Optimal Information Discovery in Correlated Environments: Measuring the Value of Information

Jaime Pérez[1], Juan S. Borrero[1], and Leonardo Lozano[2]

[1]Department of Industrial and Management Systems Engineering, University of South Florida,
jaimeyairp@usf.edu, jsborrero@usf.edu

[2]Operations, Business Analytics & Information Systems, University of Cincinnati,
lozanolo@ucmail.uc.edu

Abstract

We study linear combinatorial optimization problems where uncertain disruptions may affect the components (i.e., coefficients) of the objective function. To mitigate the impact of these disruptions, a planner can allocate a limited probing budget to fully reveal the status of selected components, while unprobed components status must be estimated. In contrast to settings assuming independence, we consider correlated disruptions, where the probability distribution of unprobed components depends on the information revealed through probing. Under this partial information, the planner then solves the optimization problem and implements the resulting solution. Afterward, the remaining uncertainty is revealed and the actual reward is realized. The planner's goal is to select a probing plan that maximizes the value of information, namely, the expected actual reward. We show that the best way to estimate the value of unprobed components is by using the conditional expected value. We then reformulate the probing problem as a decision-dependent stochastic mixed-integer bilevel optimization problem with multiple followers and propose two formulations based on the value function reformulation. The first one represents conditional probabilities as a maximum flow problem over an auxiliary graph, and the second one uses the concept of consistency sets where the decision-dependent probability constraints are modeled with auxiliary binary variables. Both reformulations are solved via a customized branch-and-cut algorithm. We present a computational study on small-to-medium size instances of two problems: the longest path problem (central to project management) and the 0-1 knapsack problem (central to project selection). We provide insights regarding which of the formulations is preferred given the structure of the uncertainty set. We demonstrate that little information goes a long way: probing a small fraction of components can lead to considerable gains in the solution quality. Moreover, we demonstrate that accounting for correlations is worthwhile, as it yields substantial improvements in solution quality.

Keywords: Stochastic programming; bilevel programming; value of information; combinatorial optimization.

1 Introduction

Decision-making under uncertainty is at the core of multiple application areas ranging from portfolio optimization to supply chain operations. A prominent approach to deal with these problems is multistage stochastic programming, where decisions are made at each stage based on sequential observations of the uncertain parameters. Traditionally, it has been assumed that the observation of uncertain parameters occurs naturally (e.g., uncertain demand for a good is observed in time) and is not influenced by the decision makers. Recently, these assumptions have been challenged as there is a growing interest in understanding the value of information and how decision makers can influence both the sequence in which uncertain parameters are observed and even their outcomes. We see examples of this line of research in the context of robust optimization with decision-dependent information discovery (Vayanos et al. 2025), decision-dependent uncertainty sets (Lozano and Borrero 2025b), and probing-enhanced stochastic programming (Ma et al. 2024, Quezada et al. 2025), among others.

The main focus of this work is to measure the value of information and decide optimal information discovery strategies in a two-stage stochastic programming setting where uncertain parameters are allowed to be correlated, and as a result, observing the realization of one or more uncertain parameters can potentially reveal information regarding unseen correlated parameters. More specifically we study a class of linear combinatorial optimization problems in which uncertain disruptions affect the coefficients of the objective function as proposed by Lozano and Borrero (2025a). Consider a baseline deterministic combinatorial optimization problem of the form:

$$\max \left\{ \sum_{i \in \mathcal{N}} r_i x_i : x \in \mathcal{X} \subseteq \{0,1\}^N \right\}, \tag{1}$$

where $\mathcal{N} = \{1, \cdots, N\}$, r_i denotes a *nominal reward* associated with the binary variable x_i, and $\mathcal{X} \neq \emptyset$ is the set of feasible solutions. Uncertain disruptions affect the reward coefficients in a component-wise manner, meaning that if a disruption impacts component $i \in \mathcal{N}$, the actual reward decreases from r_i to r'_i, where $r'_i < r_i$.

Uncertain disruptions are described by a set of possible scenarios $\Xi \subseteq \{0,1\}^N$, where a scenario $\xi \in \Xi$ happens with probability π^ξ. For each scenario, $\xi_i = 1$ means that component i is disrupted whereas $\xi_i = 0$ means that component is not. In a first decision stage, a *planner* selects B components to observe, with $0 \leq B \leq N$ given and known. We refer to the action of observing the

realization of an uncertain disruption as *probing*, which confirms whether the observed component is disrupted or not. After the planner observes the disruption status of the probed components, and since only partial information has been collected (corresponding to the probed components), the planner needs to *estimate* any uncertain parameters not revealed by probing in order to solve the combinatorial optimization problem in a second stage. Once a solution plan from the second stage is executed, all uncertainty is revealed and the planner observes the *actual* reward for the solution plan selected. As discussed in Lozano and Borrero (2025a) and as shown below, the probing optimization problem can be formulated as a two-stage stochastic optimization problem where the second stage problem is a bilevel optimization problem.

The objective of the planner is to select the first- and second-stage actions to maximize the expected reward across all scenario realizations. In contrast to standard two-stage optimization models, in this problem the second-stage decisions are made after observing only partial information of the uncertainty (that is, the observed realization of the uncertainty for the probed components). Consequently, at the second stage the planner has to estimate the rewards of the unprobed components by using the information collected by the probes. The potential mismatch between the estimated reward based on the partial information revealed by probing and the actual reward obtained after executing a solution plan is the most challenging aspect in our problem setting and it is crucial in order to understand the value of information and to decide optimal probing strategies.

Lozano and Borrero (2025a) estimate the reward by using the expectation of the actual reward conditioned on the information revealed by probing. However, they adopt the strong simplifying assumption that the component-wise disruptions are independent. That is, they assume that the disruption status of a given component does not have any incidence on the disruption status of the remaining components. We argue that this assumption is not only overly restrictive for many practical applications in which uncertain parameters may not be independent but also ignores the interaction effects of probing multiple components at the same time, which can potentially reveal valuable information about the unseen components because of their correlations. To illustrate our point consider the following example.

Example 1. Consider a project selection problem with four projects: A, B, C, and D. If successful, the projects give a profit of \$84, \$80, \$69, and \$30, respectively; however, each project can fail in which case the profit is zero. The decision-maker (DM) considers four possible future scenarios for the completion/failure of the projects: $\xi^1 = (1, 1, 0, 0)$, $\xi^2 = (0, 0, 0, 1)$, $\xi^3 = (0, 1, 1, 0)$, and $\xi^4 = (1, 0, 1, 0)$, where $\xi^i_j = 1$ only if project j fails in scenario i; $\xi^i_j = 0$ otherwise. For example, in scenario 1 both projects A and B fail whereas projects C and D are completed successfully. The scenarios happen with probabilities $\pi^1 = 0.18$, $\pi^2 = 0.21$, $\pi^3 = 0.31$, and $\pi^4 = 0.3$.

Before deciding to invest, the DM can gather more information about two projects so that the

uncertainty about the completion/failure of the probed projects is removed. If the DM assumes that the failures are independent, then the profit estimate for any unprobed project i is given by $r_i q_i$, where q_i is the probability that project i is completed (independent of the scenario and the probing action). For example, under the independence assumption, the estimated profit of B is 80(0.21+0.3)=40.8 and the estimated profit of D is 30(0.18+0.31+0.3)=26.7. Assuming independence, it can be shown that the best plan is to probe A and C, which results in the optimal solutions for each scenario given under "Independent A, C probed" in Table 1. The expected actual profit associated with this plan is \$103.59.

Table 1: Optimal probing plans and optimal action plans in each scenario assuming correlated failures and independent failures. Estimating without exploiting correlations results in \$17.01 less profit.

Scenario		Correlated: A,B probed					Independent: A,C probed				
$(\xi_A, \xi_B, \xi_C, \xi_D)$	Prob.	x_A	x_B	x_C	x_D	Profit	x_A	x_B	x_C	x_D	Profit
1. *(1,1,0,0)*	0.18	0	0	1	1	99	0	1	1	0	69
2. *(0,0,0,1)*	0.21	1	1	0	0	164	1	0	1	0	155
3. *(0,1,1,0)*	0.31	1	0	0	1	114	1	1	0	0	84
4. *(0,1,0,1)*	0.3	0	1	0	1	110	0	1	0	1	110
						120.6					103.59

By assuming independence, the DM does not exploit the correlation between scenarios; namely, observe that by probing projects A and B the uncertainty about the scenarios is completely disentangled. Indeed, if the DM observes that A and B are failed then the DM can infer that scenario 1 realizes; if A and B are completed then scenario 2 realizes, if A is completed and B failed then scenario 3 realizes, while if A is failed and B completed, scenario 4 realizes. Thus, by probing A and B the DM gathers full information about the uncertainty. The corresponding optimal solution for each scenario in this case is given under "Correlated A, B probed" in Table 1. The expected actual profit associated with this plan is \$120.6.

Consequently, by exploiting the dependence between scenarios, the DM can increase its expected actual profit by 16%. This increase is more substantial if it is consider with respect to the baseline of not probing any item. That is, if the DM does not probe any item, then she decides based solely on the initial probabilities and obtains an expected actual reward of \$84.48. Thus, the increase with respect to the baseline would be (120.6-84.48)/(103.59-84.48) which is 89%. □

As we illustrate with Example 1, assuming that disruptions are independent when in fact they are correlated can result in a significant loss of reward. However, relaxing the assumption of independence makes the problem substantially more difficult to solve. In the presence of correlated disruptions, the reward function involves ratios of decision-dependent probabilities,

namely, conditional probabilities given the information revealed by probing. By contrast, when disruptions are assumed to be independent, the reward function simplifies to a bilinear function. To address these issues, in this paper we propose exact solution approaches for the probing problem with correlated disruptions. In particular, our contributions are the following:

- We settle an open question from related works regarding which function should be used to estimate the reward given the partial information revealed by probing. We show that estimating the rewards via the conditional expectation of the actual reward is indeed the best way to decide solution plans.
- We propose a reformulation of the probing problem in which the conditional probabilities are represented by a maximum flow problem on an auxiliary network. From this representation, we develop a formulation for the probing problem that scales linearly with the size of the auxiliary graph for a fixed number of scenarios and linearly with the number of scenarios for a fixed graph.
- We propose an alternative reformulation of the probing problem using the concept of consistency sets, in which the conditional probabilities are modeled with auxiliary binary variables and linear constraints. This reformulation scales linearly in the number of components, but quadratically with the number of scenarios.
- We propose a solution method for the resulting reformulations via a tailored branch-and-cut algorithm in which relaxations are obtained via scenario decomposition.
- We conduct computational experiments on two classes of combinatorial problems, namely, longest path problems and knapsack problems. Our results show that our methods are able to solve medium-sized problem instances to optimality and that probing a small number of components often leads to considerable gains in expected rewards. Moreover, considering correlations in the scenarios consistently leads to significantly better solutions than assuming independence.

2 Related Literature

Gupta and Nagarajan (2013) introduce a general stochastic probing problem, where elements can only be selected after probing and must be committed to immediately if active. Their formulation captures a wide range of applications, including Bayesian auction design and online matching, by imposing two layers of packing constraints: one on the probed elements (outer constraint) and another on the selected elements (inner constraint); the authors propose approximation algorithms.

Later, Gupta et al. (2016) focus on non-adaptive (a priori fixed probing order satisfying the outer constraints) policies and show that the gap between the best adaptive and non-adaptive policies is small and propose approximation algorithms to find near-optimal non-adaptive strategies. Adamczyk et al. (2016) generalize the objective from linear to monotone submodular functions, thus broadening the applicability of the model to settings that capture diminishing returns. Moreover, they propose approximation algorithms for probing problems under matroid interception constraints. All these related works assume that the random variables to be probed are independent.

Goel et al. (2010) introduce a model-driven optimization framework in which the values of uncertain parameters can be revealed at a cost through selective probing, and the objective is to optimize a function of the observed values under a probing budget. They focus on minimizing the expected minimum of a set of non-negative random variables and show that even this seemingly simple objective is NP-hard. The authors develop greedy algorithms with provable approximation guarantees. Guha and Munagala (2012) formalize a general framework for Bayesian combinatorial optimization with probing, where the decision-maker can selectively probe a subset of random inputs before making a final decision. They develop approximation algorithms for a broad class of objectives and introduce a unifying technique based on the outlier version of the deterministic problem. A key insight of their work is that the benefit of adaptivity in probing is bounded by a constant factor for several fundamental problems, including weighted completion time scheduling, makespan minimization, and k-median clustering. This result provides theoretical justification for using simple nonadaptive probing strategies in complex stochastic settings. Singla (2018) addresses combinatorial optimization under uncertainty, where probing to resolve uncertainty incurs a cost, referred to as the price of information. The goal is to minimize the expected cost plus the total probing cost. They provide approximation algorithms to solve the problem. Again, the value of the elements to be probed is assumed to be independent of each other in these related works.

Regarding exact optimization approaches for decision problems with probing, Lozano and Borrero (2025a) consider a simplified version of our problem in which the scenarios are assumed to be independent. Similarly, Ma et al. (2024) consider probing in the context of two-stage stochastic programming, where probing actions do not necessarily observe the uncertain realizations directly but instead reveal information about a jointly-distributed random vector. They propose a sampling based branch-and-bound solution approach and test it on instances of facility location problem with uncertain demands, in which the customer demands are assumed to be independent. In the context of robust optimization, Paradiso et al. (2024) consider probing actions that reveal components of an uncertain vector for which the probability distribution is unknown but is assumed to belong to a polyhedral uncertainty set.

