Distributionally robust optimization under endogenous uncertainty with an application in retrofitting planning

Xuan Vinh Doan

Warwick Business School, University of Warwick, Coventry, CV4 7AL, United Kingdom, xuan.doan@wbs.ac.uk

Abstract

Endogenous uncertainty concerns uncertainty which is dependent of decisions such as link failure in the retrofitting planning application. We propose a marginal-based distributionally robust optimization framework for integer stochastic optimization with decision-dependent discrete distributions that can be applied for the retrofitting planning application. We show that the resulting model can be formulated as a mixed-integer linear optimization problem. In order to solve the problem, we develop a constraint generation algorithm given the exponentially large number of constraints. Numerical results for the retrofitting planning application show that the proposed algorithm once tailored can solve the problem efficiently.

Keywords: Stochastic programming, distributionally robust optimization, endogenous uncertainty, retrofitting planning

1. Introduction

Optimization under uncertainty concerns how to make (optimal) decisions when there is uncertainty in problem parameters and data (Diwekar [6]). Uncertainty is traditionally represented with distributional information of random parameters in stochastic optimization (see, e.g., Birge and Louveaux [3]). They could be stock prices in the application of portfolio management, customer demands in revenue management, or wind speed in energy management. In general, these random parameters are *exogenous*, i.e., they are not affected by decisions. However, there are cases when decisions can influence (distributional) information of these random parameters, which implies *endogenous* or *decision-dependent* uncertainty. For example, if a bridge is retrofitted (planning decisions), its survivability, i.e., the probability that it has no damage after a natural disaster, increases. This is the first type of endogenous uncertainty where decisions affect the probability distribution of random parameters such as the survival probabilities in the above example. The second type of endogenous uncertainty happens when decisions affect the resolution of uncertainty, i.e, when random parameters are realized. This type of endogenous uncertainty usually occurs in the multi-stage setting in which different decisions result in the realization of different parameters at different stages. For example, complete information of an oil field is only obtained when a facility is installed at that field as an investment decision in one particular period (stage) while the information of other unexplored oil fields remains unknown (see, e.g., Goel and Grossmann [13]).

Stochastic optimization models with endogenous uncertainty are more difficult to handle in general due to various technical difficulties introduced by the dependence of distributions of random parameters on decisions such as the potential loss of convexity in these models (see, e.g., Dupačová [9]). Jonsbråten et al. [17] were among the first to investigate stochastic optimization with endogenous uncertainty, which initially focuses mainly on the second type of endogenous uncertainty. They developed implicit enumeration algorithms for the models in which the realization of random parameters only depends on first-stage decisions. More recently, Goel and Grossmann ([12]) and Gupta and Grossmann ([14, 15]) have investigated further this type of endogenous uncertainty and proposed different solution approaches including non-anticipativity constraint relaxation and decomposition-based approximation algorithms. Vayanos et al. [32] studied decision rules for multi-stage stochastic optimization problems with endogenous uncertainty. As for the first type of endogenous uncertainty, Peeta et al. [26] handled the resulting highly non-linear models with linear approximation while Flach and Poggi [11] applied other convexification techniques to approximate them. Prestwich et al. [27] used the idea of distribution shaping and scenario bundling to handle potentially large scenario sets in these models and they were able to solve them efficiently without any approximation. More recently, Hellemo [16] considered a combined type of decision-dependent uncertainty and applied it in the context of oil field exploration.

