u)} x_a \qquad \forall u \in \mathcal{V} \setminus \{s, t\} \tag{45b}$$

$$x \in \{0,1\}^{|\mathcal{A}|}. \tag{45c}$$

We use a subset of problem instances from Lozano and Smith (2017a), which are in turn based on a grid network structure commonly used in the literature (Israeli and Wood 2002, Cappanera and Scaparra 2011). These networks have nodes arranged in a grid of m rows and n columns. We consider networks of sizes 5×5 (27 nodes and 86 arcs) and 10×10 (102 nodes and 416 arcs) to test the exact value function approach and networks of size 20×20 (402 nodes and 1826 arcs) and 30×30 (902 nodes and 4236 arcs) to test the heuristic approaches. For each network size there are 10 different problem instances for which the values of c_a and d_a are randomly generated between $[1, 100]$ and for the probing budget we consider values of B to be roughly in $\{0.05|\mathcal{A}|, 0.1|\mathcal{A}|, 0.2|\mathcal{A}|\}$.

We denote by $\hat{\Xi} \subseteq \{0,1\}^{|\mathcal{A}|}$, the set of scenarios of the SAA, and consider problem configurations with $|\hat{\Xi}| \in \{10, 30, 50\}$ to test the exact approach and $|\hat{\Xi}| \in \{100, 500\}$ to test the heuristics. To generate the scenarios in $\hat{\Xi}$, we set the probability of disruptions $p_a = 0.5$ for every arc. As a result, for each scenario $\xi \in \hat{\Xi}$ and each arc $a \in \mathcal{A}$, we randomly set $\xi_a = 1$ with 50% chance or $\xi_a = 0$ with 50% chance. Since we are using SAA, we set $\pi_\xi = \frac{1}{|\hat{\Xi}|}$ for all $\xi \in \hat{\Xi}$.

Table 1 shows the result of our experiments for the exact approach. The first column presents the grid size. The second and third columns show the number of nodes and arcs in the network. The fourth column shows the number of scenarios. The fifth column presents the average optimal expected actual cost with no probing (Γ_0) and the sixth column shows the average optimal expected actual cost with full information (Γ_N), where $N = |\mathcal{A}|$ for this problem. Columns seven and eight show the average bound on the minimum budget needed to obtain an optimal value equal to Γ_N (denoted by $\hat{B}$) and the value of the budget constraint (denoted by B). Columns nine and ten show the average and maximum full information gap (FIG) computed as

$$FIG = \frac{\Gamma_0 - \Gamma_N}{\Gamma_0}. \tag{46}$$

Columns 11 to 15 present for the value function approach the average CPU time, the average lower and upper bounds obtained within the time limit, the average optimality gap, and the number of instances solved to optimality.

We are also interested in measuring the value of information and the price of not having full information. The last two columns of 1 present two such measures. The first one is a standardized measure of the value of information and is computed as

$$\text{Probing Value} = \frac{\Gamma_0 - \Gamma_B}{\Gamma_0}. \tag{47}$$

The second measure standardizes the price of not having full information and is given by

$$\text{Price Gap} = \frac{\Gamma_B - \Gamma_N}{\Gamma_B}. \tag{48}$$

For instances not solved to optimality, we use the best solution available as a proxy for Γ_B in the calculation of the performance measures, which means that the computations of our performance measures are approximate: the probing values we obtain are a lower bound on the true value whereas the price gap values we obtain provide an upper bound for the true value. Each row in Table 1 summarizes the results for 10 different problem instances.

The full information gap is on average 23%, with values as high as 34%, which indicates that there are considerable potential improvement to be achieved by probing. Moreover, the bound on the minimum budget needed to obtain an optimal value equal to Γ_N is consistently less than 50% of the total number of arcs, with values ranging between roughly 16% to 49% depending on the number of scenarios, which shows that for this dataset probing a relatively small fraction of the total arcs is often sufficient to match the performance of perfect information. This result is also supported by the price gap measure, which shows that for most instances probing only 20% of the arcs yields objective values within 3% of Γ_N and in some cases produces solutions with objective equal to Γ_N. Regarding the value of information, probing leads to an 18% average improvement to the objective value with respect to Γ_0, depending on the available budget. We remark that probing only 5% of the arcs achieves improvements of up to 23% in some instances, and the only case in which the probing value is low (1%) corresponds to a set of instances that are not solved to optimality by the exact algorithm, and for which the average optimality gap is 24%, suggesting that for these instances our method fails to obtain high-quality probing plans. Our main take away from the probing value measure is that probing a relatively small fraction of arcs often results in considerable improvements to the objective value.