3 Mathematical Formulation

Let z_i be a binary variable that takes the value 1 if the planner probes component i, zero otherwise, $i \in \mathcal{N}$. Let J_i be the random variable that represents the disruption status of component i. Observe that $J_i \in \{0, 1\}$ for all $i \in \mathcal{N}$, where $J_i = 1$ if component i is disrupted; zero otherwise. Denote by $J = (J_i \colon i \in \mathcal{N})$ the random vector with the status of all components and note that $J \in \Xi$, with Ξ being the given set of scenarios. If scenario $\xi \in \Xi$ happens and the the planner selects the solution plan $x \in \mathcal{X}$, then the planner receives the *actual reward* $R(\xi, x)$, which is given by

$$R(\xi, x) = \sum_{i \in \mathcal{N}} [r_i(1 - \xi_i)x_i + r'_i \xi_i x_i]. \tag{2}$$

However, because the planner only observes the value of ξ_i for the probed components, she cannot compute $R(\xi, x)$ nor decide x in the second stage based on $R(\xi, x)$. Thus, in the second stage the planner selects x by optimizing an *estimated reward* that only has access to the values of ξ_i revealed by probing. Let $\hat{R}(z, \xi, x)$ denote the estimated reward given that the planner selects probing plan z, scenario ξ happens, and solution plan $x \in \mathcal{X}$ is selected. To estimate the reward, we use the expectation of the actual reward conditioned on the information revealed by probing. Therefore,

$$\hat{R}(z, \xi, x) = \mathbb{E}[R(J, x) | J_z = \xi_z] = \sum_{\zeta \in \Xi} R(\xi, x) P[J = \zeta \mid J_z = \xi_z], \tag{3}$$

where $J_z = (J_i \colon z_i = 1)$ and $\xi_z = (\xi_i \colon z_i = 1)$. In other words, J_z (ξ_z) is the subvector of J (ξ) that includes only the probed components. Let $\Xi(z, \xi)$ be the set of scenarios that coincide with ξ in the components probed by z; that is, $\Xi(z, \xi) = \{\zeta \in \Xi \colon \zeta_i = \xi_i \text{ for all } i \text{ such that } z_i = 1\}$. Observe that $\zeta_z = \xi_z$ if and only if $\zeta \in \Xi(z, \xi)$. As a consequence, we have that $P[J = \zeta \mid J_z = \xi_z] = \frac{P[J=\zeta, J_z=\xi_z]}{P[J_z=\xi_z]} = \frac{\sum_{\eta \in \Xi(z,\xi), \eta=\zeta} \pi^\eta}{\sum_{\eta \in \Xi(z,\xi)} \pi^\eta}$ and therefore

$$P[J = \zeta \mid J_z = \xi_z] = \begin{cases} \frac{\pi^\zeta}{\sum_{\eta \in \Xi(z,\xi)} \pi^\eta}, & \text{if } \zeta \in \Xi(z, \xi) \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

After simplifying it follows that

$$\hat{R}(z, \xi, x) = \frac{\sum_{\zeta \in \Xi(z,\xi)} \pi^\zeta R(\zeta, x)}{\sum_{\zeta \in \Xi(z,\xi)} \pi^\zeta}. \tag{5}$$

After using estimated reward $\hat{R}(z, \xi, x)$ to select a plan $x \in \mathcal{X}$, the planner observes the actual reward $R(\xi, x)$, which we remark is independent of the probing plan z. Recall that B, $0 \leq B \leq N$, is the available probing budget in the first-stage decision epoch and define $\mathcal{Z}_B = \{z \in \{0, 1\}^N \colon \sum_{i \in \mathcal{N}} z_i \leq$

$B\}$. We formulate the probing problem as a two-stage stochastic programming problem of the form

$$\Gamma_B = \max_z \Big\{ \mathbb{E}[\mathcal{F}(z, J)] : z \in \mathcal{Z}_B \Big\}, \tag{6}$$

where $\mathbb{E}[\mathcal{F}(z, J)] = \sum_{\xi \in \Xi} \pi^\xi \mathcal{F}(z, \xi)$, and for each scenario $\xi \in \Xi$, the function $\mathcal{F}(z, \xi)$ is the actual reward obtained by the planner if the probing plan in z and the scenario ξ occurs:

$$\mathcal{F}(z, \xi) = R(\xi, x), \text{ where } x \in \arg\max_{x'} \{\hat{R}(z, \xi, x') : x' \in \mathcal{X}\}. \tag{7}$$

The first-stage problem decides the probing plan with the objective of maximizing the expected actual reward $\mathbb{E}[\mathcal{F}(z, J)]$, which is computed by solving the second-stage problem. In the second-stage, for a given probing plan and a realization of the uncertainty, the planner first selects a solution plan that maximizes the estimated reward and then computes the actual reward after executing the selected solution plan. Observe that there might be several solutions in the arg max in (7). It turns out, that all such solutions x induce the same value of $R(z, x)$, see Theorem 1 in Section 3.2. Thus, we assume that in these cases the planner breaks ties by a predetermined deterministic rule.

At this stage, it is worth noticing that the planner could potentially estimate the reward using approaches different from the conditional expectation (3). It turns out, however, that using the conditional expectation to decide results in the highest possible value for the actual expected reward, see Theorem 1 in Section 3.2. In other words, the conditional expected value is the best approach that the planner can use to estimate rewards of the unprobed components.

3.1 Decision-dependent stochastic bilevel reformulations

For the sake of simplifying the notation and analysis, hereafter we assume that $r'_i = 0$ for all $i \in \mathcal{N}$, which implies that any disrupted coefficient gives no reward. Following the standard approach to solving two-stage stochastic optimization problems, we generate a monolithic reformulation of problem (6) by introducing a decision variable x^ξ for each scenario $\xi \in \Xi$, which is the vector of the solution plan under scenario ξ. Let all the solution plans be represented by $x^\Xi = \{x^\xi\}_{\xi \in \Xi}$. Thus, problem (6) can be rewritten as the following bilevel problem:

$$\Gamma_B = \max_{z, x^\Xi} \sum_{\xi \in \Xi} \pi^\xi \sum_{i \in \mathcal{N}} r_i (1 - \xi_i) x_i^\xi \tag{BLa}$$

$$\text{s.t. } x^\xi \in \arg\max \big\{ \hat{R}(z, \xi, x) : x \in \mathcal{X} \big\} \quad \forall \xi \in \Xi \tag{BLb}$$

$$z \in \mathcal{Z}_B;\ x^\xi \in \mathcal{X} \quad \forall \xi \in \Xi. \tag{BLc}$$

An alternative view represents (BLb) using the so-called value function reformulation by imposing that $\hat{R}(z, \xi, x^{\xi}) \geq \hat{R}(z, \xi, x)$ for all $x \in \mathcal{X}$ and all $\xi \in \Xi$. Exploiting the representation of $\hat{R}(z, \xi, x)$ given in (5) and the definition of the actual reward in (2), we obtain the following value function reformulation of problem (6):

$$\Gamma_B = \max_{z, x^{\Xi}} \sum_{\xi \in \Xi} \pi^{\xi} \sum_{i \in \mathcal{N}} r_i (1 - \xi_i) x_i^{\xi} \tag{VFRa}$$

$$\text{s.t.} \sum_{\zeta \in \Xi(z,\xi)} \sum_{i \in \mathcal{N}} \pi^{\zeta} r_i (1 - \zeta_i) x_i^{\xi} \geq \sum_{\zeta \in \Xi(z,\xi)} \sum_{i \in \mathcal{N}} \pi^{\zeta} r_i (1 - \zeta_i) x_i \quad \forall x \in \mathcal{X}; \forall \xi \in \Xi \tag{VFRb}$$

$$z \in \mathcal{Z}_B;\ x^{\xi} \in \mathcal{X} \quad \forall \xi \in \Xi. \tag{VFRc}$$

Observe that (VFR) can be seen as the extended reformulation of a stochastic program with decision-dependent constraints. This formulation has two main difficulties: first, (VFRb) involves as many constraints as elements in $\mathcal{X}$, which in general is of exponential size. Second, constraints (VFRb) sum terms over the decision-dependent set $\Xi(z, \xi)$. In the sections that follow we provide different ways to address these difficulties. Namely, we provide a custom branch-and-cut procedure to deal with $\mathcal{X}$, and provide two different reformulations to deal with the decision-dependent sets $\Xi(z, \xi)$.

3.2 Justification of the approach

We assume that after probing, the planner decides x in terms of the conditional expected value given the information revealed by probing, see (3). A relevant question is whether selecting x in this manner is the best way to decide given the limited information. The answer is yes, which aligns with similar results on the role of conditional expectation as the 'best predictor' of the underlying random variable, see, e.g., chapter 9 of Williams (1991). Next, we provide informal version of this result in the context of our problem; a formal proof can be found in Appendix A.

Definition 1. Let $(\Omega, \mathcal{F}, P)$ be the underlying probability space where $J\colon \Omega \to \Xi$ is a random vector. A decision rule $d\colon \Omega \to \mathcal{X}$ assigns, to each realization of the uncertainty $\omega \in \Omega$, a feasible vector $d(\omega) \in \mathcal{X}$. Given a subset $S \subseteq [n]$, we say that d is adapted to $J_S = (J_i\colon i \in S)$ if and only if, for each $\omega \in \Omega$, d only uses the value of $J_i(\omega)$, $i \in S$, (and does not use the value of $J_i(\omega)$, $i \notin S$), to determine $d(\omega)$.

Fix $\omega \in \Omega$ and let $\xi = J(\omega)$. Observe that if $\hat{d}(\omega) = x^{J(\omega)} = x^{\xi}$, where x^{ξ} is computed as in Equation (BLb), then $\hat{d}$ is a decision rule adapted to $J_z = (J_i\colon z_i = 1)$.

Theorem 1. *Let d be a decision rule that is adapted to J_z and let $\hat{d}$ be the decision rule obtained by solving* (BLb) *on each scenario. Then, $E[R(J, \hat{d})] \geq E[R(J, d)]$.*

Besides being the best way to decide given the information in J_z, the decision rule obtained from (BLb) assures that there is always a non-negative *value of information*. Thus, the information provided by probing more components is always beneficial to improve the actual expected reward. This result also aligns with similar statements of the value of information in other contexts, see e.g., Blackwell (1953). Specifically, we have the following result:

Theorem 2. *Let $w, z \in \{0,1\}^N$ be two probing plans such that $w \leq z$. If $\hat{d}$ is the decision rule obtained by solving* (BLb) *based on J_z on each scenario, and d is a decision rule based in J_w, then $E[R(J,\hat{d})] \geq E[R(J,d)]$.*

4 Maximum flow reformulation

In this section, we introduce an alternative representation of $\tilde{R}(z,\xi,x) = \sum_{\zeta\in\Xi(z,\xi)} \pi^\zeta R(\zeta,x)$ by using a maximum flow problem on an auxiliary network constructed based on the set of scenarios Ξ. Using this representation, we obtain a reformulation of (VFR) with no decision-dependent sets in the constraints. Consider a set of scenarios $\Xi \subseteq \{0,1\}^N$ and recall scenario $\xi \in \Xi$ happens with probability π^ξ. Let $P_k(\Xi)$ be the set of *prefixes of length* k, $1 \leq k \leq N$, that appear in the vectors of Ξ, namely

$$P_k(\Xi) = \big\{(i_1,\ldots,i_k) \in \{0,1\}^k \colon \exists (i_1,\ldots,i_k,i_{k+1},\ldots,i_N) \in \Xi\big\}. \tag{10}$$

Define a directed graph $D = (V,E)$ where the set of vertices is defined as $V = \{s,t\} \cup \bigcup_{k=1}^N P_k(\Xi)$, ($s$ and t being auxiliary vertices) and the set of edges is defined as

$$E = \big\{(s,i_1) \colon i_1 \in P_1(\Xi)\big\} \cup \bigcup_{k=1}^{N-1} \big\{\big((i_1,\ldots,i_k),(i_1,\ldots,i_k,i_{k+1})\big) \colon (i_1,\ldots,i_k,i_{k+1}) \in P_{k+1}(\Xi)\big\}$$

$$\cup \big\{\big((i_1,\ldots,i_N),t\big) \colon (i_1,\ldots,i_N) \in \Xi\big\} \tag{11}$$

An illustration of the resulting network in a small instance is shown in Figure 1.

Next, we define the capacities of the edges in E as functions of z, ξ, and x. Let $M = \mu \cdot \max_{\zeta\in\Xi} R(\zeta,x)$, where $\mu > 0$ is a sufficiently large constant. For the edges $(s,i_1) \in E$ in the first layer of the graph, the capacities are defined by:

$$c_{s,i_1}(z,\xi,x) = M\big((\mathbb{I}\{i_1 = \xi_1\} - 1)z_1 + 1\big). \tag{12}$$

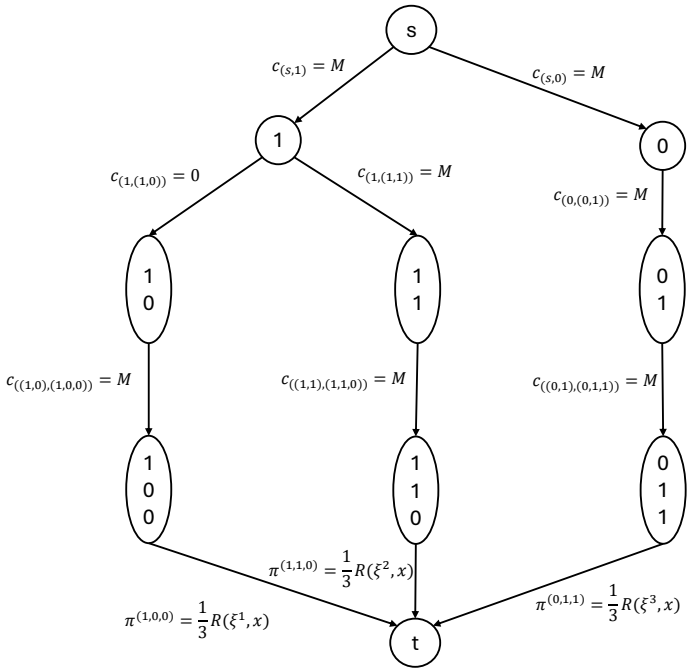


Figure 1: A of graph D with $N = 3$ and $\Xi = \{(1,0,0), (1,1,0), (0,1,1)\}$. The edge capacities are computed assuming that $M = \mu \cdot \max_{\zeta \in \Xi} R(\zeta, x)$ with $\mu = |\Xi|$. The scenario probabilities are assumed to be $\pi^{\xi} = 1/3$ for all scenarios. For computing the capacities $c(z, \xi, x)$ we assume that $z = (0,1,0)$ and that $\xi = (1,1,0)$.

Next, for a given edge $\big((i_1, \ldots, i_k), (i_1, \ldots, i_k, i_{k+1})\big) \in E$ the capacity is:

$$c_{(i_1,\ldots,i_k),(i_1,\ldots,i_k,i_{k+1})}(z, \xi, x) = M\big((\mathbb{I}\{i_{k+1} = \xi_{k+1}\} - 1)z_{k+1} + 1\big). \tag{13}$$

Finally, for an edge $(\zeta, t) \in E$ in the last layer the capacity is

$$c_{\zeta,t}(z, \xi, x) = \pi^{\zeta} R(\zeta, x). \tag{14}$$

In other words, if $k \in \mathcal{N}$ is not probed (i.e., $z_k = 0$), then the capacity of every edge having an endpoint in $P_k(\Xi)$ is M. Else, if $k \in \mathcal{N}$ is probed (i.e., $z_k = 1$), then the capacity of $\big((i_1, \ldots, i_{k-1}), (i_1, \ldots, i_k)\big)$ is M if and only if the 'partial' scenario $(i_1, \ldots, i_k)$ and ξ agree on the status of component k. These definitions imply that $(i_1, \ldots, i_N)$ is equal to ξ in every probed component if and only if there is a path with positive capacity from s to $(i_1, \ldots, i_N)$ in graph D. In fact, the capacity of each arc on that path is precisely M.