In addition to stochastic optimization, robust optimization is another research area in optimization under uncertainty which follows the principle of "immunized against worst case" (see, e.g., Ben-Tal et al. [1]) instead of expected performance. Robust optimization assumes uncertain parameters belong to *uncertainty sets* without any distributional information. Distributionally robust optimization, on the other hand, makes the assumption that random parameters follow unknown probability distributions which belong to *ambiguity sets*. Distributionally robust optimization or robust/minimax stochastic optimization was first investigated by Žáčková [34], and has been studied extensively more recently within the research communities of both stochastic and robust optimization. With respect to endogenous uncertainty, there are only few very recent research publications discussing decision-dependent uncertainty sets for robust optimization models (see, e.g., Nohadani and Sharma [22], Lappas and Gounaris [18] and references therein). Similarly, there has not much research focussing on distributionally robust optimization with endogenous uncertainty. Royset and Wets [29] investigated the approximation of general optimization problems under stochastic ambiguity (which includes both endogenous and exogenous uncertainty) using cummulative distribution functions and their hypo distance as a metric to establish convergence results. Zhang et al. [35] analyzed the stability of a general distributionally robust optimization problem under endogenous uncertainty with parametric ambiguity sets. Noyan et al. [23] studied the earth mover's distance-based ambiguity sets for decision-dependent distributions while Luo and Mehrotra [19] extended the framework of distributionally robust optimization with decision-dependent parametric ambiguity sets. In this paper, motivated by the stochastic optimization problem under the first type of endogenous uncertainty considered by [26], we focus on a different distributionally robust optimization model under endogenous uncertainty which focuses on probability dependence of decision-dependent distributions.

Contributions and paper outline

Specifically, our contributions and the structure of the paper are as follows.

(1) We propose a new model of distributionally robust optimization under the first type of endogenous uncertainty for integer stochastic optimization problems in Section 2. More specifically, we are going to use Fréchet classes of distributions, i.e., classes of distributions defined by known marginal distributions, as ambiguity sets in the proposed model. We show that the resulting model can be reformulated as a mixed-integer linear optimization problem.

(2) We provide a general constraint generation algorithm to solve the proposed distributionally robust optimization model in Section 3 to handle exponentially large number of constraints. Numerical results are reported for the retrofitting planning application in Section 4 with a tailored algorithm which can generate constraints efficiently by exploiting some structural properties of the retrofitting problem.

2. Mathematical Model

Stochastic optimization problems with the first type of endogenous uncertainty concerns decision making with decision-dependent probability distributions. In the retrofitting planning application studied by Peeta et al. [26], the survival probability of links in a transport network depends on retrofitting decisions. More concretely, let us consider a network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of links. We are concerned about the traversal cost between origin-destination (OD) pairs $(o, d) \in \mathcal{OD}$ within the network after a natural disaster happens, where $\mathcal{OD}$ is the set of all OD pairs in the network. The traversal cost between two nodes o and d in the network is computed as the total travel cost on the shortest path connected o and d. The uncertainty is represented by the state of the transport network after a disaster happens. Each link in the network is either operational or non-operational with complete damage, which in general, affects the traversal costs between OD pairs. Retrofitting a particular link would increase its survival probability, which in turn would affect the probability distribution of the random post-disaster state of the transport network. We aim to determine which links in the network to retrofit so as to minimize the expected traversal cost given the random post-disaster network state. This retrofitting planning problem can be written as the following integer stochastic optimization problem with decision-dependent discrete probability distributions:

$$\begin{array}{rl} \min\limits_{\boldsymbol{x}} & \mathbb{E}_{P(\boldsymbol{x})}\left[f(\boldsymbol{x}, \xi)\right] \\ \text{s.t.} & \boldsymbol{x} \in \mathcal{X}, \end{array} \tag{1}$$

which will be explained in detail next.

The decision variables $\boldsymbol{x} \in \mathcal{X}_0^n$, where $\mathcal{X}_0 = \{0, 1\}$ and $n = |\mathcal{E}|$, indicate whether to retrofit links in a transport network or not. The feasible set $\mathcal{X} \subseteq \mathcal{X}_0^n$, which is defined by additional constraints. For example, if there is a retrofitting budget B and each link i, $i = 1, \ldots, n$, needs a cost b_i to retrofit, the budget constraint $\sum_{i=1}^{n} b_i x_i \leq B$ would be used to define $\mathcal{X}$ in the retrofitting planning application. The set $\mathcal{X}_0$ can be extended to a finite set of more than two options to indicate different levels of investment.