Table 1 Assessing the performance of the value function exact approach on shortest path problems

Grid	$\|\mathcal{V}\|$	$\|\mathcal{A}\|$	$\|\hat{\Xi}\|$	Γ_0	Γ_N	$\hat{B}$	B	*FIG*		Value Function						
								Avg	**Max**	**Time (s)**	**LB**	**UB**	**Gap**	**# Sol**	**Probing Value**	**Price Gap**
				262.2	208.7		4	20%	34%	19	221.3	221.3	0%	10	16%	6%
			10	262.2	208.7	24.6	8	20%	34%	21	213.3	213.3	0%	10	19%	2%
				262.2	208.7		16	20%	34%	7	209.1	209.1	0%	10	20%	0.2%
				266.2	208.9		4	21%	29%	821	230.4	230.4	0%	10	13%	9%
5x5	27	86	30	266.2	208.9	34.8	8	21%	29%	1178	218.9	219.9	0%	9	17%	5%
				266.2	208.9		16	21%	29%	572	211.2	211.7	0%	9	20%	1%
				268.8	211.2		4	21%	29%	2069	234.7	235.7	0%	8	12%	10%
			50	268.8	211.2	41.9	8	21%	29%	2254	222.6	225.3	1%	6	16%	6%
				268.8	211.2		16	21%	29%	2290	215.0	216.9	1%	7	19%	3%
				403.0	303.0		20	25%	30%	1182	306.9	310.6	1%	8	23%	2%
			10	403.0	303.0	65.0	40	25%	30%	13	303.0	303.0	0%	10	25%	0%
				403.0	303.0		80	25%	30%	9	303.0	303.0	0%	10	25%	0%
				412.9	305.3		20	26%	30%	3600	309.3	359.6	14%	0	13%	15%
10x10	102	416	30	412.9	305.3	103.9	40	26%	30%	3308	306.5	332.2	8%	1	20%	8%
				412.9	305.3		80	26%	30%	292	305.3	305.3	0%	10	26%	0%
				414.6	303.5		20	27%	31%	3600	306.8	408.4	24%	0	1%	26%
			50	414.6	303.5	125.7	40	27%	31%	3600	304.4	344.7	12%	0	17%	12%
				414.6	303.5		80	27%	31%	3600	303.8	310.7	2%	1	25%	2%
							Total	23%	31%	1580	262.5	275.6	4%	119	18%	6%

Regarding the computational performance of our exact algorithm, Table 1 shows that we solve 119 out of the 180 instances in this dataset to optimality within the time limit. The solution time is highly dependent on the number of scenarios and the budget. Instances with higher budgets seem to be solved faster than instances with tight budgets and the solution times increase considerably for instances with more scenarios. The average optimality gap is 4% with values ranging from 1% to 24%, where the worst optimality gaps correspond to instances with tight budget, reinforcing the idea that instances with low budgets tend to be harder to solve.

We now turn our attention to the heuristic approaches. We refer to the randomized greedy heuristic as `H1`, to the heuristic based on a perfect information solution as `H2`, and to the heuristic based on minimizing the estimated cost function as `H3`.

We first use the lower bounds obtained with the value function approach for the 5×5 and 10×10 instances to assess the performance of the heuristics in terms of optimality gap. We find that for these problem instances `H1` finds solutions within 13% of the lower bound on average, `H2` finds solutions within 6% of the lower bound, and `H3` is the best performer on average finding solutions within 4% of the lower bound. The detailed results of this experiment are reported in Table 7 in Appendix C.

We also compare the performance of the heuristic approaches over the larger problem instances 20×20 and 30×30. Table 2 reports the results of this experiment. Column "Obj" shows the average objective function value for the best solution found be each heuristic. The remaining columns present the same information as before, where each row summarizes the results for 10 different problem instances. Bold numbers in the objective column indicate the best performing heuristic for each row and are used to compute the average probing value and price gap. In this experiment we used a small subset of the scenarios in $\hat{\Xi}$ when minimizing the estimated cost function for `H3` since solving the resulting discrete quadratic problem becomes too computationally taxing for large values of $|\hat{\Xi}|$.