As an illustration, consider again Figure 1. Because components 1 and 3 are not probed, then the capacity of the arcs $(s,0)$, $(s,1)$, $\big((1,0),(1,0,0)\big)$, $\big((1,1),(1,1,0)\big)$, and $\big((0,1),(0,0,1)\big)$ are all equal to M. By contrast, because component 2 is probed and $\xi_2 = 1$, only the edges that end in

$(1,1)$ and $(0,1)$ are going to have capacity of M. The edge that ends in $(1,0)$ has a zero capacity because the second component of this 'partial' scenario is different from the value of the second component in scenario ξ, i.e., $\xi_2 = 1 \neq 0$. Note that the fact that the capacity of $\big(1,(1,0)\big)$ is zero implies that no flow can be pushed from s to $(1,0,0)$.

In general, when solving a max flow problem between s and t on the graph D, M units of flow can be pushed from s to $(i_1,\ldots,i_N) \in P_N(\Xi)$ if and only if i_k coincides with ξ_k on every component probed by z. Since the capacity of arc $\big((i_1,\ldots,i_N),t\big)$ is precisely $\pi^\zeta \cdot R(\zeta,x)$ where ζ is scenario $(i_1,\ldots,i_N)$, then the maximum flow that can be pushed between s and t is equal to the sum of $\pi^\zeta \cdot R(\zeta,x)$ of all scenarios in $\zeta \in \Xi$ that are consistent with ξ given z; that is, the max flow between s and t is precisely $\tilde{R}(z,\xi,x)$. More concisely, we have proven the following result.

Proposition 1. *Let $\xi \in \Xi$, $z \in \{0,1\}^N$ and $x \in \mathcal{X}$ be given. Then $\tilde{R}(z,\xi,x)$ is the maximum s-t flow in graph D assuming that the capacities are given by* (12)–(14).

4.1 Tightening the 'big-M' constant

Whereas Proposition 1 remains generally valid for a sufficiently large choice of M, and in particular when $\mu \geq 1$, in practice, having a tight 'big-M' results in stronger MIP relaxations. Next, we derive a procedure to tighten the value of M based on node-dependent bounds that exploit the structure of D. To this end, let $p^{v,q}$ be a partial path in D connecting nodes v and q in V, let $\mathcal{P}$ of be the set of all $v-q$ partial paths in D, and let $\mathcal{R}(v) = \big\{\zeta \in P_N(\Xi) : \exists p^{v,\zeta} \in \mathcal{P}\big\}$ be the set of reachable nodes (scenarios) in $P_N(\Xi) = \Xi$ from a given $v \in V$. We define

$$\rho(v) = \sum_{\zeta \in \mathcal{R}(v)} \pi^\zeta R(\zeta,x); \tag{15}$$

in other words, $\rho(v)$ is the sum of $\pi^\xi R(\xi,x)$ across all scenarios ξ that are reachable from v in D. Intuitively, all scenarios outside $\mathcal{R}(v)$ are incompatible with the partial path leading to node v, and thus they cannot contribute to the reward associated with any continuation from v. Observe that computing (15) requires knowing all $v-t$ paths in D for all $v \in V \setminus \{s\}$, whose complexity is in $O((|V|+|E|)|\Xi|)$ using a dynamic programming algorithm from bottom (node t) to top (node s).

A weaker alternative (although still stronger with respect to the original definition of the 'Big-M' constant) is to use node-specific 'Big-M' constants instead of a global one. To this end, for each scenario $\zeta \in \Xi$, let $\hat{M}^\zeta$ be a valid scenario dependent upper bound satisfying $\hat{M}^\zeta \geq \max_{x\in\mathcal{X}} R(\zeta,x)$, and for each node $v \in V$ define

$$\hat{M}_v = \sum_{\zeta \in \mathcal{R}(v)} \pi^\zeta \hat{M}^\zeta. \tag{16}$$

Next, we can replace M with $\rho(i_1)$ (or with its node-dependent 'big-M' constant) in the definition

of the capacities in equation (12). Similarly, we can replace M with $\rho\big((i_1, \ldots, i_{k+1})\big)$ (or with its node-dependent 'big-M' constant) in the definition of the capacity in equation (13). Note that Proposition 1 remains valid after these changes.

4.2 The maximum flow and min cut formulation

Consider the graph D constructed previously with edge set $\hat{E} = E \cup \{(t, s)\}$), where the artificial edge (t, s) has unlimited capacity and let A denote the incidence matrix of $(V, \hat{E})$. For a solution plan z, a scenario $\xi \in \Xi$ and a solution plan $x \in \mathcal{X}$, we have that, by construction, the value of $\tilde{R}(z, \xi, x)$ can be formulated as a max-flow problem given by

$$\tilde{R}(z, \xi, x) = \max_{y \in \mathcal{Y}(z,\xi,x)} y_{ts} \tag{17}$$

where $\mathcal{Y}(z, \xi, x) = \left\{ y \in \mathbb{R}_+^{|\hat{E}|} \colon Ay = 0,\ y_{uv} \leq c_{uv}(z, \xi, x)\ \forall (u, v) \in E \right\}$ is the set of feasible flows. Next, we write the dual of problem $\tilde{R}(z, \xi, x)$, which is precisely the minimum cut problem. For the flow conservation constraints, we introduce dual variables α_v for each $v \in V$, which can be interpreted as whether a node v belongs to the "source-side" of the cut in the dual formulation. For the capacity constraints, introduce the variables $\lambda_{u,v} \geq 0$ for each edge $(u, v) \in E$, which represent whether arc (u, v) is in the cut. Then, the corresponding dual problem is

$$\tilde{R}(z, \xi, x) = \min_{(\lambda,\alpha) \in \mathcal{D}} \sum_{(u,v) \in E} c_{uv}(z, \xi, x) \lambda_{uv} \tag{18}$$

where $\mathcal{D} = \left\{ (\lambda, \alpha) \in \mathbb{R}_+^{|E|} \times \mathbb{R}^{|V|} \colon \lambda_{uv} \geq \alpha_u - \alpha_v\ \forall (u, v) \in E,\ \alpha_s - \alpha_t \geq 1 \right\}$ denotes the dual feasible region associated with the maximum-flow problem.

4.3 Flow-based value function reformulation

We provide a reformulation that leverages the representation of the function $\tilde{R}$ as a maximum-flow problem. To this end, observe that each the value of each $\tilde{R}$ in problem (VFRb) corresponds to a maximum flow problem or a minimum cut problem defined on a directed acyclic graph constructed from the set of uncertainties Ξ. By replacing each of such terms with its definition, we get the following reformulation, which is correct by Proposition 2 (see the proof in Appendix B.1).

$$\Gamma_B = \max_{z, x^\Xi} \sum_{\xi \in \Xi} \pi^\xi \sum_{i \in \mathcal{N}} r_i (1 - \xi_i) x_i^\xi \tag{19a}$$

$$\text{s.t. } y_{ts}^\xi \geq \sum_{(u,v) \in E} c_{uv}(z, \xi, x) \lambda_{uv}^\xi \qquad \forall x \in \mathcal{X},\ \xi \in \Xi \tag{19b}$$

$$y^{\xi} \in \mathcal{Y}(z, \xi, x^{\xi}) \quad \forall \xi \in \Xi; \quad (\lambda^{\xi}, \alpha^{\xi}) \in \mathcal{D} \quad \forall \xi \in \Xi \tag{19c}$$

$$z \in \mathcal{Z}_B;\ x^{\xi} \in \mathcal{X} \quad \forall \xi \in \Xi \tag{19d}$$

Proposition 2. *Problem* (19) *in an exact single-level reformulation of* (VFR).

Formulation (19) is a mixed-integer nonlinear programming problem. In fact, all the nonlinearities appear in constraints (19b) and (19c). Fix a scenario $\xi \in \Xi$. First, observe that each product in the term $\sum_{(u,v)\in E} c_{uv}(z, \xi, x)\lambda^{\xi}_{uv}$ appearing in constraints in (19b) can be linearized by using the same arguments of the standard McCormick (1976) linearization since we have that $\lambda^{\xi}_{uv} \in [0, 1]$ for each $(u, v) \in E$ and that the decision variable z in the term $c_{uv}(z, \xi, x)$ is binary.

Next, each product term of the form $(z_\ell x^{\xi}_i)$ appearing in the constraints in (19c) is readily linear using a similar procedure as before since both decision variables are binary. Alternatively, such constraints can be reformulated by using the valid 'big-M' constant defined in (16), which in turn removes the dependence of the value of the intermediate capacities of the graph from the variable x^{ξ}; that is, the value of the intermediate capacities of the graph only depends on the variable z. Since the capacities of the last layer only depend on x^{ξ} and not on z, we conclude that the capacity constraints in (19c) are readily linear when using the 'big-M' values in (16). We use the latter reformulation since it showed better performance in preliminary computational experiments, and we will refer to such linearized version of (19) as the flow-based linear value function reformulation (FLVR).

5 Consistency-set based value-function reformulation

We use an alternative representation to deal with terms $\Xi(z, \xi)$ in the constraints in (VFRb). To this end, for each pair of scenarios $\xi, \zeta \in \Xi$, define $U(\xi, \zeta) := \{i \in \mathcal{N} : \xi_i \neq \zeta_i\}$; that is, $U(\xi, \zeta)$ contains the components that distinguish scenarios ξ and ζ. We have the following result (see the proof in Appendix B.2).

Proposition 3. *Fix $\xi, \zeta \in \Xi$. For any $z \in \{0, 1\}^N$ we have that $\zeta \in \Xi(z, \xi)$ if and only if $z_i = 0$ for all $i \in U(\xi, \zeta)$.*

Note that Proposition 3 gives a complementary representation of the consistency condition. Instead of checking whether ξ and ζ agree on every probed component, it checks whether any component that distinguishes such scenarios has been probed. Thus, ζ remains consistent with ξ precisely when none of the distinguishing components in $U(\xi, \zeta)$ are selected by the probing decision. We remark that this representation is useful because the set $U(\xi, \zeta)$ is known in advance for every pair of scenarios. To model this condition, we introduce scenario-consistency variables $\phi^{\xi,\zeta}$, which takes the value of one if and only if $\zeta \in \Xi(z, \xi)$. This relationship can be enforced by

linear constraints using its equivalent representation stated in Proposition 3, from which we obtain the following consistency set based value function reformulation (CS-VFR):

$$\Gamma_B = \max_{z, x^{\Xi}} \sum_{\xi \in \Xi} \pi^{\xi} \sum_{i \in \mathcal{N}} r_i (1 - \xi_i) x_i^{\xi} \tag{20a}$$

$$\text{s.t.} \sum_{\zeta \in \Xi} \pi^{\zeta} \sum_{i \in \mathcal{N}} r_i (1 - \zeta_i) x_i^{\xi} \phi^{\xi,\zeta} \geq \sum_{\zeta \in \Xi} \pi^{\zeta} \sum_{i \in \mathcal{N}} r_i (1 - \zeta_i) x_i \phi^{\xi,\zeta} \quad \forall x \in \mathcal{X}, \forall \xi \in \Xi \tag{20b}$$

$$\phi^{\xi,\zeta} \leq 1 - z_i \quad \forall \zeta, \xi \in \Xi,\ \forall i \in U(\xi, \zeta); \quad \phi^{\xi,\zeta} \geq 1 - \sum_{i \in U(\xi,\zeta)} z_i \quad \forall \xi, \zeta \in \Xi \tag{20c}$$

$$z \in \mathcal{Z}_B;\ x^{\xi} \in \mathcal{X} \quad \forall \xi \in \Xi; \phi^{\xi,\zeta} \in [0,1] \quad \forall \xi, \zeta \in \Xi. \tag{20d}$$

The first set of inequalities in (20c) capture the negation of consistency; i.e., if any component on which ξ and ζ differ is probed, then $\phi^{\xi,\zeta}$ must be zero. The second set of inequalities in (20c) captures the opposite case; i.e., if no distinguishing component is probed, then $\phi^{\xi,\zeta}$ must be one. By linearizing the terms $(x^{\xi}\phi^{\xi,\zeta})$ through the introduction of a decision variable $w^{\xi,\zeta}$ and adding the associated McCormick envelopes for each pair $\xi, \zeta \in \Xi$, we obtain that (20) can be casted as a linear mixed integer optimization problem.

5.1 Comparing the (FLVR) and the (CS-VFR)

In this section we provide a comparison between (FLVR) and (CS-VFR) in terms of the number of variables of each formulation, where we say that one formulation *dominates* the other if it has fewer variables. Let $L = \sum_{k=1}^{n} |P_k(\Xi)|$, which is the number of internal prefix arcs in the auxiliary graph D. Observe that the number of arcs in D is $A := |E| = L + |\Xi|$, where the latter $|\Xi|$ accounts for the last layer connecting the scenarios with the sink node. Similarly, the number of vertices in D is $B := |V| = L + 2$, where the latter term accounts for the source and sink node of the graph. Table 2 summarizes the number of variables, in addition to z and x^{Ξ}, in each formulation.

Table 2: Comparison of the number of additional variables

Variable	Representation	Size	Total
Primal flow	$y \in \mathbb{R}_{+}^{\lvert\hat{E}\rvert \times \lvert\Xi\rvert}$	$\lvert\Xi\rvert(A+1)$	$3\lvert\Xi\rvert^2 + \lvert\Xi\rvert(4L+3)$
Dual flow	$\lambda \in [0,1]^{\lvert E\rvert \times \lvert\Xi\rvert}$, $\alpha \in [0,1]^{\lvert V\rvert \times \lvert\Xi\rvert}$	$\lvert\Xi\rvert A + \lvert\Xi\rvert B$	
Linearization $(\lambda_{uv} z_\ell)$	$\gamma \in [0,1]^{\lvert E\rvert \times \lvert\Xi\rvert}$	$\lvert\Xi\rvert A$	
Consistency	$\phi \in [0,1]^{\lvert\Xi\rvert \times \lvert\Xi\rvert}$	$\lvert\Xi\rvert^2$	$\lvert\Xi\rvert^2(n+1)$
Linearization $(x_i^{\xi}\phi^{\xi,\zeta})$	$w \in [0,1]^{\lvert\Xi\rvert \times \lvert\Xi\rvert \times n}$	$\lvert\Xi\rvert^2 n$	

As an example, suppose that Ξ has all possible combinations between the components states, namely $\Xi = 2^n$. Then, the graph G would have $L = |\Xi|n$ number of internal prefix nodes, and it can be verified that in this case, formulation (CS-VFR) dominates formulation (FLVR). Next, we

show that if L is smaller than a fraction κ of $|\Xi|n$, then (FLVR) dominates (CS-VFR). In particular, we have the following result whose proof is found in Appendix B.2.