The uncertainty ξ is represented by scenarios $\boldsymbol{s} \in \mathcal{S}_0^n$, where $\mathcal{S}_0 = \{0, 1\}$, which indicates whether a link is operational ($s = 1$) or not ($s = 0$) after a natural disaster. The set $\mathcal{S}_0$ can be generalized to a finite set of possible states including different levels of damages caused by a natural disaster (see, e.g., Chang et al. [4]).

Given a scenario $\boldsymbol{s} \in \mathcal{S}_0^n$, for the retrofitting planning application with a single OD pair, $f(\boldsymbol{x}, \boldsymbol{s})$ is the traversal cost between the origin and destination of the OD pair within the transport network given the network state $\boldsymbol{s}$. (Note that for multiple OD pairs, it is straightforward to generalize the objective to be the weighted sum of traversal costs of all OD pairs and it will be discussed in detail later.) The traversal cost between two nodes in the network is the travel cost on the shortest path connecting these two nodes, which can be computed using the following network flow problem:

$$
\begin{aligned}
f(\boldsymbol{x}, \boldsymbol{s}) = \min_{\boldsymbol{0} \leq \boldsymbol{w} \leq \bar{\boldsymbol{w}}} \quad & \sum_{i=1}^{n} c_i(x_i, s_i) \cdot w_i \\
\text{s.t.} \quad & \boldsymbol{w} \in \mathcal{W},
\end{aligned}
\tag{2}
$$

where $\boldsymbol{w}$ are flow decision variables, $\boldsymbol{w} \in [0, 1]^n$ if the links are directed. If the links are undirected, the cost function $f(\boldsymbol{x}, \boldsymbol{s})$ can also be formulated as shown in (2), which will be discussed in detail later. The feasible set $\mathcal{W}$ is defined by flow conservation constraints, which can be represented explicitly as $\boldsymbol{A}\boldsymbol{w} = \boldsymbol{b}$, and the cost $c_i(x_i, s_i)$ is the travel cost of link $i \in \mathcal{E}$ given its post-disaster state. In retrofitting planning application, when the link i is non-operational ($s_i = 0$), $c_i(x_i, 0)$ is set to be large enough, which implies $c_i(x_i, 0) \geq c_i(x_i, 1)$, where $c_i(x_i, 1)$ is the travel cost when the link i is operational ($s_i = 1$). We therefore make the following formal assumption:

Assumption 1. *The cost function $c_i(x, \cdot)$, $x \in \mathcal{X}_0$, is non-increasing in $s \in \mathcal{S}_0$, i.e., $c_i(x, 0) \geq c_i(x, 1)$ for all $i = 1, \ldots, n$.*

The probability distributions of the random state of each individual link i, $i = 1, \ldots, n$, are decision-dependent or endogenous. Let $p_i(x_i, s_i) \geq 0$ be the probability of link i to be in state $s_i \in \mathcal{S}_0$ if $x_i \in \mathcal{X}_0$ has been selected as the decision, which constitutes the decision-dependent distribution $P_i(x_i)$

of the random state of link i, $i = 1, \ldots, n$. Clearly, $\sum_{s \in \mathcal{S}_0} p_i(x, s) = 1$ for all $x \in \mathcal{X}_0$ and i, $i = 1, \ldots, n$. Naturally, one should have $p_i(0, 1) \leq p_i(1, 1)$ in this application, i.e., retrofitting actions ($x = 1$) increase link survivability, the probability of a link to be operational after a natural disaster ($s = 1$).