Table 2 Assessing the performance of the heuristic approaches on shortest path problems

Grid	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\hat{\Xi}\rvert$	Γ_0	Γ_N	$\hat{B}$	B	FIG		H1		H2		H3		Probing Value	Price Gap
								Avg	**Max**	**Time (s)**	**Obj**	**Time (s)**	**Obj**	**Time (s)**	**Obj**		
				740.8	512.9		90	31%	34%	61	680.6	89	589.2	127	**588.2**	21%	13%
			100	740.8	512.9	452.5	180	31%	34%	104	643.6	147	**564.8**	148	570.9	24%	9%
20×20	402	1826		740.8	512.9		360	31%	34%	130	592.9	183	**550.1**	177	570.1	26%	7%
				741.2	511.6		90	31%	34%	364	685.2	650	594.3	514	**592.7**	20%	14%
			500	741.2	511.6	680.5	180	31%	34%	245	645.3	396	**566.4**	308	574.5	24%	10%
				741.2	511.6		360	31%	34%	228	596.4	355	**553.1**	275	567.4	25%	8%
				1095.2	750.6		200	31%	36%	103	1005.2	184	**868.6**	261	875.9	21%	14%
			100	1095.2	750.6	879.1	400	31%	36%	150	949.9	260	**832.3**	255	855.4	24%	10%
				1095.2	750.6		800	31%	36%	160	882.6	278	**813.1**	265	854.6	26%	8%
30×30	902	4236		1096.7	749.6		200	32%	36%	461	1010.5	898	**870.0**	625	888.0	21%	14%
			500	1096.7	749.6	1418.6	400	32%	36%	523	958.9	1074	**833.5**	681	869.2	24%	10%
				1096.7	749.6		800	32%	36%	499	886.2	1058	**814.9**	642	865.4	26%	8%
							Total	31%	35%	252	794.8	464	704.2	356	722.7	23%	10%

The full information gap for these larger problem instances is on average 31%, with values as high as 36%, which is larger than for the smaller networks. The bound for the amount of budged needed to achieve Γ_N ranges from roughly 21% to 33% of the total number of arcs, which again shows that for this problem probing a relatively small fraction of the arcs often guarantees the same performance as having full information. This idea is again reinforced by the price gap, which is on average 10%, indicating that we are able to find solutions with objective value within 10% of Γ_N on average with the proposed heuristics. We remark that optimal probing plans would probably yield even lower values for the price gap. Regarding the value of information, the probing value measure is consistently above 20% even for small budgets of roughly 5% of the total number of arcs. This again shows that probing a small fraction of arcs can lead to considerable improvements to the objective value even when using our proposed heuristics that do not guarantee an optimal probing plan.

In terms of the computational performance, `H1` is the fastest with an average CPU time of 252 seconds, followed by `H3` with 356 seconds, and by `H2` with 464 seconds. Regarding solution quality, `H2` is clearly the best performer finding the best solutions in 10 out of 12 instance configurations, outperforming `H1` on average by 11% and `H3` by 3%.

6.2. Results for a project selection problem with disruptions and probing

We also study a project selection problem in which projects are susceptible to uncertain failures that reduce their profitability. Before deciding their selection, the decision maker is able to probe a limited set of projects, revealing if a failure impacts (or not) each project probed with the objective of maximizing the expected actual profit.

We consider a set of projects $\mathcal{N} = \{1, \dots, N\}$, each project $i \in \mathcal{N}$ has a nominal profit denoted by c_i and an investment cost denoted by w_i. When a project fails, its profit is reduced to 0 and the decision maker has a total fund of W to invest among the projects. Let x_i be a binary variable that takes the value of 1 if project i is selected and a value of 0, otherwise. The set $\mathcal{X}$ of feasible solutions for this problem is given by a single knapsack constraint:

$$\sum_{i \in \mathcal{N}} w_i x_i \leq W \tag{49a}$$

$$x \in \{0,1\}^N. \tag{49b}$$

We generate a set of synthetic instances as follows. We consider a number of projects $N \in \{20, 30, 40\}$ to test the exact approach and $N \in \{100, 200\}$ to test the heuristics. For each project we draw coefficients w_i independently at random from a discrete uniform distribution $U(1, 50)$ and coefficients c_i from a discrete uniform distribution $U(50, 100)$. We then set the available funds $W = 0.1 \sum_{i \in \mathcal{N}} w_i$. We generate 10 different problem instances for each value of N considered.

Similar to our first problem, we consider scenario configurations with $|\hat{\Xi}| \in \{10, 30, 50\}$ to test the exact approach and $|\hat{\Xi}| \in \{100, 500\}$ to test the heuristics. To generate the scenarios in $\hat{\Xi}$, we set the probability of failure $p_i = 0.5$ for every project and randomly set $\xi_i = 1$ with 50% chance or $\xi_i = 0$ with 50% chance.

Table 3 shows the results of our experiments for the exact approach. The first column shows the number of projects and the remaining columns display the same metrics as Table 1 adjusted to reflect the fact that the project selection objective is to maximize the expected profit (as opposed to minimize the cost). As before, each line summarizes the results for 10 different problem instances.