Proposition 4. *Let* $\kappa = \frac{L}{|\Xi|n}$. *Then, (FLVR) dominates (CS-VFR) if and only if* $\kappa < \frac{n-2-3/|\Xi|}{4n}$.

Since $\frac{n-2-3/|\Xi|}{4n} \to 1/4$ for large n and $|\Xi|$, we conclude that the max-flow reformulation becomes attractive (in terms of additional variables) when the prefix graph D uses less than approximately one quarter of the internal arcs required by all possible states combinations between components. One of such cases might arise when having highly correlated scenario sets because highly correlated scenarios tend to share long prefixes, which would decrease the value of L. On the other hand, it can be shown that that the ordering of the components in the set $\mathcal{N}$ plays an important role in the size of the size of the (FLVR); see Remark 6 in Appendix B.2.

6 Branch and cut algorithm for the (FLVR)

In this section, we present a branch and cut algorithm that can be used to solve the (FLVR) and the (CS-VFR). For ease of exposition, we describe the algorithm in terms of the (FLVR). The same algorithm applies to the (CS-VFR) after replacing the relevant constraints with their corresponding counterparts.

First, we note that the main challenges in solving the (FLVR) arise from the large number of constraints in (19b) and the presence of a large number of scenarios. To address this, first we define a relaxed value function problem by considering a subset of the second-stage plans $\hat{\mathcal{X}} \subseteq \mathcal{X}$; i.e., this relaxed problem is the (FLVR) in which $\mathcal{X}$ is replaced by $\hat{\mathcal{X}}$. Problem $\Gamma_B(\hat{\mathcal{X}})$ can be solved by means of a scenario-based decomposition approach as follows. Define the set $S(\hat{\mathcal{X}}) = \{(z, x^\Xi) : \text{ Eqs.(19b) – (19d) hold}\}$. In this set, $\mathcal{X}$ is replaced by $\hat{\mathcal{X}}$. Next, by introducing copies $z^{\xi^1}, \cdots, z^{\xi^r}$, where $r = |\Xi|$, of the first stage variables for each scenario, problem $\Gamma_B(\hat{\mathcal{X}})$ can be written as

$$\Gamma_B(\hat{\mathcal{X}}) = \max\left\{\sum_{\xi\in\Xi} \pi^\xi \sum_{i\in\mathcal{N}} r_i(1-\xi_i)x_i^\xi : (z^\xi, x^\Xi) \in S(\hat{\mathcal{X}})\ \forall \xi \in \Xi, z^{\xi^1} = \cdots = z^{\xi^r}\right\}, \tag{21}$$

where the non-anticipativity conditions $z^{\xi^1} = \cdots = z^{\xi^r}$ enforce that the probing plan chosen in the first-stage must remain the same across all scenarios. Let $\hat{\Gamma}_B(\hat{\mathcal{X}})$ define the value of problem (21) in which the non-anticipativity constraints are relaxed, and observe that there are no coupling requirements between the z^ξs in this relaxation. As a result, $\hat{\Gamma}_B(\hat{\mathcal{X}})$ can be solved for each scenario

independently and potentially in parallel as:

$$\hat{\Gamma}_B(\hat{\mathcal{X}}) = \sum_{\xi\in\Xi} \pi^\xi \max\left\{\sum_{i\in\mathcal{N}} r_i(1-\xi_i)x_i^\xi : (z^\xi, x^\xi) \in S^\xi(\hat{\mathcal{X}})\right\}, \tag{22}$$

where $S^\xi(\hat{\mathcal{X}})$ denotes constraints (19b)-(19d) for a fixed $\xi \in \Xi$ where $\mathcal{X}$ is replaced by $\hat{\mathcal{X}}$. We leverage this property through a specialized branch-and-cut (B&C) procedure to solve the (FLVR) to optimality. The two main ingredients of our B&C algorithm are a branch and bound (BB) procedure and and a cutting plane strategy. In the BB the branching is done by fixing the probing variables z to either 0 or 1; the bounding is done by solving relaxations of the form (22), potentially in parallel, assuming the corresponding z variables are fixed. Specifically, for a node N_ℓ of the BB tree let $C_\ell^0, C_\ell^1 \subseteq \mathcal{N}$ the components for which the probing variables z have been set to 0 and 1, respectively. The relaxation solved at N_ℓ is given by

$$\hat{\Gamma}_B^\ell(\hat{\mathcal{X}}) = \sum_{\xi\in\Xi} \pi^\xi \max\left\{\sum_{i\in\mathcal{N}} r_i(1-\xi_i)x_i^\xi : (z^\xi, x^\xi) \in S^\xi(\hat{\mathcal{X}}), z_i^\xi = k\ \forall i \in C_\ell^k, k \in \{0,1\}\right\}, \tag{23}$$

and let $\hat{z}^{\Xi,\ell}$ be the vector that contains the resulting values of the probing variables. Note that node N_0 corresponds to the root node that contains the relaxed problem $\hat{\Gamma}_B(\hat{\mathcal{X}})$; i.e., $\hat{\Gamma}_B^0(\hat{\mathcal{X}}) = \hat{\Gamma}_B(\hat{\mathcal{X}})$ and thus $C_0^0 = C_0^1 = \emptyset$. During the BB procedure, whenever a probing plan $\hat{z}^{\Xi,\ell}$ is found to be feasible in (21), we evaluate if the associated solution plans $\hat{x}^{\Xi,\ell}$ are feasible in the original problem; i.e., if they satisfy the constraints in (BLb). If they do, we can conclude that $(\hat{z}^{\Xi,\ell}, \hat{x}^{\Xi,\ell})$ induces a feasible solution of (FLVR) and update the lower bound. Else, the set $\hat{\mathcal{X}}$ is expanded by adding the solution plans $\hat{x}^{\xi,\ell} \in \mathcal{X}$, $\xi \in \Xi$, for which the constraints (BLb) are violated.

Let $\mathcal{L}$ denote the list of nodes that must still be solved (i.e., those that have neither been pruned nor branched on). Algorithm 1 formalizes our branch-and-cut procedure, which can be initialized with some z^H corresponding to a heuristic solution with value α^H. The cutting part of the algorithm is detailed in Step 7(a), whereas the branching strategy is presented in Step 8. We note that Algorithm 1 finds the optimal solution of (FLVR) in finitely many iterations by standard convergence arguments of BB and cutting-plane algorithms.

We next turn to a discussion of the BB strategy. Without loss of generality, assume that the components are given in a non-increasing order with respect to the actual expected reward $\tilde{r}_i$, where $\tilde{r}_i = r_i \sum_{\xi\in\Xi:\xi_i=0} \pi^\xi$. For a given node N_ℓ of the BB tree let $F_\ell = \mathcal{N} \setminus \{C_\ell^0 \cup C_\ell^1\}$ be the set of components that are not fixed to either zero or one in previous branching decisions. For each component $i \in F_\ell$, we create a child node where component i is set to take the value one and all components of F_ℓ that precede i in the ordering are removed from F_ℓ and set to take the value zero; i.e., a new child node $N_{\ell'}$ is created with $C_{\ell'}^1 = C_\ell^1 \cup \{i\}$, $C_{\ell'}^0 = C_\ell^0 \cup \{i' \in F_\ell : i' < i\}$, and

Algorithm 1 Branch and Cut algorithm

Input: Data of (FLVR) and the set Ξ.

Output: An optimal probing plan $\bar{z}$ for problem Γ_B.

1. Set $\underline{Z} = -\infty$, $\overline{Z}_0 = \infty$, and $\bar{z} = \emptyset$, and initialize $\mathcal{L} = \{N_0\}$ and $\hat{\mathcal{X}} = \emptyset$.
2. If $\mathcal{L} = \emptyset$, the probing plan $\bar{z}$ is optimal, end.
3. Else, remove node N_ℓ with relaxation bound $\overline{Z}_\ell$ from $\mathcal{L}$.
4. If $\overline{Z}_\ell \leq \underline{Z}$, go to step 2 (*pruned by bound*).
5. Solve (22) and obtain $\hat{\Gamma}_B^\ell(\hat{\mathcal{X}})$, $\hat{z}^{\Xi,\ell}$, and $\hat{x}^{\Xi,\ell}$.
6. If $\hat{\Gamma}_B^\ell(\hat{\mathcal{X}}) \leq \underline{Z}$, go to step 2 (*pruned by bound*).
7. If $\hat{z}^{\Xi,\ell}$ is feasible to (21), set $\hat{z} = z^{\xi,\ell}$ for arbitrary ξ and: (*evaluate if cutting planes are needed*)
   (a) For each $\xi \in \Xi$ solve the problem $\max_{x\in\mathcal{X}} \hat{R}(\hat{z}, \xi, x)$, and call the optimal solution plan $\bar{x}^\xi$.
   (b) If $\hat{R}(\hat{z}, \xi, \hat{x}^{\xi,\ell}) = \hat{R}(\hat{z}, \xi, \bar{x}^\xi)$ for all $\xi \in \Xi$, update $\underline{Z} \leftarrow \hat{\Gamma}_B^\ell(\hat{\mathcal{X}})$, $\bar{z} \leftarrow \hat{z}$, and go to Step 2.
   (c) Else, compute $\alpha = \sum_{\xi\in\Xi} \pi^\xi R(\xi, \bar{x}^\xi)$ and update $\hat{\mathcal{X}} \leftarrow \hat{\mathcal{X}} \cup \left\{\bigcup_{\xi\in\Xi} \bar{x}^\xi\right\}$.
   (d) If $\alpha \geq \underline{Z}$, then update $\underline{Z} \leftarrow \alpha$, and $\bar{z} \leftarrow \hat{z}$. Regardless of the update, go to Step 2.
8. Else (*branching*), from $\hat{\Gamma}_B^\ell(\hat{\mathcal{X}})$ construct $\hat{\Gamma}_B^{\ell_1}(\hat{\mathcal{X}}), \cdots, \hat{\Gamma}_B^{\ell_k}(\hat{\mathcal{X}})$ for $k \geq 2$. Add the new nodes $N_{\ell_1}, \cdots, N_{\ell_k}$ with dual (relaxation) bound $\overline{Z}_{\ell_r} = \hat{\Gamma}_B^{\ell_r}(\hat{\mathcal{X}})$ for $r = 1, \cdots k$, to $\mathcal{L}$ and go to Step 2.

$F_{\ell'} = F_\ell \setminus \{i' \in F_\ell : i' \leq i\}$, where the condition $i' \leq i$ is evaluated according to the ordering. This multiple branching strategy is build upon the one used in Sadykov and Vanderbeck (2013), which was introduced in Carraghan and Pardalos (1990) to solve maximum clique problems. Further, we utilize a depth-first search node selection in Step 3 of Algorithm 1, which is carried out by going deeper into the enumeration tree through the direction of nodes that have a higher expected actual reward. This strategy potentially leads to finding good lower-bounds early and improve the pruning process. Example 2 in Appendix B.3 illustrates the branching procedure utilized.

7 Computational experiments

Next we present computational experiments using longest path and knapsack problem instances. Our objectives are to compare our reformulations; to evaluate the value of probing against a benchmark that does not probe; and to determine what gains can be achieved by exploiting correlated information by comparing against a benchmark that treats components as statistically independent.

7.1 Experiments for the longest path problem

We perform computational experiments over instances of the longest path problem in which arcs are prone to disruptions that decrease the nominal length (referred to as reward). The planner,

however, is able to probe a limited set of arcs and reveal their disruption status before the route is decided. Consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of arcs. When an arc $(i, j) \in \mathcal{A}$ is disrupted, its length decreases from $r_{ij} > 0$ to $r'_{ij} = 0$. We consider instances with $|\mathcal{A}| \in \{18, 28, 38\}$ and $|\Xi| \in \{10, 20, 30\}$. The details of the instance generation is presented in detail in Appendix C.1.

Table 3 reports the results of comparing the performance of (FLVR) and (CS-VFR), which are solved via the B&C method described in Algorithm 1. Each line summarizes five different randomly generated instances with the same number of nodes, arcs, and $|\Xi|$. The first three columns present the number of nodes, arcs in the graph, and the number of scenarios, respectively. The fourth and fifth columns present the average optimal expected actual reward with no probing (Γ_0) and the average optimal expected actual reward with full information (Γ_N), where $N = |\mathcal{A}|$. Columns 6 presents the probing budget. Columns 7 and 8 show the average and maximum full information gap (FIG) computed as $FIG = \frac{\Gamma_N - \Gamma_0}{\Gamma_N}$. Columns 9 to 16 summarize the results using (FLVR), where columns 9 to 14 present the average CPU time, the average lower and upper bounds obtained within the time limit (which is set to two hours), the average optimality gap, the number of instances solved to optimality, and the average number of nodes in the B&C trees (within parenthesis is the average number of optimality cuts added by the algorithm). Column 15 presents the probing value (PV), the average of a standardized measure of the value of information, computed as PV $= \frac{\Gamma_B - \Gamma_0}{\Gamma_B}$, and column 16 presents the price gap (PG), the average of a standardized measure of the price of not having full information, computed as PG $= \frac{\Gamma_N - \Gamma_B}{\Gamma_N}$. For the previous two measures, we use the best solution available as an estimate for Γ_B when given instances are not solved to optimality. Observe that in this case, the probing values we obtain are a lower-bound on the true value, while the price gap values we obtain provide an upper bound for the true value. Finally, Columns 17 to 24 present the same results for the (CS-VFR).

The full information gap is on average 27.2%, with values of up to 50.5%, which shows the high potential improvement achieved by probing. We consistently note that probing a relatively small fraction attains the same performance of perfect information. We highlight that probing at most 30% of the total number of arcs results in objective values within 1.8% of Γ_N and in some cases it is equal to Γ_N. Moreover, with regards to the PV, probing leads to an average of 24.5% of improvement over Γ_0 and a maximum of up to 36.6% depending on the available budget. Finally, even when instances are not solved to optimality, our methods are able to find high-quality solutions with improvements of up to 22.3% with respect to Γ_0, and being within at most 0.1% of Γ_N.

The results show that both B&C approaches are able to solve most instances to optimality; however, (CS-VFR) is computationally more efficient than (FLVR). In particular, the average solution time decreases from 219.2 seconds with (FLVR) to 106.3 seconds with (CS-VFR), which corresponds to roughly a two times speedup. Moreover, both (FLVR) and (CS-VFR) solve 134 out

Table 3: Results comparing (FLVR) and (CS-VFR) using the B&C approach on longest path problems.