In the retrofit planning application, link survivability is normally assumed to be known as it can be computed by structural engineers for each and every link using domain-specific information (see, e.g., [26]). Given the known decision-dependent probability distributions $P_i(x)$ of individual items, one still need to define the (decision-dependent) joint probability distribution $P(\boldsymbol{x})$ of the random network state $\boldsymbol{s}$. Peeta et al. [26] and subsequently, Prestwich et al. [27], assume that $P(\boldsymbol{x})$ is an independent joint probability distribution, i.e., $P(\boldsymbol{x}) = \times_{i=1}^{n} P_i(x_i)$, which makes Problem (1) highly nonlinear and difficult to solve due to the computation of the expectation with respect to this joint probability distribution. Peeta et al. [26] had to use linear approximation while Prestwich et al. [27] used distribution shaping and scenario bundling to solve Problem (1) with independent $P(\boldsymbol{x})$. The independence assumption is also quite a strong assumption for the retrofitting planning application given that the survival probability of an individual link not only depends on its own structure but also the properties of the happening natural disaster. For example, damages caused by an earthquake depend on its magnitude and where its epicenter is within the transport network (Chang [4]), which implies the survival provabilities of individual links are not completely independent. In this paper, we relax the independence assumption and consider a distributionally robust optimization model with Fréchet classes of joint probability distributions whose decision-dependent marginal distributions are known. This ambiguity model is chosen given its inherent ability to handle dependence ambiguity when the independence assumption is relaxed for the retrofitting planning application. For applications with decision-dependent uncertainty in which independence assumption holds such as the application of network interdiction, the proposed model might not be relevant. The proposed model is indeed different from the stochastic network interdiction models which concerns random OD pairs instead of the ambiguity of decision-dependent probability distributions (see, e.g., [24] and references therein). On the other hand, the proposed model is different from other disrtributionally robust optimization models ([23, 19]) which require different assumptions such as the availability of a known nominal joint probability distribution $P_0(\boldsymbol{x})$ to construct the corresponding ambiguity sets. The proposed model is instead motivated by the assumption that individual link survival probabilities are known, which is appropriate for the retrofit planning application. We now focus on the development of the proposed distributionally robust optimization model with Fréchet classes of joint probability distributions.

The Fréchet classes of distributions have been investigated by Rüschendorf [30] among others and they are used to evaluate bounds on the cumulative distribution function of a sum of random variables (see, e.g., Embretchs and Puccetti [10]). The related multimarginal optimal transportation problem is also studied extensively (see, e.g., Pass [25]). Distributionally robust optimization models with Fréchet classes of distributions under exogenous uncertainty have been investigated in many applications (see, e.g., [8, 7, 28]). Mathematically, let $\mathcal{P}(\boldsymbol{x})$ be the Fréchet class of distributions with known decision-dependent marginals $P_i(x_i)$ for $i = 1, \dots, n$, i.e.,

$$\mathcal{P}(\boldsymbol{x}) = \{P \in \mathcal{M}(\mathcal{S}_0^n) \,|\, \mathrm{proj}_i(P) = P_i(x_i),\ i = 1, \dots, n\}, \tag{3}$$

where $\mathcal{M}(\mathcal{S}_0^n)$ is the set of probability measures defined on the (finite) discrete set $\mathcal{S}_0^n$ of random network state $\boldsymbol{s}$ and $\mathrm{proj}_i(P)$ is the i-th marginal distribution of the joint probability distribution P. The robust counterpart of Problem (1) can be written as follows:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \max\limits_{P \in \mathcal{P}(\boldsymbol{x})} \mathbb{E}_P\left[f(\boldsymbol{x}, \xi)\right] \\ \text{s.t.} & \boldsymbol{x} \in \mathcal{X}. \end{array} \tag{4}$$