The gap between Γ_N and Γ_0 is on average 36%, which shows that in this problem setting there are also considerable potential improvements to be achieved by probing. The bounds on the minimum budget needed to obtain an optimal value equal to Γ_N are slightly larger than for the shortest path problem, with values ranging from roughly 50% to 65% of the total number of projects. We believe that this is due to the lack of structure connecting the different projects, opposed to the shortest path setting in which arcs interdependent because of the underlying network structure. However, it is still holds for these problem instances that the performance of perfect information can often be achieved without having to probe every single project (but about 60% of the projects). The price gap shows that probing a small fraction of the projects yields on average objective values within 15% of Γ_N and in some cases produces solutions with objective within less than 5% of Γ_N. Regarding the value of information, probing leads to a 15% average improvement to the objective value with respect to Γ_0, depending on the available budget, with values reaching as high as 32%. We conclude that for this project selection problem it is also true that probing a relatively small fraction of the projects can lead to considerable improvements to the objective value.

Table 3 **Assessing the performance of the value function approach on knapsack problems**

N	$\|\hat{\Xi}\|$	$\hat{B}$	B	Γ_0	Γ_N	Value Function						
						Time (s)	LB	UB	Gap	# Sol	Probing Value	Price Gap
			1	226.9	297.1	2	250.2	250.2	0%	10	10%	16%
	10	11.1	3	226.9	297.1	7	274.2	274.2	0%	10	21%	8%
			6	226.9	297.1	5	292.9	292.9	0%	10	29%	1%
			1	220.7	300.1	15	243.9	243.9	0%	10	10%	19%
20	30	12.2	3	220.7	300.1	46	270.2	270.2	0%	10	22%	10%
			6	220.7	300.1	123	290.4	290.4	0%	10	32%	3%
			1	219.0	299.2	34	240.5	240.5	0%	10	10%	20%
	50	13	3	219.0	299.2	128	266.7	266.7	0%	10	22%	11%
			6	219.0	299.2	345	287.6	287.6	0%	10	31%	4%
			1	361.6	484.6	7	387.2	387.2	0%	10	7%	20%
	10	16.4	3	361.6	484.6	90	423.1	423.1	0%	10	17%	13%
			6	361.6	484.6	1598	453.0	454.3	0%	8	25%	7%
			1	353.5	485.2	72	377.7	377.7	0%	10	7%	22%
30	30	18.4	3	353.5	485.2	2271	409.3	409.3	0%	10	16%	16%
			6	353.5	485.2	3600	437.5	451.7	3%	2	24%	10%
			1	349.3	482.5	592	369.1	369.1	0%	10	6%	24%
	50	19.2	3	349.3	482.5	3579	398.4	414.7	4%	2	14%	17%
			6	349.3	482.5	3600	424.5	463.5	8%	0	22%	12%
			1	480.6	643.3	39	506.8	506.8	0%	10	5%	21%
	10	21.7	3	480.6	643.3	1871	543.8	544.3	0%	10	13%	15%
			6	480.6	643.3	3601	568.4	605.8	6%	0	18%	12%
			1	469.0	648.1	1665	493.0	493.0	0%	10	5%	24%
40	30	24	3	469.0	648.1	3600	521.9	567.9	8%	0	11%	19%
			6	469.0	648.1	3600	548.4	628.3	13%	0	17%	15%
			1	465.6	646.5	3549	486.2	507.4	4%	5	4%	25%
	50	24.8	3	465.6	646.5	3600	493.1	591.7	17%	0	6%	24%
			6	465.6	646.5	3600	503.7	636.0	21%	0	8%	22%
					Total	1527	398.6	416.6	3%	187	15%	15%

In terms of the computational performance, our proposed exact algorithm solves 187 out of the 270 instances in this dataset to optimality within the time limit. The solution time is again highly dependent on the number of scenarios and the budget. Contrary to the shortest path problem, instances with higher budgets seem to be more challenging than instances with tight budgets (see for example instances with $N = 30$ and 10 scenarios). As before, solution times increase considerably for instances with more scenarios. The average optimality gap is 3% with values as high as 21%, where the worst optimality gaps correspond to instances with larger budget values.

We run two sets of experiments with the heuristic approaches. The first one compares the best solutions obtained by the heuristics against the upper bound from the value function approach. For this problem class, `H1` finds solutions within 7% of the upper bound on average, `H2` finds solutions within 5% of the bound, and `H3` is the best performer on average finding solutions within 3% of the upper bound. The detailed results of this experiment are reported in Table 8 in Appendix C.

Our second experiment compares the performance of the heuristics over the larger problem instances. Table 4 reports the results of this experiment. As before, where each row summarizes the results for 10 different problem instances.

The full information gap for these larger problem instances is on average 29%, with values as high as 32%. The bound for the amount of budged needed to achieve Γ_N is consistently under 60% of the total number of projects. The price gap is on average 18% and for many instances we are able to find solutions with objective value within 11% of Γ_N with the proposed heuristics. The probing value measure is on average 17% and even for small budgets of roughly 5% of the total number of projects, the heuristics are able to find solutions that are almost 10% better than Γ_0.