						FIG		B&C with (FLVR) formulation								B&C with (CS-VFR) formulation							
$\|\mathcal{V}\|$	$\|\mathcal{A}\|$	$\|\Xi\|$	Γ_0	Γ_N	B	Avg	Max	Time	LB	UB	Gap	Opt	Nodes	PV	PG	Time	LB	UB	Gap	Opt	Nodes	PV	PG
10	18	10	48.4	72.2	1	32.9%	50.5%	3.4	65.0	65.0	0.0%	5	11 (2)	25.2%	10.1%	3	65.0	65.0	0%	5	11 (2)	25.2%	10.1%
			48.4	72.2	2	32.9%	50.5%	2.6	70.4	70.4	0.0%	5	20 (1)	31.1%	2.6%	2	70.4	70.4	0%	5	24 (1)	31.1%	2.6%
			48.4	72.2	3	32.9%	50.5%	2.7	72.0	72.0	0.0%	5	14 (1)	32.7%	0.3%	2	72.0	72.0	0%	5	15 (1)	32.7%	0.3%
		20	46.6	73.7	1	36.8%	44.3%	6.6	65.5	65.5	0.0%	5	11 (1)	28.6%	11.2%	5	65.5	65.5	0%	5	11 (1)	28.6%	11.2%
			46.6	73.7	2	36.8%	44.3%	6.8	72.0	72.0	0.0%	5	30 (1)	35.2%	2.4%	5	72.0	72.0	0%	5	30 (1)	35.2%	2.4%
			46.6	73.7	3	36.8%	44.3%	9.5	73.2	73.2	0.0%	5	45 (1)	36.3%	0.7%	6	73.2	73.2	0%	5	48 (1)	36.3%	0.7%
		30	46.1	73.8	1	37.7%	46.0%	10.1	65.3	65.3	0.0%	5	11 (2)	29.3%	11.7%	8	65.3	65.3	0%	5	11 (2)	29.3%	11.7%
			46.1	73.8	2	37.7%	46.0%	14.5	70.6	70.6	0.0%	5	43 (2)	34.7%	4.6%	12	70.6	70.6	0%	5	42 (2)	34.7%	4.6%
			46.1	73.8	3	37.7%	46.0%	18.1	72.5	72.5	0.0%	5	76 (2)	36.6%	1.8%	13	72.5	72.5	0%	5	74 (2)	36.6%	1.8%
14	28	10	124.2	154.3	1	20.2%	27.2%	4.4	145.0	145.0	0.0%	5	21 (2)	15.0%	6.2%	3	145.0	145.0	0%	5	21 (2)	15.0%	6.2%
			124.2	154.3	2	20.2%	27.2%	7.6	152.8	152.8	0.0%	5	102 (1)	19.5%	0.9%	4	152.8	152.8	0%	5	97 (1)	19.5%	0.9%
			124.2	154.3	4	20.2%	27.2%	2.0	154.3	154.3	0.0%	5	3 (0)	20.2%	0.0%	2	154.3	154.3	0%	5	7 (1)	20.2%	0.0%
		20	108.3	148.7	1	27.1%	29.6%	12.7	131.3	131.3	0.0%	5	21 (3)	17.4%	11.6%	9	131.3	131.3	0%	5	21 (3)	17.4%	11.6%
			108.3	148.7	2	27.1%	29.6%	41.7	144.5	144.5	0.0%	5	193 (3)	25.0%	2.7%	16	144.5	144.5	0%	5	193 (3)	25.0%	2.7%
			108.3	148.7	4	27.1%	29.6%	75.7	148.4	148.4	0.0%	5	512 (1)	27.0%	0.2%	25	148.4	148.4	0%	5	513 (1)	27.0%	0.2%
		30	109.3	150.8	1	27.5%	28.7%	53.6	132.0	132.0	0.0%	5	21 (3)	17.0%	12.5%	43	132.0	132.0	0%	5	21 (3)	17.0%	12.5%
			109.3	150.8	2	27.5%	28.7%	96.6	144.9	144.9	0.0%	5	188 (2)	24.%	3.9%	51	144.9	144.9	0%	5	188 (2)	24.5%	3.9%
			109.3	150.8	4	27.5%	28.7%	213.5	150.6	150.6	0.0%	5	623 (2)	27.3%	0.2%	87	150.6	150.6	0%	5	610 (2)	27.3%	0.2%
18	38	10	183.8	222.7	2	18.1%	26.0%	38.6	219.1	219.1	0.0%	5	391 (3)	16.8%	1.6%	19	219.1	219.1	0%	5	391 (3)	16.8%	1.6%
			183.8	222.7	3	18.1%	26.0%	58.7	222.2	222.2	0.0%	5	602 (1)	18.0%	0.2%	21	222.2	222.2	0%	5	602 (1)	18.0%	0.2%
			183.8	222.7	6	18.1%	26.0%	2.0	222.7	222.7	0.0%	5	3 (0)	18.1%	0.0%	1	222.7	222.7	0%	5	3 (0)	18.1%	0.0%
		20	170.5	218.6	2	22.2%	27.3%	135.5	212.3	212.3	0.0%	5	435 (3)	19.8%	3.0%	46	212.3	212.3	0%	5	435 (3)	19.8%	3.0%
			170.5	218.6	3	22.2%	27.3%	648.3	216.2	216.2	0.0%	5	2697 (2)	21.3%	1.1%	171	216.2	216.2	0%	5	2697 (2)	21.3%	1.1%
			170.5	218.6	6	22.2%	27.3%	48.8	218.6	218.6	0.0%	5	168 (1)	22.2%	0.0%	23	218.6	218.6	0%	5	173 (1)	22.2%	0.0%
		30	169.0	217.6	2	22.4%	26.2%	278.1	209.4	209.4	0.0%	5	453 (4)	19.3%	3.8%	100	209.4	209.4	0%	5	453 (4)	19.3%	3.8%
			169.0	217.6	3	22.4%	26.2%	1454.5	213.8	213.8	0.0%	5	3164 (3)	21.0%	1.8%	365	213.8	213.8	0%	5	3164 (3)	21.0%	1.8%
			169.0	217.6	6	22.4%	26.2%	2671.1	217.4	217.6	0.1%	4	7292 (2)	22.3%	0.1%	1827	217.5	217.6	0.01%	4	18730 (2)	22.3%	0.1%
Summary			111.8	148.0	-	27.2%	34.0%	219.2	143.8	143.8	0.01%	134	635.2 (1.8)	24.5%	3.5%	106.3	143.8	143.8	0.01%	134	1058.7 (1.9)	24.5%	3.5%

of 135 instances to optimality. Furthermore, we note that (CS-VFR) explores more branch-and-bound nodes on average than (FLVR), with 1058.7 nodes compared with 635.2. However, a closer look shows that this difference is mainly driven by the instances for which (FLVR) fails to prove optimality within the time limit.

7.2 Experiments for the knapsack problem

We also study the 0-1 knapsack problem in which the items to be selected are susceptible to uncertain disruptions. The planner is able to probe a limited set of items in order to reveal their disruption status before deciding on its selection, so that the total reward is maximized. We consider a set of items $\mathcal{N} = \{1, \ldots, N\}$. When an item $i \in \mathcal{N}$ fails, its nominal reward $r_i > 0$ is reduced to $r'_i = 0$. We consider instances with $N \in \{10, 20, 30\}$ and $|\Xi| \in \{10, 20, 30\}$; details of the generation of instances is found in Appendix C.2.

Table 4 presents computation experiments that compare the performance of (FLVR) and (CS-VFR). The headers of the columns here are consistent with the headers of the columns of previous experiments on longest path problems. The full information gap is on average 20.1% and up to 29.4%, showing potential improvements achieved by probing. As before, probing a small fraction of the total number of items often attains the value of the full information. Moreover, using (CS-VFR) often outperforms (FLVR) in terms of computational time and solution quality. Finally, instances with higher budget seem to be solved faster than instances with tight budget.

Table 4: Results comparing the (FLVR) and the (CS-VFR) using the B&C approach on Knapsack Problems

					FIG		B&C with (FLVR) formulation								B&C with (CS-VFR) formulation							
$\|\mathcal{N}\|$	$\|\Xi\|$	Γ_0	Γ_N	B	Avg	Max	Time	LB	UB	Gap	Opt	Nodes	PV	PG	Time	LB	UB	Gap	Opt	Nodes	PV	PG
10	10	131.0	155.7	1	16.1%	29.4%	2	146.7	146.7	0%	5	10 (1)	11.0%	5.9%	2	146.7	146.7	0%	5	10 (1)	11.0%	5.9%
		131.0	155.7	3	16.1%	29.4%	2	155.7	155.7	0%	5	17 (0)	16.1%	0.0%	2	155.7	155.7	0%	5	17 (1)	16.1%	0.0%
		131.0	155.7	6	16.1%	29.4%	1	155.7	155.7	0%	5	3 (0)	16.1%	0.0%	1	155.7	155.7	0%	5	3 (0)	16.1%	0.0%
	20	128.7	154.2	1	16.7%	22.2%	5	143.0	143.0	0%	5	11 (1)	10.4%	7.1%	5	143.0	143.0	0%	5	11 (1)	10.4%	7.1%
		128.7	154.2	3	16.7%	22.2%	10	153.3	153.3	0%	5	62 (2)	16.2%	0.5%	8	153.3	153.3	0%	5	62 (2)	16.2%	0.5%
		128.7	154.2	6	16.7%	22.2%	3	154.2	154.2	0%	5	5 (0)	16.7%	0.0%	3	154.2	154.2	0%	5	5 (0)	16.7%	0.0%
	30	125.7	152.6	1	17.1%	23.1%	8	140.1	140.1	0%	5	11 (1)	10.1%	7.9%	6	140.1	140.1	0%	5	11 (1)	10.1%	7.9%
		125.7	152.6	3	17.1%	23.1%	24	150.2	150.2	0%	5	84 (2)	15.9%	1.4%	14	150.2	150.2	0%	5	84 (2)	15.9%	1.4%
		125.7	152.6	6	17.1%	23.1%	8	152.6	152.6	0%	5	21 (1)	17.1%	0.0%	6	152.6	152.6	0%	5	21 (1)	17.1%	0.0%
20	10	289.9	351.0	1	17.7%	23.0%	9	309.3	309.3	0%	5	21 (4)	6.5%	12.0%	6	309.3	309.3	0%	5	21 (4)	6.5%	12.0%
		289.9	351.0	3	17.7%	23.0%	70	348.1	348.1	0%	5	1017 (6)	17.0%	0.8%	30	348.1	348.1	0%	5	1017 (6)	17.0%	0.8%
		289.9	351.0	6	17.7%	23.0%	2	351.0	351.0	0%	5	5 (0)	17.7%	0.0%	2	351.0	351.0	0%	5	5 (0)	17.7%	-0.0%
	20	258.1	331.4	1	22.3%	27.4%	20	277.5	277.5	0%	5	21 (5)	7.1%	16.4%	15	277.5	277.5	0%	5	21 (5)	7.1%	16.4%
		258.1	331.4	3	22.3%	27.4%	300	316.0	316.0	0%	5	1209 (11)	18.6%	4.6%	102	316.0	316.0	0%	5	1209 (11)	18.6%	4.6%
		258.1	331.4	6	22.3%	27.4%	8	331.4	331.4	0%	5	17 (1)	22.3%	0.0%	6	331.4	331.4	0%	5	17 (1)	22.3%	0.0%
	30	249.3	319.4	1	21.9%	24.8%	76	270.7	270.7	0%	5	21 (5)	7.9%	15.2%	23	270.7	270.7	0%	5	21 (5)	7.9%	15.2%
		249.3	319.4	3	21.9%	24.8%	612	299.8	299.8	0%	5	1222 (10)	16.9%	6.1%	191	299.8	299.8	0%	5	1222 (10)	16.9%	6.1%
		249.3	319.4	6	21.9%	24.8%	775	319.4	319.4	0%	5	2609 (4)	21.9%	0.0%	278	319.4	319.4	0%	5	2609 (4)	21.9%	0.0%
30	10	404.8	516.3	1	21.8%	25.1%	21	442.0	442.0	0%	5	31 (10)	8.5%	14.5%	14	442.0	442.0	0%	5	31 (10)	8.5%	14.5%
		404.8	516.3	3	21.8%	25.1%	373	508.6	508.6	0%	5	3913 (15)	20.7%	1.5%	169	508.6	508.6	0%	5	3913 (15)	20.7%	1.5%
		404.8	516.3	6	21.8%	25.1%	3	516.3	516.3	0%	5	16 (1)	21.8%	0.0%	3	516.3	516.3	0%	5	16 (1)	21.8%	0.0%
	20	399.0	517.6	1	23.1%	26.0%	84	426.1	426.1	0%	5	31 (8)	6.5%	17.8%	26	426.1	426.1	0%	5	31 (8)	6.5%	17.8%
		399.0	517.6	3	23.1%	26.0%	2424	479.4	479.4	0%	5	4405 (30)	16.9%	7.4%	593	479.4	479.4	0%	5	4405 (30)	16.9%	7.4%
		399.0	517.6	6	23.1%	26.0%	141	517.6	517.6	0%	5	418 (5)	23.1%	0.0%	48	517.6	517.6	0%	5	418 (5)	23.1%	0.0%
	30	388.1	513.6	1	24.5%	27.9%	127	414.7	414.7	0%	5	31 (7)	6.5%	19.3%	38	414.7	414.7	0%	5	31 (7)	6.5%	19.3%
		388.1	513.6	3	24.5%	27.9%	6581	459.4	494.0	6%	2	4039 (39)	15.7%	10.5%	1387	459.4	459.4	0%	5	4442 (40)	15.7%	10.5%
		388.1	513.6	6	24.5%	27.9%	4196	512.6	513.6	0%	3	5557 (11)	24.4%	0.2%	1974	513.6	513.6	0%	5	12928 (13)	24.5%	0.0%
Summary		263.8	334.6	-	20.1%	25.4%	588.3	313.0	314.3	0.22%	130	918.8 (6.7)	15.2%	5.5%	183.4	313.1	313.1	0.0%	135	1206.7 (6.8)	15.2%	5.5%

7.3 Additional Knapsack experiments with relevant disrupted items

To showcase a setting in which (FLVR) could outperform (CS-VFR) we consider structured knapsack problem instances with $|\mathcal{N}| = 100$ where the value of the reward and weight of each item are generated randomly with the procedure previously described. Items then are ordered in a non-increasing way with respect to the reward/weight ratio, and the first N^R items are selected to be prone to disruptions while the remaining items are not affected by disruptions. We define $\mathcal{N}^R = \{1, \cdots, N^R\}$ as the set of *relevant items*. We consider instances with $|\mathcal{N}^R| \in \{8, 10, 12\}$ and sets $|\Xi| \in \{40, 60\}$ where the scenarios are generated with the previously described procedure.