A joint probability distribution $P \in \mathcal{P}(\boldsymbol{x})$ can be characterized by the probabilities $p_{\boldsymbol{s}}$ of scenarios $\boldsymbol{s} \in \mathcal{S}_0^n$. Clearly, $p_{\boldsymbol{s}} \geq 0$ for all $\boldsymbol{s} \in \mathcal{S}_0^n$ and $\sum\limits_{\boldsymbol{s} \in \mathcal{S}_0^n} p_{\boldsymbol{s}} = 1$. The information of marginal distribution $P_i(x_i)$, $i = 1, \dots, n$, can be represented by the following constraints:

$$\sum_{\boldsymbol{s}: s_i = 1} p_{\boldsymbol{s}} = p_i(x_i, 1).$$

We do not need to impose the above constraint for $p_i(x_i, 0)$ as $\sum\limits_{s \in \mathcal{S}_0} p_i(x_i, s) = 1$ for all $i = 1, \dots, n$. Given this presentation of $P \in \mathcal{P}(\boldsymbol{x})$, we are ready to reformulate Problem (4). The inner optimization problem is a linear optimization problem with $p_{\boldsymbol{s}}$ as decision variables:

$$\begin{aligned}
\max_{p_{\boldsymbol{s}}} \quad & \sum_{\boldsymbol{s}\in\mathcal{S}_0^n} f(\boldsymbol{x},\boldsymbol{s})\cdot p_{\boldsymbol{s}} \\
\text{s.t.} \quad & \sum_{\boldsymbol{s}\in\mathcal{S}_0^n} p_{\boldsymbol{s}} = 1, \\
& \sum_{\boldsymbol{s}:s_i=1} p_{\boldsymbol{s}} = p_i(x_i,1), \quad \forall\, i=1,\ldots,n, \\
& p_{\boldsymbol{s}} \geq 0, \quad \forall\, \boldsymbol{s}\in\mathcal{S}_0^n,
\end{aligned} \tag{5}$$

whose dual problem can be written as follows:

$$\begin{aligned}
\min_{u,\boldsymbol{v}} \quad & u + \sum_{i=1}^n p_i(x_i,1)\cdot v_i \\
\text{s.t.} \quad & u + \sum_{i:s_i\neq 0} v_i \geq f(\boldsymbol{x},\boldsymbol{s}), \quad \forall\, \boldsymbol{s}\in\mathcal{S}_0^n,
\end{aligned} \tag{6}$$

where u and v_i for $i=1,\ldots,n$, are dual decision variables. Note that the total number of dual decision variables is only $n+1$ while the number of dual constraints is 2^n, which is exponential large. Given strong linear duality, we obtain the following reformulation of Problem (4):

$$\begin{aligned}
\min_{\boldsymbol{x},u,\boldsymbol{v}} \quad & u + \sum_{i=1}^n p_i(x_i,1)\cdot v_i \\
\text{s.t.} \quad & u + \sum_{i:s_i\neq 0} v_i \geq f(\boldsymbol{x},\boldsymbol{s}), \quad \forall\, \boldsymbol{s}\in\mathcal{S}_0^n, \\
& \boldsymbol{x}\in\mathcal{X}.
\end{aligned} \tag{7}$$

Problem (7) has a non-linear objective function due to the terms $p_i(x_i,1)\cdot v_i$ which involve both decision variables x_i and v_i. To further reformulate the problem, we need the following lemma which provides bounds for the dual decision variables $\boldsymbol{v}$ and in turn allows us to linearize the objective function.

Lemma 1. *There exist optimal solutions of Problem* (7) *such that* $-\Delta f_i \leq v_i \leq 0$ *for all* $i=1,\ldots,n$, *where*

$$\Delta f_i = \max_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{s}_{-i}\in\mathcal{S}_0^{n-1}} \{f(\boldsymbol{x},(0,\boldsymbol{s}_{-i})) - f(\boldsymbol{x},(1,\boldsymbol{s}_{-i}))\}, \tag{8}$$

with $\boldsymbol{s}_{-i}$ *denoting the vector of all elements of* $\boldsymbol{s}$ *except* s_i.