In terms of the computational performance, `H1` is again the fastest with an average CPU time of 301 seconds, followed by `H2` with 200 seconds, and by `H3` with 419 seconds. Regarding solution quality, this time `H3` is clearly the best performer finding the best solutions for all instance configurations and outperforming `H1` on average by 6% and `H2` by 5%.

Table 4 Assessing the performance of the heuristics on knapsack problems

N	$\lvert\hat{\Xi}\rvert$	Γ_0	Γ_N	$\hat{B}$	B	FIG		H1		H2		H3		Probing Value	Price Gap
						Avg	**Max**	**Time (s)**	**Obj**	**Time (s)**	**Obj**	**Time (s)**	**Obj**		
		1175.7	1654.5		5	29%	32%	9	1240.4	9	1240.0	17	**1278.4**	9%	23%
	100	1175.7	1654.5	57.9	10	29%	32%	35	1279.9	27	1293.8	46	**1351.4**	15%	18%
100		1175.7	1654.5		20	29%	32%	48	1354.3	35	1403.7	56	**1472.7**	25%	11%
		1174.9	1657.5		5	29%	32%	26	1240.0	58	1242.5	116	**1273.3**	8%	23%
	500	1174.9	1657.5	61.3	10	29%	32%	515	1282.0	340	1294.0	592	**1350.8**	15%	19%
		1174.9	1657.5		20	29%	32%	927	1357.3	556	1397.1	1044	**1469.7**	25%	11%
		2397.6	3398.0		10	29%	31%	23	2522.2	24	2519.0	57	**2607.0**	9%	23%
	100	2397.6	3398.0	105.9	20	29%	31%	40	2622.5	33	2631.0	77	**2773.8**	16%	18%
200		2397.6	3398.0		40	29%	31%	53	2775.8	38	2868.5	93	**3032.9**	26%	11%
		2368.8	3386.7		10	30%	32%	325	2500.5	246	2496.0	579	**2577.4**	9%	24%
	500	2368.8	3386.7	110.6	20	30%	32%	680	2588.0	441	2611.8	1005	**2747.6**	16%	19%
		2368.8	3386.7		40	30%	32%	933	2754.0	592	2845.2	1342	**3015.2**	27%	11%
						29%	32%	301	1959.7	200	1986.9	419	2079.2	17%	18%

6.3. Additional experiments

We consider smaller instances of the project selection problem with $N = 10$ projects in order to solve them with all possible scenarios rather than using SAA. The objective of this experiment is to check if using all scenarios, instead of using SAA, leads to similar conclusions. In this case, $|\Xi| = 1024$ scenarios and we solve 40 instances with different probing budgets. The results are in Table 6. For these experiments, the average and maximum FIG (independently of the budget) are 36% and 51%, respectively. Also, the upper bound $\hat{B}$ is 5.2 and all 40 instances are solved to optimality.

Table 5 **Solving small knapsack problems with all the scenarios**

B	Γ_0	Γ_N	**Value Function**			
			Time (s)	Γ_B	**Probing Value**	**Price Gap**
1	90.1	120.6	615	106.2	18%	12%
2	90.1	120.6	1038	114.3	27%	5%
3	90.1	120.6	334	118.0	31%	2%
4	90.1	120.6	192	119.9	33%	1%

The results show that similar conclusions are obtained when one uses all scenarios rather than SAA. We got values for the FIG and $\hat{B}/N$ of around 30% and 50%, respectively, which are comparable to the corresponding values in Tables 3 and 4. The average probing value is 27%, which is larger than the values obtained in the previous experiments; however, in this case B is proportionally larger than the other experiments (in Tables 3 and 4 the largest budget represented at most 20% of the number of projects, whereas here the largest budget represents 50% of the projects), which explains the increase. The average price gap is 5%, which is smaller than in the previous experiments, which can again be explained by the larger proportion of budget available in this experiment. In conclusion, the results of this experiment give evidence to suggest that using all scenarios rather than SAA does not result in a significant different performance.

Using the same small instances we next evaluate the tightness of the bounds given in Section 4; specifically, that $\Gamma_0 - \Gamma_B \leq (1/4)\sum_{i\in\mathcal{N}}(c_i' - c_i)$ (referred to as Bound 1) and that $\Gamma_B - \Gamma_N \leq (1/2)\sum_{i\in\mathcal{N}\setminus\mathcal{B}}(c_i' - c_i)$ (referred as Bound 2), see Equations (42) and (43). These values are shown in Table 6.