Table 5 presents results that compare the performance of (FLVR) and (CS-VFR). The name headers of the columns are consistent with the tables of our previous experiments with the addition of the column $|\mathcal{N}^R|$. The full information gap is on average 3.9% with values as high as 7.2%, showing non-trivial improvements achieved by probing. In contrast to previous experiments, our highest budget is often not enough to reach the value of perfect information, but the price gap is closed to being 0%. In addition, (FLVR) consistently outperforms (CS-VFR) in terms of both computational time and solution quality. Specifically, (FLVR) attains an average solution time of 701.5 seconds and solves 90 instances to optimality, whereas the (CS-VFR) attains an average solution time of 1665.6 seconds and solves 84 instances to optimality. These results show that when the scenario set has especially structured scenarios, as in for example, when only a fraction of the components are subject to uncertainty, then (FLVR) outperforms (CS-VFR).

Table 5: Results comparing the (FLVR) and (CS-VFR) on Knapsack Problems with relevant disrupted items

						FIG		B&C with (FLVR) formulation								B&C with (CS-VFR) formulation							
$\|\mathcal{N}\|$	$\|\mathcal{N}^R\|$	$\|\Xi\|$	Γ_0	Γ_N	B	Avg	Max	Time	LB	UB	Gap	Opt	Nodes	PV	PG	Time	LB	UB	Gap	Opt	Nodes	PV	PG
100	8	40	1975.8	2029.1	1	2.6%	3.8%	104	1987.3	1987.3	0%	5	9 (4)	0.6%	2.0%	148	1987.3	1987.3	0%	5	9 (3)	0.6%	2.0%
			1975.8	2029.1	3	2.6%	3.8%	383	2008.8	2008.8	0%	5	78 (44)	1.6%	1.0%	542	2008.8	2008.8	0%	5	78 (11)	1.6%	1.0%
			1975.8	2029.1	6	2.6%	3.8%	182	2028.4	2028.4	0%	5	86 (42)	2.6%	0.0%	313	2028.4	2028.4	0%	5	86 (4)	2.6%	0.0%
		60	1977.6	2030.4	1	2.6%	3.8%	208	1988.7	1988.7	0%	5	9 (4)	0.6%	2.0%	338	1988.7	1988.7	0%	5	9 (3)	0.6%	2.0%
			1977.6	2030.4	3	2.6%	3.8%	757	2008.5	2008.5	0%	5	72 (39)	1.5%	1.1%	1214	2008.5	2008.5	0%	5	72 (11)	1.5%	1.1%
			1977.6	2030.4	6	2.6%	3.8%	388	2028.2	2028.2	0%	5	87 (64)	2.5%	0.1%	865	2028.2	2028.2	0%	5	87 (4)	2.5%	0.1%
	10	40	1912.9	1986.3	1	3.7%	5.2%	93	1927.5	1927.5	0%	5	11 (7)	0.7%	2.9%	186	1927.5	1927.5	0%	5	11 (6)	0.7%	2.9%
			1912.9	1986.3	3	3.7%	5.2%	489	1952.7	1952.7	0%	5	157 (70)	2.0%	1.7%	1058	1952.7	1952.7	0%	5	157 (18)	2.0%	1.7%
			1912.9	1986.3	6	3.7%	5.2%	292	1984.1	1984.1	0%	5	378 (61)	3.5%	0.1%	904	1984.1	1984.1	0%	5	378 (8)	3.5%	0.1%
		60	1903.3	1979.3	1	3.8%	5.4%	278	1918.7	1918.7	0%	5	11 (7)	0.8%	3.0%	336	1918.7	1918.7	0%	5	11 (6)	0.8%	3.0%
			1903.3	1979.3	3	3.8%	5.4%	1185	1941.6	1941.6	0%	5	159 (77)	2.0%	1.9%	2593	1941.6	1941.6	0%	5	159 (24)	2.0%	1.9%
			1903.3	1979.3	6	3.8%	5.4%	840	1974.4	1974.4	0%	5	427 (111)	3.6%	0.2%	3072	1974.4	1974.4	0%	5	426 (14)	3.6%	0.2%
	12	40	1831.2	1934.3	1	5.3%	7.0%	197	1849.5	1849.5	0%	5	13 (11)	1.0%	4.3%	241	1849.5	1849.5	0%	5	13 (8)	1.0%	4.3%
			1831.2	1934.3	3	5.3%	7.0%	1288	1884.2	1884.2	0%	5	282 (134)	2.8%	2.6%	2699	1884.2	1884.2	0%	5	282 (36)	2.8%	2.6%
			1831.2	1934.3	6	5.3%	7.0%	847	1931.2	1931.2	0%	5	1293 (83)	5.1%	0.2%	2711	1931.2	1931.2	0%	5	1293 (16)	5.1%	0.2%
		60	1828.5	1932.5	1	5.3%	7.2%	504	1846.5	1846.5	0%	5	13 (10)	1.0%	4.4%	548	1846.5	1846.5	0%	5	13 (8)	1.0%	4.4%
			1828.5	1932.5	3	5.3%	7.2%	2286	1878.0	1878.0	0%	5	285 (116)	2.6%	2.8%	5009	1877.3	1894.4	1%	4	257 (31)	2.6%	2.8%
			1828.5	1932.5	6	5.3%	7.2%	2306	1924.1	1924.1	0%	5	1560 (158)	4.9%	0.4%	7200	1924.1	1932.5	0%	0	990 (19)	4.9%	0.4%
Summary			1904.9	1982.0	-	3.9%	5.4%	701.5	1947.9	1947.9	0.0%	90	273.9 (57.9)	2.2%	1.7%	1665.6	1947.9	1949.3	0.06%	84	240.6 (12.8)	2.2%	1.7%

7.4 Comparison between the dependent and independent case

We now turn our attention to the potential gains related with considering correlations as opposed to assuming independence between the components of the uncertainty vector. We compare our method with the method in which the random variables J_i are assumed to be independent (Lozano and Borrero 2025a). In this case, the probability that component $i \in \mathcal{N}$ is disrupted is given by $p_i = P[J_i = 1 | J_z = \xi_z] = P[J_i = 1] = \sum_{\xi \in \Xi : \xi_i = 1} \pi^\xi$. Then, for a solution plan z and a scenario ξ, we replace $P_i(z, \xi)$ with p_i, $i \in \mathcal{N}$; that is, the probability of a component being disrupted is independent of the information revealed by probing. The resulting formulation is solved by using a branch-and-cut algorithm akin to Algorithm 1.

Table 6 and compares the results obtained by using the independent and the correlated assumptions for the probing problem over instances of the longest path problem (results for knapsack instances are relegated to Appendix C.3). The last column, column Δ, in each table presents the percentage of increase between the two approaches with respect to Γ_0. We observe that the maximum percentage of increase is 59.72% while the average is 29.93%, showing that despite being a more computationally expensive approach, the dependent version always leads to significant improvements in the objective value.

8 Conclusions

We study a class of combinatorial optimization problems subject to uncertain objective disruptions in which the planner can collect information to learn the disruption status of some components. In contrast to settings assuming independence, we consider correlated disruptions. We propose two mixed-integer linear programming reformulations of the problem and a branch and cut algorithm to solve it. Moreover, we perform a computational study over instances of the longest path problem

Table 6: Comparison of independent and correlated disruptions in longest path instances.

					Assuming Independence						Considering Correlations						
$\|N\|$	$\|A\|$	$\|\Xi\|$	Γ_0	B	Time	LB	UB	Gap	Opt	Nodes	Time	LB	UB	Gap	Opt	Nodes	Δ
10	18	10	48.4	1	2	63.4	63.4	0%	5	11 (1)	3	65.0	65.0	0%	5	11 (2)	10.67%
			48.4	2	3	65.1	65.1	0%	5	18 (2)	2	70.4	70.4	0%	5	25 (1)	31.74%
			48.4	3	3	65.6	65.6	0%	5	24 (2)	2	72.0	72.0	0%	5	18 (1)	37.21%
		20	46.6	1	4	62.2	62.2	0%	5	11 (1)	5	65.5	65.5	0%	5	11 (2)	21.15%
			46.6	2	6	64.5	64.5	0%	5	21 (2)	5	72.0	72.0	0%	5	30 (1)	41.9%
			46.6	3	6	65.5	65.5	0%	5	21 (2)	6	73.2	73.2	0%	5	48 (1)	40.74%
		30	46.1	1	7	62.1	62.1	0%	5	11 (1)	8	65.3	65.3	0%	5	11 (2)	20.0%
			46.1	2	9	65.3	65.3	0%	5	22 (2)	12	70.6	70.6	0%	5	42 (2)	27.60%
			46.1	3	9	66.6	66.6	0%	5	32 (2)	13	72.5	72.5	0%	5	74 (1)	28.68%
14	28	10	124.2	1	3	139.4	139.4	0%	5	21 (2)	3	145.0	145.0	0%	5	21 (2)	36.84%
			124.2	2	4	144.9	144.9	0%	5	55 (2)	4	152.8	152.8	0%	5	97 (1)	38.16%
			124.2	4	7	151.7	151.7	0%	5	117 (4)	2	154.3	154.3	0%	5	7 (0)	9.45%
		20	108.3	1	10	122.7	122.7	0%	5	21 (3)	9	131.3	131.3	0%	5	21 (3)	59.72%
			108.3	2	14	134.7	134.7	0%	5	79 (3)	16	144.5	144.5	0%	5	193 (3)	37.12%
			108.3	4	31	144.9	144.9	0%	5	226 (4)	25	148.4	148.4	0%	5	514 (1)	9.56%
		30	109.3	1	12	124.3	124.3	0%	5	21 (3)	43	132.0	132.0	0%	5	21 (3)	51.33%
			109.3	2	14	135.3	135.3	0%	5	83 (3)	51	144.9	144.9	0%	5	188 (2)	36.92%
			109.3	4	35	144.8	144.8	0%	5	253 (4)	87	150.6	150.6	0%	5	607 (2)	16.34%
18	38	10	183.8	2	5	207.9	207.9	0%	5	118 (2)	19	219.1	219.1	0%	5	391 (3)	46.47%
			183.8	3	8	214.1	214.1	0%	5	235 (4)	21	222.2	222.2	0%	5	602 (1)	26.73%
			183.8	6	26	219.7	219.7	0%	5	1075 (6)	1	222.7	222.7	0%	5	3 (0)	8.36%
		20	170.5	2	11	199.9	199.9	0%	5	133 (3)	46	212.3	212.3	0%	5	435 (3)	42.18%
			170.5	3	21	206.4	206.4	0%	5	386 (4)	171	216.2	216.2	0%	5	2697 (2)	27.30%
			170.5	6	77	213.3	213.3	0%	5	1899 (6)	23	218.6	218.6	0%	5	173 (1)	12.38%
		30	169.0	2	40	196.3	196.3	0%	5	152 (3)	100	209.4	209.4	0%	5	453 (4)	47.99%
			169.0	3	83	203.9	203.9	0%	5	393 (3)	365	213.8	213.8	0%	5	3164 (3)	28.4%
			169.0	6	399	211.9	211.9	0%	5	2021 (6)	1827	217.5	217.6	0.1%	4	18730 (2)	13.1%

and the knapsack problem. Our results suggest some interesting practical insights for decision making under uncertainty, which we summarize as follows. *A little information goes a long way:* we consistently observe that even small probing budgets can lead to considerable gains in expected rewards. *Considering correlations is worth the effort:* even though models that consider correlations in the scenarios instead of assuming independence are considerably more complex and challenging to solve, our results suggest that there is a substantial gain in solution quality when correlations are exploited. This is specially interesting since scenario independence is one of the most commonly used simplifying assumptions in the literature, which is often needed to ensure the models are tractable or for theoretical results to hold. *Different formulations may affect the computational performance:* the analysis of the two proposed formulations and the computational experiments suggest that the flow-based reformulation of the probing problem may be preferred when the scenarios are highly structured, whereas the consistency-set reformulation may be preferred for unstructured settings; for example, when the disruption scenarios are weakly dependent on one another. *Conditional expectation rules:* we show that using conditional expectation to estimate the reward given the partial information collected by probing is indeed the best approach, settling this open question from previous works in the literature.

References

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

Blackwell D (1953) Equivalent comparisons of experiments. *The annals of mathematical statistics* 265–272.

Carraghan R, Pardalos PM (1990) An exact algorithm for the maximum clique problem. *Operations Research Letters* 9(6):375–382.

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

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

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).

Lozano L, Borrero JS (2025a) A bilevel optimization approach for a class of combinatorial problems with disruptions and probing. *INFORMS Journal on Computing* 37(6):1478–1499.

Lozano L, Borrero JS (2025b) On solving integer robust optimization problems with integer decision-dependent uncertainty sets: L. lozano, js borrero. *Mathematical Programming* 1–33.

Ma Z, Kim Y, Linderoth J, Luedtke JR, Matthews LR (2024) Probing-enhanced stochastic programming. *arXiv preprint arXiv:2407.10669* .

McCormick GP (1976) Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming* 10(1):147–175.

Paradiso R, Georghiou A, Dabia S, Tönissen D (2024) Exact and approximate schemes for robust optimization problems with decision-dependent information discovery. *INFORMS Journal on Computing* .

Quezada F, Gicquel C, Kedad-Sidhoum S, Pagnoncelli B (2025) A probing-enhanced stochastic programming approach for the capacitated multi-item lot-sizing problem. *26ème Congrès Annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2025),*.

Resnick SI (1999) *A Probability Path*. Modern Birkhäuser Classics (Boston: Birkhäuser), ISBN 978-0-8176-8409-9.

Sadykov R, Vanderbeck F (2013) Bin packing with conflicts: a generic branch-and-price algorithm. *INFORMS Journal on Computing* 25(2):244–255.

Singla S (2018) The price of information in combinatorial optimization. *Proceedings of the twenty-ninth annual ACM-SIAM symposium on discrete algorithms*, 2523–2532 (SIAM).

Vayanos P, Georghiou A, Yu H (2025) Robust optimization with decision-dependent information discovery. *Management Science* .

Williams D (1991) *Probability with martingales* (Cambridge university press).