Proof. We first show that there exist optimal solutions such that $v_i \leq 0$ for all $i = 1, \ldots, n$. We fix the (optimal) solution $\boldsymbol{x}$ and let us consider an optimal solution $(u, \boldsymbol{v})$ with $v_i > 0$ for all $i \in \mathcal{I}_p \subseteq \{1, \ldots, n\}$. We construct another solution $(u', \boldsymbol{v}')$ with $u' = u$, $v'_i = 0$ for all $i \in \mathcal{I}_p$ and $v'_i = v_i$ otherwise. For all $\boldsymbol{s} \in \mathcal{S}_0^n$, let consider $\boldsymbol{s}' \in \mathcal{S}_0^n$ such that $s'_i = 0$ for all $i \in \mathcal{I}_p$ and $s'_i = s_i$ otherwise, we have:

$$u + \sum_{i \notin \mathcal{I}_p : s'_i \neq 0} v_i \geq f(\boldsymbol{x}, \boldsymbol{s}').$$

Thus we have:

$$u' + \sum_{i : s_i \neq 0} v'_i \geq f(\boldsymbol{x}, \boldsymbol{s}') \geq f(\boldsymbol{x}, \boldsymbol{s}).$$

The second inequality is due to the monotonicity of the cost function $c_i(x_i, \cdot)$ stated in Assumption 1 and how $f(\boldsymbol{x}, \boldsymbol{s})$ is computed in (2). It shows the modified solution $(u', \boldsymbol{v}')$ is feasible. In addition, given that $p_i(x_i, s_i) \geq 0$ for all x_i and s_i, the new solution $(u', \boldsymbol{v}')$ (together with $\boldsymbol{x}$) is also optimal. It implies that there exist optimal solutions such that $v_i \leq 0$ for all $i = 1, \ldots, n$.

Now, among optimal solutions with non-positive v_i, we will show that there exist solutions such that $v_i \geq -\Delta f_i$ for all $i = 1, \ldots, n$. Let consider an optimal solution with $v_{\bar{i}} < -\Delta f_{\bar{i}} \leq 0$ for some $\bar{i}$. We will construct another optimal solution $(u', \boldsymbol{v}')$ as follows. Let $u' = u + \Delta f_{\bar{i}} + v_{\bar{i}} < u$ and $v'_{\bar{i}} = -\Delta f_{\bar{i}}$. Clearly $v_{\bar{i}} \leq v'_{\bar{i}} \leq 0$. Finally, let $v'_i = v_i$ for all $i \neq \bar{i}$. For all $\boldsymbol{s} \in \mathcal{S}_0^n$, let consider $\boldsymbol{s}' \in \mathcal{S}_0^n$ such that $s'_{\bar{i}} = 0$ and $s'_i = s_i$ for $i \neq \bar{i}$. Similarly, let $\boldsymbol{s}'' \in \mathcal{S}_0^n$ such that $s''_{\bar{i}} = 1$ and $s''_i = s_i$ for $i \neq \bar{i}$. We have

$$u + \sum_{i \neq \bar{i} : s'_i \neq 0} v_i \geq f(\boldsymbol{x}, \boldsymbol{s}'),$$

and

$$u + v_{\bar{i}} + \sum_{i \neq \bar{i} : s''_i \neq 0} v_i \geq f(\boldsymbol{x}, \boldsymbol{s}'').$$

If $s_{\bar{i}} = 1$, i.e., $\boldsymbol{s} = \boldsymbol{s}''$, then

$$u' + v'_{\bar{i}} + \sum_{i \neq \bar{i} : s_i \neq 0} v'_i = u + v_{\bar{i}} + \sum_{i \neq \bar{i} : s_i \neq 0} v_i \geq f(\boldsymbol{x}, \boldsymbol{s}).$$