The results in Table 6 show that the theoretical bounds, at least in these instances, are fairly loose, being several times larger than the true value. This suggest that the bounds, whereas tight in general (as shown by Remark 1), might be very loose depending

Table 6 Evaluating the quality of the bounds on small knapsack problems with all the scenarios

B	$\Gamma_B - \Gamma_0$	Bound 1	$\Gamma_N - \Gamma_B$	Bound 2
1	16.1	189.3	14.4	337.7
2	24.2	189.3	6.3	299.0
3	27.9	189.3	2.6	257.1
4	29.8	189.3	0.7	221.7
Total	**24.5**	**189.3**	**6.0**	**278.9**

on the instance type and data. Consequently, tighter problem-dependent bounds might be available. For instance, to derive tighter bounds in this class of problems, one might use the fact that the variables in $\mathcal{X}$ are subject to a budget constraint.

7. Conclusions

We study a class of combinatorial problems subject to uncertain disruptions, in which the decision maker has the ability to gather information (probing) to confirm the occurrence or absence of disruptions before solving the combinatorial problem. The main focus of our work is on measuring the value of the information provided by the probing stage.

We represent the problem as a bilevel problem with multiple followers and provide an exact approach as well as three heuristic approaches. To the best of our knowledge, we are the first ones to contribute exact approaches for this type of probing problems. We complete our contributions with bounds on the value of information.

We conduct computational experiments on two problem classes for which the underlying problems are a shortest path problem and a knapsack problem. Our computations suggest that even small probing budgets could yield considerable improvements in solution quality when compared to not doing any probing. This is true not only for the exact approach but also for the heuristics, which are able to find considerably better solutions when probing is allowed, compared to the baseline in which there is no probing.

Future research includes developing specialized exact algorithms and acceleration techniques to tackle larger problem instances. Another research venue is applying our modeling framework to problems stemming from domain areas such as defense, surveillance operations, or humanitarian logistics, in which uncertain disruptions are likely to occur and probing could play a major role in improving solution quality.

Acknowledgments

The authors gratefully acknowledge the support of the *Air Force Office of Scientific Research* under grant FA9550-22-1-0236. The second author gratefully acknowledges the support of the *National Science Foundation* under grant CMMI 2145553.

References

Adamczyk M, Sviridenko M, Ward J (2016) Submodular stochastic probing on matroids. *Mathematics of Operations Research* 41(3):1022–1038.

Alizadeh SM, Marcotte P, Savard G (2013) Two-stage stochastic bilevel programming over a transportation network. *Transportation Research Part B: Methodological* 58:92–105.

Artstein Z (1994) Probing for information in two-stage stochastic programming and the associated consistency. *Asymptotic Statistics: Proceedings of the Fifth Prague Symposium, held from September 4–9, 1993*, 21–33 (Springer).

Artstein Z (1999) Gains and costs of informationin stochastic programming. *Annals of operations research* 85(0):129–152.

Artstein Z, Wets RJ (1993) Sensors and information in optimization under stochastic uncertainty. *Mathematics of Operations Research* 18(3):523–547.

Beck Y, Ljubić I, Schmidt M (2023) A survey on bilevel optimization under uncertainty. *European Journal of Operational Research* .

Cappanera P, Scaparra MP (2011) Optimal allocation of protective resources in shortest-path networks. *Transportation Science* 45(1):64–80.

Caprara A, Carvalho M, Lodi A, Woeginger GJ (2013) A complexity and approximability study of the bilevel knapsack problem. *Integer Programming and Combinatorial Optimization: 16th International Conference, IPCO 2013, Valparaíso, Chile, March 18-20, 2013. Proceedings 16*, 98–109 (Springer).

Cormican KJ, Morton DP, Wood RK (1998) Stochastic network interdiction. *Operations Research* 46(2):184–197.

Dempe S (2002) *Foundations of Bilevel Programming* (Boston: Kluwer Academic Publishers).

DeNegre S, Ralphs TK (2009) A Branch-and-Cut algorithm for Bilevel Integer Programming. *Proceedings of the Eleventh INFORMS Computing Society Meeting*, 65–78 (Charleston, SC).

Fischetti M, Ljubić I, Monaci M, Sinnl M (2017) A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research* 65(6):1615–1637.

Goel A, Guha S, Munagala K (2010) How to probe for an extreme value. *ACM Transactions on Algorithms (TALG)* 7(1):1–20.

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

Guha S, Munagala K (2012) Adaptive uncertainty resolution in bayesian combinatorial optimization problems. *ACM Transactions on Algorithms (TALG)* 8(1):1–23.

Gümüş ZH, Floudas CA (2005) Global optimization of mixed-integer bilevel programming problems. *Computational Management Science* 2:181–212.