A Proof that the decision rule induced by (BLb) is the best

A.1 Probability model and conditional expectations

The probability model for our setting is as follows. Consider an abstract probability space $(\Omega, \mathcal{F}, P)$ and the probability space $(\Xi, 2^{\Xi}, Q)$, which is a finite probability space and the measure Q is defined by $Q(\{\xi\}) = \pi^{\xi}$. The random variables J_i, $i \in \mathcal{N}$, are functions $J_i \colon \Omega \to \{0,1\}$ such that $J_i^{-1}(\xi_i) = \{\omega \in \Omega \colon J_i(\omega) = \xi_i\} \in \mathcal{F}$ for all $\xi_i \in \{0,1\}$ and $i \in \mathcal{N}$.

Fix $z \in \{0,1\}^{|\mathcal{N}|}$, let $J_z = (J_i \colon z_i = 1)$ and $B(z) = \{i \in \mathcal{N} \colon z_i = 1\}$, and let $\sigma(J_z)$ be the σ-algebra generated by the vector J_z. Note that $\sigma(J_z)$ is generated by sets of the form $J_i^{-1}(\xi_i)$, $i \in B(z)$ and $\xi_i \in \{0,1\}$, and as such it is a finite σ-algebra. More specifically, its *atoms* are given by sets of the form $J_i^{-1}(\xi_i)$, $i \in B(z)$ and $\xi_i \in \{0,1\}$, and any event $A \in \sigma(J_z)$ can be written as a union of such atoms. Moreover, for a given $\xi \in \Xi$ let $\xi_z = (\xi_i \colon i \in B_z)$. Then it can be be verified that the atoms of $\sigma(J_z)$ are given by $\{\omega \in \Omega \colon J_z(\omega) = \xi_z\}$, for any $\xi_z \in \Xi_z = \{0,1\}^{|B_z|}$. We will denote $[J_z = \xi_z] = \{\omega \in \Omega \colon J_z(\omega) = \xi_z\}$.

Recall the following definitions and properties, see Resnick (1999): suppose we are given a probability space $(\Omega, \mathcal{F}, P)$, any σ-algebra $\mathcal{G} \subseteq \mathcal{F}$, and a random variable $X \colon \Omega \to \mathbb{R}$ with $E[|X|] < \infty$. Then $E[X|\mathcal{G}]$, the conditional expected value of X given $\mathcal{G}$, is a random variable that is $\mathcal{G}$-measurable such that

$$\int_A E[X|\mathcal{G}]\, dP(\omega) = \int_A X\, dP(\omega) \qquad \forall A \in \mathcal{G}. \tag{24}$$

Among others, the conditional expected value satisfies the following properties:

a. $E[E[X|\mathcal{G}]] = E[X]$.

b. $E[X|\mathcal{G}](\omega) = X(\omega)$ almost surely (a.s) if X is $\mathcal{G}$-measurable,

where almost surely means that the property holds for all $\omega \in \Omega$ except in a set S with probability zero (i.e., such that $P(S) = 0$).

Next, for any random variable $V \colon \Omega \to \mathcal{V} \subset \mathbb{R}$ with $E[|V|] < \infty$ and where $\mathcal{V}$ is finite, we show that:

$$E[V|\sigma(J_z)](\omega) = \sum_{v \in \mathcal{V}} vP[V = v|J_z = J_z(\omega)]. \tag{25}$$

Indeed, because $\sigma(J_z)$ is finite, it can be shown from (24) that

$$E[V|\sigma(J_z)] = \sum_{\xi_z \in \Xi_z} \frac{E[V\mathbb{I}_{[J_z=\xi_z]}]}{P[J_z = \xi_z]} \mathbb{I}_{[J_z=\xi_z]}, \tag{26}$$

where $\mathbb{I}_A : \Omega \to \{0, 1\}$ is the indicator function of event $A \in \mathcal{F}$. Observe that

$$E[V\mathbb{I}_{[J_z=\xi_z]}] = \sum_{v\in\mathcal{V}} vP[V\mathbb{I}_{[J_z=\xi_z]} = v] \tag{27a}$$

$$= \sum_{v\in\mathcal{V}} vP[V = v, J_z = \xi_z] \tag{27b}$$

$$= \sum_{v\in\mathcal{V}} vP[V = v|J_z = \xi_z]P[J_z = \xi_z], \tag{27c}$$

where the second equation follows as $[V\mathbb{I}_{[J_z=\xi_z]} = v]$ happens if and only if $[V = v, J_z = \xi_z]$ happens, and the last equation follows from the definition of conditional probability. It can be concluded that

$$E[V|\sigma(J_z)] = \sum_{\xi_z\in\Xi_z} \mathbb{I}_{[J_z=\xi_z]} \sum_{v\in\mathcal{V}} vP[V = v|J_z = \xi_z]. \tag{28}$$

Pick $\omega \in \Omega$, and let $\xi'_z = J_z(\omega)$. Then $\mathbb{I}_{[J_z=\xi_z]}(\omega) = 1$ if and only if $\xi_z = \xi'_z$. Thus, from (28) it follows that:

$$E[V|\sigma(J_z)](\omega) = \sum_{v\in\mathcal{V}} vP[V = v|J_z = \xi'_z] = \sum_{v\in\mathcal{V}} vP[V = v|J_z = J_z(\omega)], \tag{29}$$

and Equation (25) follows.

In the context of our problem, for any $x \in \mathcal{X}$ we have that

$$E[R(z, x)|\sigma(J_z)] = E\Big[\sum_{i\in\mathcal{N}} r_i(1 - J_i)x_i|\sigma(J_z)\Big] \tag{30a}$$

$$= E\Big[\sum_{i\,:\,z_i=1} r_i(1 - J_i)x_i)|\sigma(J_z)\Big] + E\Big[\sum_{i\,:\,z_i=0} r_i(1 - J_i)x_i|\sigma(J_z)\Big] \tag{30b}$$

$$= \sum_{i\,:\,z_i=1} r_i(1 - J_i)x_i + \sum_{i\,:\,z_i=0} r_i x_i(1 - E[J_i|\sigma(J_z)])\Big). \tag{30c}$$

where the last equation follows from the fact that $J_i, i \in B_z$, are $\sigma(J_z)$ measurable and from property (b.) of conditional expectations. Pick $\omega \in \Omega$ and let $\xi = J(\omega)$. We have from (29) that

$$E[J_i|\sigma(J_z)](\omega) = P[J_i = 1|J_z = \xi_z] = P_i(z, \xi), \tag{31}$$

and therefore,

$$E[R(z, x)|\sigma(J_z)](\omega) = \sum_{i\,:\,z_i=1} r_i(1 - \xi_i)x_i + \sum_{i\,:\,z_i=0} r_i x_i(1 - P_i(z, \xi)). \tag{32}$$

The right hand side of the above equation is precisely how the planner estimates the reward for any

given $x \in \mathcal{X}$; that is, $E[R(z,x)|\sigma(J_z)](\omega) = \hat{R}(z,\xi,x)$, where $\xi = J(\omega)$.

A.2 Optimality of the decision rule (BLb)

A *decision rule* is a function $d\colon \Omega \to \mathcal{X}$, that for any realization of the uncertainty ω assigns a vector $d(\omega) \in \mathcal{X}$. Hereafter, we will denote by $\hat{d}$ be the decision rule obtained by solving (BLb) (thus $\hat{d}(\omega) = x^{J(\omega)}$). Notice that there might be several $\xi \in \Xi$ for which (BLb) has multiple optimal solutions. We will assume that $\hat{d}$ breaks ties on each ξ by using a deterministic rule (e.g., by using the lexicographical ordering).

Lemma 1. *$\hat{d}$ is $\sigma(J_z)$-measurable.*

Proof. Fix $\omega \in \Omega$ and let $\xi = J(\omega)$. Observe that from (BLb) and (32) we have that

$$\hat{d}(\omega) \in \arg\max\left\{\hat{R}(z, J(\omega), x)\colon x \in X\right\} = \arg\max\left\{E[R(x,J)|\sigma(J_z)](\omega)\colon x \in X\right\}. \tag{33}$$

For any $x \in X$, the function inside the arg max is $\sigma(J_z)$-measurable because because $E[R(J,x)|\sigma(J_z)]$ is $\sigma(J_z)$-measurable by the definition of conditional expectations.

On the other hand, the maximum of finitely many functions is measurable. Indeed, for a given $x \in X$, define $g(\xi) = \max\{f_x(\xi)\colon x \in \mathcal{X}\}$, where each $f_X(\cdot)$ is measurable, $x \in \mathcal{X}$. We have that for any interval of the form (a,∞), with $a \in \mathbb{R}$,

$$g^{-1}(a,\infty) = \bigcup_{x\in\mathcal{X}} f_x^{-1}(a,\infty). \tag{34}$$

In other words $g(\xi) > a$ if and only if there exist $x \in \mathcal{X}$ such that $f_x(\xi) > a$. As $f_x^{-1}(a,\infty)$ belongs to the underlying σ-algebra as f_x is measurable, and because the union in (34) is finite, we can conclude that $g^{-1}(a,\infty)$ is a finite union of measurable sets, which implies that it is a measurable set and that g is measurable. Finally, because in case of ties, the element in the argmax is selected in a deterministic manner (and thus in a measurable way, as deterministic functions are measurable), it follows that $\hat{d}$ is a composition of measurable functions and hence it is measurable. □

Lemma 2. *Let d be a decision rule that is measurable with respect to $\sigma(J_z)$. Then, for each $\omega \in \Omega$*

$$E[R(z,d)|\sigma(J_z)](\omega) = \hat{R}(z, J(\omega), d(\omega)). \tag{35}$$

*P*roof: This result is very similar to the derivation of (32), with the difference that instead of a fixed

x we have a decision rule d. Note that

$$E[R(z,d)|\sigma(J_z)] = E\Big[\sum_{i\in\mathcal{N}} r_i(1-J_i)d_i|\sigma(J_z)\Big] \tag{36a}$$

$$= E\Big[\sum_{i:\,z_i=1} r_i(1-J_i)d_i|\sigma(J_z)\Big] + E\Big[\sum_{i:\,z_i=0} r_i(1-J_i)d_i|\sigma(J_z)\Big] \tag{36b}$$

$$= \sum_{i:\,z_i=1} r_i(1-J_i)d_i + \sum_{i:\,z_i=0} r_i\Big(E[d_i(1-J_i)|\sigma(J_z)]\Big) \tag{36c}$$

$$= \sum_{i:\,z_i=1} r_i(1-J_i)d_i + \sum_{i:\,z_i=0} r_i\Big(d_iE[(1-J_i)|\sigma(J_z)]\Big), \tag{36d}$$

where the last two equations follow from the fact that J_i, $i \in B_z$, and d are $\sigma(J_z)$-measurable and from property (b.) of conditional expectations. The desired result follows after evaluating in ω, noting that from (29) we have $E[(1-J_i)|\sigma(J_z)](\omega) = 1 - P_i(z, J(\omega))$.

Remark 3. Notice the measurability of d with respect to $\sigma(J_z)$ is fundamental for the proof of Lemma 2. If d is not $\sigma(J_z)$-measurable, then equations (36c)–(36c) are not necessarily true.

Lemma 3. *Let d be a decision rule that is measurable with respect to $\sigma(J_z)$. Then*

$$E[R(z,\hat{d})|\sigma(J_z)](\omega) \geq E[R(z,d)|\sigma(J_z)](\omega) \qquad \forall \xi \in \Xi. \tag{37}$$

Proof. Fix $\omega \in \Omega$. We have that

$$E[R(z,\hat{d})|\sigma(J_z)](\omega) = \hat{R}(z, J(\omega), \hat{d}(\omega)) \tag{38a}$$

$$\geq \hat{R}(z, J(\omega), d(\omega)) \tag{38b}$$

$$= E[R(z,d)|\sigma(J_z)](\omega), \tag{38c}$$

where the first and last equation from Lemma 2 as $\hat{d}$ and d is measurable, and the inequality follows from the definition of $\hat{d}$. □

Theorem 4. *For any decision rule $d\colon \Omega \to \mathcal{X}$ that is measurable with respect to $\sigma(J_z)$, we have that*

$$E[R(z,\hat{d})] \geq E[R(z,d)]. \tag{39}$$

Proof. From Lemma 3 we have $E[R(z,\hat{d})|\sigma(J_z)](\omega) \geq E[R(z,d)|\sigma(J_z)](\omega)$ a.s. This implies $E[E[R(z,\hat{d})|\sigma(J_z)]] \geq E[E[R(z,d)|\sigma(J_z)]]$. Moreover, by property (b.) of conditional expectations, we have $E[E[R(z,\hat{d})|\sigma(J_z)]] = E[R(z,\hat{d})]$ and $E[E[R(z,d)|\sigma(J_z)]] = E[R(z,d)]$. We can conclude that $E[R(z,\hat{d})] \geq E[R(z,d)]$, as desired. □

Theorem 5. *(Information always has non-negative value). Let $w < z$ and let d' the optimal solution of* (BLb) *for w. Then, $R[(z, \hat{d})] \geq R[(z, d')]$.*

Proof. The result follows from the fact that d' is $\sigma(J_z)$-measurable. Indeed, let $\xi_w \in \{0,1\}^{|B_w|}$ be given, and let $\xi_w^-, \xi_w^+ \in \{0,1\}^{|B_z|}$ be defined by

$$\xi_{w,i}^- = \begin{cases} \xi_{w,i}, & \text{if } i \in B_w \\ 0, & \text{otherwise,} \end{cases} \qquad \xi_{w,i}^+ = \begin{cases} \xi_{w,i}, & \text{if } i \in B_w \\ 1, & \text{otherwise.} \end{cases} \tag{40}$$

Then, it is readily seen that $[J_w = \xi_w] = [J_z = \xi_w^-] \cup [J_z = \xi_w^+]$. Therefore, the atoms $[J_w = \xi_w]$, $\xi_w \in \{0,1\}^{|B_w|}$, are all members of $\sigma(J_z)$, which implies that $\sigma(J_w) \subseteq \sigma(J_z)$, and therefore, that d' is $\sigma(J_z)$-measurable. The desired result now follows from Theorem 4. □

B Proofs of selected results

B.1 Proofs of Section 4.3

Proof. Proof of Proposition 2. We prove that the problems are equal by proving that their feasible regions are the same.

Let (z, x^Ξ) be feasible in (VFR). Fix $\xi \in \Xi$ and $x \in \mathcal{X}$. Consider the maximum flow representation of $\tilde{R}(z, \xi, x^\xi)$ and note that from the maximization sense of problem (17), constraint (VFRb) holds for ξ and $x \in \mathcal{X}$ if and only if there exists a $y^\xi \in \mathcal{Y}(z, \xi, x^\xi)$ such that

$$y_{ts}^\xi \geq \tilde{R}(z, \xi, x), \tag{41}$$

On the other hand, consider the min cut representation of $\tilde{R}(z, \xi, x)$ and note that the minimization sense of (18) implies that (41) holds if and only if there exists a $(\lambda^\xi, \alpha^\xi)$ feasible in (19c) such that

$$t_{ts}^\xi \geq \sum_{(u,v)\in E} c_{uv}(z, \xi, x)\lambda_{uv}^\xi \tag{42}$$

It can be concluded that (z, x^Ξ) satisfies (VFRb) if and only if there exist non-negative y^ξ and λ^ξ for each $\xi \in \Xi$ such that (19b) and (19c) hold, which shows that (z, x^Ξ) is feasible in (FLVR) .