Now, if $s_{\bar{i}} = 0$, then $\boldsymbol{s} = \boldsymbol{s}'$ and we have:

$$\begin{aligned} u' + \sum_{i \neq \bar{i}: s_i \neq 0} v'_i &= u + \Delta f_{\bar{i}} + v_{\bar{i}} + \sum_{i \neq \bar{i}: s_i \neq 0} v_i \\ &\geq \Delta f_{\bar{i}} + f(\boldsymbol{x}, \boldsymbol{s}'') \\ &\geq f(\boldsymbol{x}, \boldsymbol{s}), \end{aligned}$$

where the second inequality comes from the definition of Δf_i. It shows that the new solution $(u', \boldsymbol{v}')$ is feasible in both cases. In addition, we have: $u - u' = v'_{\bar{i}} - v_{\bar{i}}$ and $p_{\bar{i}}(x_{\bar{i}}, 1) \leq 1$. It implies that the new solution $(u', \boldsymbol{v}')$ (together with $\boldsymbol{x}$) is also optimal. The lemma is then proved. □

Lemma 1 shows the relationship between the dual decision variable v_i and Δf_i, which can be interpreted as the highest potential increase in cost due to the failure of link i given any scenario and any retrofitting decision in this retrofitting planning application. We now use this result to reformulate the robust problem (4) as stated in the following theorem.

Theorem 1. *Under Assumption 1, the robust problem* (4) *can be reformulated as the following mixed-integer linear optimization problem:*

$$\begin{aligned} \min_{\boldsymbol{x},u,\boldsymbol{v},\boldsymbol{y},\boldsymbol{w},\boldsymbol{q}} \quad & u + \sum_{i=1}^{n} \left[p_i(0,1) \cdot v_i + (p_i(1,1) - p_i(0,1)) \cdot y_i \right] \\ \text{s.t.} \quad & y_i \geq v_i, \quad \forall\, i = 1, \ldots, n \,:\, p_i(1,1) \geq p_i(0,1), \\ & y_i \geq -\Delta f_i \cdot x_i, \quad \forall\, i = 1, \ldots, n \,:\, p_i(1,1) \geq p_i(0,1), \\ & y_i \leq 0, \quad \forall\, i = 1, \ldots, n, \,:\, p_i(1,1) < p_i(0,1), \\ & y_i \leq v_i + \Delta f_i \cdot (1 - x_i), \quad \forall\, i = 1, \ldots, n, \,:\, p_i(1,1) < p_i(0,1), \\ & -\Delta \boldsymbol{f} \leq \boldsymbol{v} \leq \boldsymbol{0}, \\ & u + \sum_{i: s_i \neq 0} v_i \geq \sum_{i=1}^{n} \left[c_i(0, s_i) \cdot w_i(\boldsymbol{s}) + (c_i(1, s_i) - c_i(0, s_i)) \cdot q_i(\boldsymbol{s}) \right], \\ & \quad \forall\, \boldsymbol{s} \in \mathcal{S}_0^n, \\ & q_i(\boldsymbol{s}) \geq 0, \quad \forall\, i = 1, \ldots, n, \, \boldsymbol{s} \in \mathcal{S}_0^n \,:\, c_i(1, s_i) \geq c_i(0, s_i), \\ & q_i(\boldsymbol{s}) \geq w_i(\boldsymbol{s}) + \bar{w}_i \cdot (x_i - 1), \\ & \quad \forall\, i = 1, \ldots, n, \, \boldsymbol{s} \in \mathcal{S}_0^n \,:\, c_i(1, s_i) \geq c_i(0, s_i), \\ & q_i(\boldsymbol{s}) \leq w_i(\boldsymbol{s}), \quad \forall\, i = 1, \ldots, n, \, \boldsymbol{s} \in \mathcal{S}_0^n \,:\, c_i(1, s_i) < c_i(0, s_i), \\ & q_i(\boldsymbol{s}) \leq \bar{w}_i \cdot x_i, \quad \forall\, i = 1, \ldots, n, \, \boldsymbol{s} \in \mathcal{S}_0^n \,:\, c_i(1, s_i) < c_i(0, s_i), \\ & \boldsymbol{w}(\boldsymbol{s}) \in \mathcal{W}, \quad \forall\, \boldsymbol{s} \in \mathcal{S}_0^n, \\ & \boldsymbol{0} \leq \boldsymbol{w}(\boldsymbol{s}) \leq \bar{\boldsymbol{w}}, \quad \forall\, \boldsymbol{s} \in \mathcal{S}_0^n, \\ & \boldsymbol{x} \in \mathcal{X}. \end{aligned} \tag{9}$$