Gupta A, Nagarajan V (2013) A stochastic probing problem with applications. *Integer Programming and Combinatorial Optimization: 16th International Conference, IPCO 2013, Valparaíso, Chile, March 18-20, 2013. Proceedings 16*, 205–216 (Springer).

Gupta A, Nagarajan V, Singla S (2016) Algorithms and adaptivity gaps for stochastic probing. *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, 1731–1747 (SIAM).

Hellemo L, Barton PI, Tomasgard A (2018) Decision-dependent probabilities in stochastic programs with recourse. *Computational Management Science* 15:369–395.

Israeli E, Wood RK (2002) Shortest-path network interdiction. *Networks* 40(2):97–111.

Janjarassuk U, Linderoth J (2008) Reformulation and sampling to solve a stochastic network interdiction problem. *Networks* 52(3):120–132.

Jonsbråten TW, Wets RJ, Woodruff DL (1998) A class of stochastic programs withdecision dependent random elements. *Annals of Operations Research* 82(0):83–106.

Kleinert T, Labbé M, Ljubić I, Schmidt M (2021) A survey on mixed-integer programming techniques in bilevel optimization. *EURO Journal on Computational Optimization* 9:100007.

Kleywegt AJ, Shapiro A, Homem-de Mello T (2002) The sample average approximation method for stochastic discrete optimization. *SIAM Journal on optimization* 12(2):479–502.

Lozano L, Smith JC (2017a) A backward sampling framework for interdiction problems with fortification. *INFORMS Journal on Computing* 29(1):123–139.

Lozano L, Smith JC (2017b) A value-function-based exact approach for the bilevel mixed-integer programming problem. *Operations Research* 65(3):768–786.

Mitsos A (2010) Global solution of nonlinear mixed-integer bilevel programs. *Journal of Global Optimization* 47(4):557–582.

Özaltın OY, Prokopyev OA, Schaefer AJ (2018) Optimal design of the seasonal influenza vaccine with manufacturing autonomy. *INFORMS Journal on Computing* 30(2):371–387.

Tahernejad S, Ralphs TK, DeNegre ST (2020) A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. *Mathematical Programming Computation* 12(4):529–568.

Verweij B, Ahmed S, Kleywegt AJ, Nemhauser G, Shapiro A (2003) The sample average approximation method applied to stochastic routing problems: a computational study. *Computational optimization and applications* 24:289–333.

Appendix A: The cost estimate is the conditional expected value

For any probing plan z define $J_z = (J_i : z_i = 1)$ and $\xi_z = (\xi_i : z_i = 1)$. We have that

$$\begin{aligned}\mathbb{E}[C(J,x)|J_z=\xi_z] &= \mathbb{E}\Big[\sum_{i\in\mathcal{N}}\big[c_i(1-J_i)x_i + c_i' J_i x_i\big]|J_z=\xi_z\Big]\\ &= \sum_{i\in\mathcal{N}}\big[c_i(1-\mathbb{E}[J_i|J_z=\xi_z])x_i + c_i'\mathbb{E}[J_i|J_z=\xi_z]x_i\big]. \end{aligned} \tag{50}$$

Observe that if $z_i = 1$ then $\mathbb{E}[J_i|J_z = \xi_z] = \xi_i$ whereas if $z_i = 0$ then the independence of the J_is imply that $\mathbb{E}[J_i|J_z = \xi_z] = \mathbb{E}[J_i] = p_i$. Consequently,

$$\mathbb{E}[C(J,x)|J_z=\xi_z] = \sum_{i\in\mathcal{N},z_i=1}\big[c_i(1-\xi_i)x_i + c_i'\xi_i x_i\big] + \sum_{i\in\mathcal{N},z_i=0}\big[c_i(1-p_i)x_i + c_i' p_i x_i\big],$$

which is precisely Equation (2).

Appendix B: Enforcing the optimistic assumption

To make sure that the optimistic assumption is satisfied we need to make a simple solution check in Line 3 of Algorithm 1.

Let x^1 be the solution obtained from solving the RVF in Line 2 and let x^2 be the solution obtained by solving the lower-level problem in Line 3. For each scenario $\xi \in \Xi$ we check if $\hat{C}(\hat{z},\xi,x^{\xi,1}) = \hat{C}(\hat{z},\xi,x^{\xi,2})$, that is, the solution obtained by solving RVF is an alternative optimal solution to the lower-level problem. If this is the case, we record $\hat{x}^\xi = x^{\xi,1}$; otherwise, we record $\hat{x}^\xi = x^{\xi,2}$.