Conversely, let $(z, x^\Xi, y^\Xi, \lambda^\Xi, \alpha^\Xi)$ be feasible in (FLVR) . By directly using the if and only if arguments of the previous sense of the proof, it follows that (z, x^Ξ) satisfies (VFR). Since x and ξ were chosen arbitrarily, the result then follows. □

B.2 Proofs of Section 5

Proof. Proof of Proposition 3. Suppose $\zeta \in \Xi(z, \xi)$. Pick $i \in U(\xi, \zeta)$; then, by definition we have that $\xi_i \neq \zeta_i$. If $z_i = 1$, then ζ would have to satisfy that $\zeta_i = \xi_i$ in order to belong to the set $\Xi(z, \xi)$, which contradicts $\xi_i \neq \zeta_i$. Since $i \in U(\xi, \zeta$ was chosen arbitrarily, it follows that $z_i = 0$ for all $i \in U(\xi, \zeta)$.

Next, suppose that $z_i = 0$ for all $i \in U(\xi, \zeta)$. Pick $i \in \mathcal{N}$ such that $z_i = 1$. Since $z_i = 1$, the assumption implies that $i \notin U(\xi, \zeta)$. Therefore, by definition of $U(\xi, \zeta)$, we have $\xi_i = \zeta_i$. Thus, ζ agrees with ξ on every probed component. Hence, $\zeta \in \Xi(z, \xi)$. This completes the proof. □

Proof. Proof of Proposition 4. The number of additional variables in the (FLVR) formulation is $\text{total}_{\text{FLVR}} = |\Xi|(3A + B + 1) = |\Xi|\big(3(L + |\Xi|) + (L + 2) + 1\big) = |\Xi|(4L + 3|\Xi| + 3)$ whereas $\text{total}_{\text{CS}} = |\Xi|^2(n + 1)$. Therefore, the (FLVR) reformulation dominates formulation (CS-VFR) if and only if

$$|\Xi|(4L + 3|\Xi| + 3) < |\Xi|^2(n + 1) \iff L < \frac{|\Xi|(n - 2) - 3}{4}.$$

Dividing both sides by $|\Xi|n$ yields

$$\kappa = \frac{L}{|\Xi|n} < \frac{|\Xi|(n - 2) - 3}{4|\Xi|n} = \frac{n - 2 - 3/|\Xi|}{4n}.$$

This completes the proof. □

Remark 6. Suppose that $n = 40$ and the number of prefixes at each layer is given in Table 7; the first row of the Table indicates that the first 30 components have the same value in all scenarios, say the value is 1. The second row indicates that at layer 31, the value of the component in 31 creates two branches, say the first one having the value of 1 and the second one having the value of 0. Therefore, $L = 30(1) + 3(2) + 1(4) + 1(8) + 1(16) + 4(32) = 192$ and therefore the total number of variables of the (FLVR) is $\text{total}_{\text{FLVR}} = |\Xi|(4L + 3|\Xi| + 3) = 32(4(192) + 3(32) + 3) = 27,444$. On the other hand, the number of variables of (CS-VFR) is $\text{total}_{\text{CS}} = |\Xi|^2(n + 1) = 1024(41) = 41,984$.

Now suppose that the components are reordered so that all components that are constant across scenarios appear first, and the five components that vary across scenarios appear at the end. Then the first 35 layers have only one prefix, and the last five layers split as 2, 4, 8, 16, 32. Thus, we would have $L = 35(1) + 2 + 4 + 8 + 16 + 32 = 97$, and therefore $\text{total}_{\text{FLVR}} = 32(4(97) + 3(32) + 3) = 32(487) = 15,584$. □

B.3 Illustration of the BB strategy

Next, we have an example that illustrate the multiple branching technique used in our B&C algorithm.

Table 7: Numerical example

Layers	$\lvert P_k(\Xi)\rvert$
$1, \cdots, 30$	1
$31, \cdots, 33$	2
34	4
35	8
36	16
$37, \cdots, 40$	32

Example 2. Consider a problem with a set of components $\mathcal{N} = \{i_1, i_2, i_3\}$; we assume that the expected actual reward of i_k is larger than the expected actual reward of i_{k+1}, $k = 1, 2$ and that $\hat{\mathcal{X}} = \mathcal{X}$. Figure 2 illustrates the multi-branching strategy utilized in the BB procedure. At the root node (node 0), problem $\hat{\Gamma}_B(\hat{\mathcal{X}})$ is solved. If the resulting probing decisions differ across scenarios, three child nodes are created, namely, nodes 1, 5 and 7. In node 1, we fix $z_{i_1} = 1$ and the remaining variables remain free. In node 5, we set $z_{i_2} = 1$, $z_{i_1} = 0$, and z_{i_3} remains free. Finally, in node 7, we fix $z_{i_3} = 1$ and $z_{i_1} = z_{i_2} = 0$.

Assuming a depth-first search strategy, node 1 of the BB tree is explored next. Suppose the solution of the relaxation here is not feasible in the original problem. Then, we branch on the remaining free components, namely, on the components in the set $F_1 = \{i_2, i_3\}$, which results in creating nodes 2 and 4. In node 2, we fix $z_{i_1} = z_{i_2} = 1$ and z_{i_3} remains free. On the other hand, in node 4, we fix $z_{i_1} = z_{i_3} = 1$ and $z_{i_2} = 0$. Observe that if the budget $B = 1$, only the first layer of the BB tree is explored, whereas if $B = 3$, all the nodes depicted in Figure 2 may be explored.

Assume $B = 3$. The node numbering in Figure 2 indicates the order in which the corresponding problems are solved. Since the components are ordered according to the highest expected actual reward, node 3 potentially gives a good lower-bound which is obtained early in the BB procedure. Moreover, observe that node 4 is not solved (pruned by bound) because node 3 is solved first. By a similar argument, neither node 6 nor node 7 are solved either. It can be verified that the Figure 2 enumerates all feasible solutions.

C Details and complementary results of Section 7

C.1 Instance generation: Longest path problems

The nodes of $\mathcal{G}$ are randomly located in a two-dimensional space of size 100×100. We consider acyclic grid graph structures of sizes 4×2, 4×3, and 4×4 nodes. Additionally, we introduce a source node connected to all nodes in the first layer and a sink node connected from all nodes in the last layer. Edges emanating from source node or terminating at the sink node cannot be

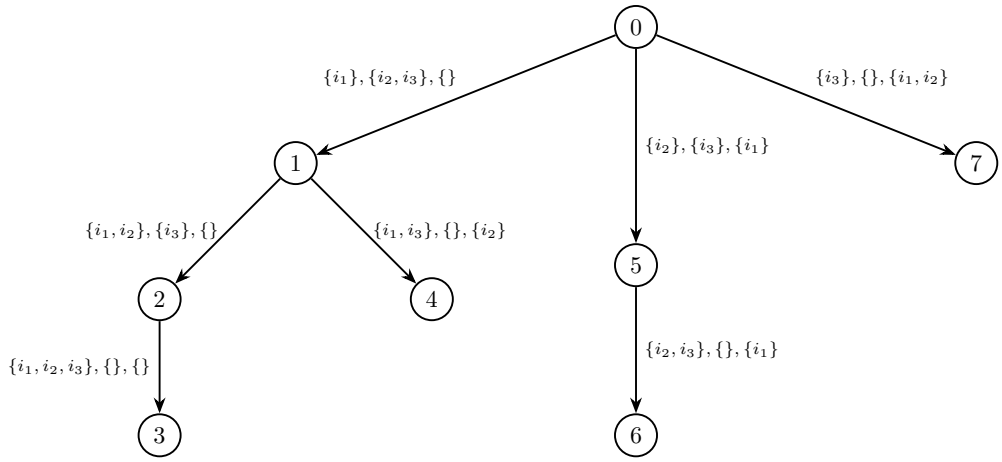


Figure 2: The BB tree for Example 2. The leave entering each node ℓ depicts the sets C^1_ℓ, F_ℓ, and C^0_ℓ, respectively.

disrupted. Consequently, the resulting graphs contain 18, 28, and 38 edges, of which 10, 20, and 30 edges, respectively, are susceptible to disruptions. The value of r_{ij} for each $(i, j) \in \mathcal{A}$ is obtained by computing the distance in the ℓ_2-norm between the location of node i and the location node j. The set of scenarios Ξ is *structured*, namely, the planner knows a finite set of coordinates $\mathbf{P} = \{\alpha^1, \cdots, \alpha^{|\Xi|}\}$ where the epicenter of the disruption might be. Scenario ξ^k, $k = 1, \ldots, |\Xi|$, is defined by $\xi^k_{ij} = 1$ if and only if $d(\alpha, p_{ij}) \leq Q$. Here, $Q > 0$ is the radius of the disruption, p_{ij} has the coordinates of the midpoint of arc $(i, j) \in \mathcal{A}$, and d is a distance function. We set Q equal to one-third of the length of one side of the region under consideration, and consider that the planner sets $|\Xi| \in \{10, 20, 30\}$.

C.2 Instance generation: Knapsack problems

Each item has an associated weight given by w_i, $i \in \mathcal{N}$, and the knapsack has a capacity of W. Let x_i be the binary variable that takes the value of one if and only if item i is selected. The set of feasible solutions $\mathcal{X}$ for this problem is given by $\mathcal{X} = \{x \in \{0, 1\}^N : \sum_{i\in\mathcal{N}} w_i x_i \leq W\}$. To generate random instances, we consider $N \in \{10, 20, 30\}$ and for each item, we draw the coefficients w_i independently from a discrete random uniform distribution $U(1, 50)$ and the coefficients r_i from a discrete random uniform distribution $U(50, 100)$. We set $W = 0.1 \sum_{i\in\mathcal{N}} w_i$ and consider scenario configurations with $|\Xi| \in \{10, 20, 30\}$, where we set the probability of failure to 0.5 for each item.

C.3 Comparison between the dependent and independent case: knapsack instances

Table 8 compares the results obtained by using the independent and the correlated assumptions for the probing problem over instances of the knapsack problem. We observe that the maximum percentage of increase is 65.02% while the average is 18.64%. Note that for small instances with the largest budget, the difference between the two approaches is 0% because the budget being used is large (with respect to N). These results shows that exploiting correlations can significantly improve the performance of the decision-maker.

Table 8: Comparison of independent and correlated disruptions in knapsack problem instances.

				Assuming Independence						Considering Correlations						
$\|N\|$	$\|\Xi\|$	Γ_0	B	Time	LB	UB	Gap	Opt	Nodes	Time	LB	UB	Gap	Opt	Nodes	Δ
10	10	131.0	1	2	145.1	145.1	0%	5	8 (1)	2	146.7	146.7	0%	5	10 (1)	11.35%
		131.0	3	3	153.9	153.9	0%	5	43 (2)	2	155.7	155.7	0%	5	17 (1)	7.86%
		131.0	6	2	155.7	155.7	0%	5	17 (1)	1	155.7	155.7	0%	5	3 (0)	0.00%
	20	128.7	1	5	142.7	142.7	0%	5	11 (2)	5	143.1	143.1	0%	5	11 (2)	2.86%
		128.7	3	8	153.1	153.1	0%	5	53 (2)	8	153.3	153.3	0%	5	62 (2)	0.82%
		128.7	6	5	154.2	154.2	0%	5	16 (1)	3	154.2	154.2	0%	5	5 (0)	0.00%
	30	125.7	1	6	140.1	140.1	0%	5	9 (1)	6	140.1	140.1	0%	5	11 (2)	0.00%
		125.7	3	13	150.0	150.0	0%	5	53 (3)	14	150.2	150.2	0%	5	84 (2)	0.82%
		125.7	6	8	152.6	152.6	0%	5	22 (1)	6	152.6	152.6	0%	5	21 (1)	0.00%
20	10	289.9	1	6	305.3	305.3	0%	5	21 (3)	6	309.3	309.3	0%	5	21 (4)	25.97%
		289.9	3	19	328.6	328.6	0%	5	430 (9)	30	348.1	348.1	0%	5	1017 (6)	50.39%
		289.9	6	72	346.2	346.2	0%	5	2843 (13)	2	351.0	351.0	0%	5	5 (0)	8.53%
	20	258.1	1	12	276.2	276.2	0%	5	21 (5)	15	277.5	277.5	0%	5	21 (5)	7.18%
		258.1	3	60	302.0	302.0	0%	5	655 (12)	102	316.0	316.0	0%	5	1209 (11)	31.89%
		258.1	6	284	322.3	322.3	0%	5	4958 (20)	6	331.4	331.4	0%	5	17 (1)	14.17%
	30	249.3	1	18	268.0	268.0	0%	5	21 (4)	23	270.7	270.7	0%	5	21 (5)	14.44%
		249.3	3	94	292.0	292.0	0%	5	669 (11)	191	299.8	299.8	0%	5	1221 (10)	18.27%
		249.3	6	432	310.5	310.5	0%	5	4541 (14)	278	319.4	319.4	0%	5	2609 (4)	14.54%
30	10	404.8	1	11	432.9	432.9	0%	5	31 (7)	14	442.0	442.0	0%	5	31 (10)	32.38%
		404.8	3	385	467.7	467.7	0%	5	1885 (18)	169	508.6	508.6	0%	5	3913 (15)	65.02%
		404.8	6	1918	493.0	493.0	0%	5	46378 (34)	3	516.3	516.3	0%	5	16 (1)	26.42%
	20	399.0	1	13	424.6	424.6	0%	5	31 (5)	26	426.1	426.1	0%	5	31 (8)	5.86%
		399.0	3	166	455.1	455.1	0%	5	2336 (21)	593	479.4	479.4	0%	5	4405 (30)	43.32%
		399.0	6	3023	483.7	483.7	0%	5	44727 (35)	48	517.6	517.6	0%	5	418 (5)	40.02%
	30	388.1	1	26	412.8	412.8	0%	5	31 (8)	38	414.7	414.7	0%	5	31 (7)	7.69%
		388.1	3	323	443.1	443.1	0%	5	3006 (27)	1387	459.4	459.4	0%	5	4442 (40)	29.64%
		388.1	6	5709	475.3	491.7	3%	3	55483 (34)	1974	513.6	513.6	0%	5	12928 (13)	43.92%