Doing this ensures that the optimistic assumption is satisfied by the optimal solution obtained at the termination of the algorithm. To show this consider an optimal solution $\bar{z}$ obtained via Algorithm 1, and its corresponding second-stage solution $\bar{x}$ and optimal objective value Γ_B. Assume by contradiction that the optimistic assumption is not satisfied, i.e., there exists an alternative solution x' such that $\hat{C}(\bar{z},\xi,\bar{x}^\xi) = \hat{C}(\bar{z},\xi,x'^\xi)$ for all scenarios, and $C(\xi,x'^\xi) < C(\xi,\bar{x}^\xi)$ for at least one scenario $\xi \in \Xi$. This contradicts that Γ_B is the optimal objective value, because solution x' is a feasible solution to RVF that yields an upper bound strictly lower than Γ_B. In turn, following the update rule described above after solving the lower-level problems for $\bar{z}$ would yield an upper bound strictly lower than Γ_B as well.

Appendix C: Additional tables

Tables 7 show the results of the experiments for the heuristic approaches over the small and medium-sized networks for the shortest path problem. The column “gap” displays the optimality gap measured using the lower bound obtained via the exact value function algorihtm. As before, each row summarizes the results for 10 different problem instances.

Table 8 show the results for the heuristic approaches over the small and medium-sized project selection instances.

Table 7 **Assessing the performance of the heuristic approaches on shortest path problems**

Grid	$\|\mathcal{V}\|$	$\|\mathcal{A}\|$	$\|\hat{\Xi}\|$	B	Heu1		Heu2		Heu3	
					Obj	**Gap**	**Obj**	**Gap**	**Obj**	**Gap**
				4	248.4	11%	235.2	6%	227.8	3%
			10	8	240.1	11%	226.3	6%	220.4	3%
				16	226.8	8%	217.9	4%	213.7	2%
				4	252.5	9%	243.6	5%	234.0	2%
5x5	27	86	30	8	246.5	11%	230.6	5%	223.1	2%
				16	235.2	10%	217.6	3%	215.2	2%
				4	255.2	8%	246.9	5%	236.9	1%
			50	8	250.0	11%	235.0	5%	226.3	2%
				16	238.5	10%	221.3	3%	218.2	1%
				20	373.0	18%	339.8	10%	325.9	6%
			10	40	360.8	16%	324.1	7%	311.1	3%
				80	334.4	9%	313.4	3%	311.1	3%
				20	382.6	19%	351.2	12%	344.7	10%
10x10	102	416	30	40	366.2	16%	332.2	8%	331.0	7%
				80	344.2	11%	319.9	5%	322.2	5%
				20	385.2	20%	355.1	14%	346.0	11%
			50	40	369.7	18%	333.8	9%	329.6	8%
				80	347.5	13%	321.3	5%	321.7	6%
				Total	303.2	13%	281.4	6%	275.5	4%

Table 8 **Assessing the performance of the heuristic approaches on knapsack problems**

n	$\|\hat{\Xi}\|$	B	`H1`		`H2`		`H3`	
			Obj	**Gap**	**Obj**	**Gap**	**Obj**	**Gap**
		1	248.5	1%	249.2	0.4%	248.5	1%
	10	3	256.4	6%	265.0	3%	271.0	1%
		6	269.3	8%	288.2	2%	291.8	0.4%
		1	242.9	0%	243.5	0.2%	243.5	0.2%
20	30	3	252.3	7%	261.7	3%	268.4	1%
		6	266.7	8%	286.8	1%	288.9	1%
		1	240.5	0%	240.5	0%	240.5	0%
	50	3	251.8	6%	259.8	3%	266.0	0%
		6	265.1	8%	283.6	1%	287.0	0%
		1	381.5	1%	385.0	1%	384.6	1%
	10	3	395.2	7%	402.7	5%	418.4	1%
		6	414.3	9%	434.4	4%	443.5	2%
		1	375.1	1%	375.3	1%	375.2	1%
30	30	3	382.7	7%	393.3	4%	405.3	1%
		6	404.5	10%	424.7	6%	438.1	3%
		1	369.0	0%	367.9	0.3%	369.1	0%
	50	3	381.4	8%	385.6	7%	398.6	4%
		6	399.4	14%	421.3	9%	431.7	7%
		1	503.1	1%	499.1	2%	505.4	0.3%
	10	3	510.5	6%	516.7	5%	537.4	1%
		6	530.7	12%	553.8	9%	574.9	5%
		1	491.0	0%	490.4	1%	492.1	0.2%
40	30	3	503.5	11%	511.0	10%	525.1	8%
		6	526.0	16%	540.5	14%	564.0	10%
		1	485.4	4%	485.1	4%	486.0	4%
	50	3	498.8	16%	502.4	15%	519.6	12%
		6	519.2	18%	534.5	16%	558.2	12%
			383.9	7%	392.7	5%	401.2	3%