Proof. Given that $x_i \in \mathcal{X}_0 = \{0, 1\}$, we have:

$$p_i(x_i, 1) = p_i(1, 0) \cdot (1 - x_i) + p_i(1, 1) \cdot x_i,$$

which means the terms $p_i(x_i, 1) \cdot v_i$ of the objective function can be written as

$$p_i(x_i, 1) \cdot v_i = p_i(1, 0) \cdot v_i + (p_i(1, 1) - p_i(1, 0)) \cdot x_i \cdot v_i.$$

The non-linearity of the objective function is expressed through the products $x_i \cdot v_i$. Given Lemma 1, we can focus on the solutions which satisfy the condition $-\Delta f_i \leq v_i \leq 0$ for all $i = 1, \dots, n$. Given this condition, the non-linear term $x_i \cdot v_i$ in the minimization problem (4) can be reformulated with an auxiliary decision variable y_i. If $p_i(1, 1) \geq p_i(1, 0)$, we impose the following constraints:

$$\begin{cases} y_i \geq v_i, \\ y_i \geq -\Delta f_i \cdot x_i. \end{cases}$$

Clearly, if $x_i = 0$, $y_i \geq \max\{v_i, 0\} = 0$; otherwise, $y_i \geq \max\{v_i, -\Delta f_i\} = v_i$ or equivalently, $(p_i(1, 1) - p_i(0, 1)) \cdot y_i \geq (p_i(1, 1) - p_i(0, 1)) \cdot x_i \cdot v_i$. Similarly, if $p_i(1, 1) < p_i(1, 0)$, we impose the following constraints:

$$\begin{cases} y_i \leq v_i + \Delta f_i \cdot (1 - x_i), \\ y_i \leq 0. \end{cases}$$

If $x_i = 0$, $y_i \leq \min\{v_i + \Delta f_i, 0\} = 0$; otherwise, $y_i \leq \min\{v_i, 0\} = v_i$ or equivalently, $y_i \leq x_i \cdot v_i$, which implies $(p_i(1, 1) - p_i(0, 1)) \cdot y_i \geq (p_i(1, 1) - p_i(0, 1)) \cdot x_i \cdot v_i$.

Now consider the main constraints $u + \sum_{i:s_i \neq 0} v_i \geq f(\boldsymbol{x}, \boldsymbol{s})$, we have:

$$\begin{aligned} f(\boldsymbol{x}, \boldsymbol{s}) = \min_{\mathbf{0} \leq \boldsymbol{w} \leq \bar{\boldsymbol{w}}} \quad & \sum_{i=1}^{n} c_i(x_i, s_i) \cdot w_i \\ \text{s.t.} \quad & \boldsymbol{w} \in \mathcal{W}. \end{aligned}$$

Thus, in order to satisfy these constraints, there should be a feasible solution $\boldsymbol{w}(\boldsymbol{s})$, i.e., $\boldsymbol{w}(\boldsymbol{s}) \in \mathcal{W}$ and $\mathbf{0} \leq \boldsymbol{w}(\boldsymbol{s}) \leq \bar{\boldsymbol{w}}$, for all $\boldsymbol{s} \in \mathcal{S}_0^n$ such that

$$u + \sum_{i:s_i \neq 0} v_i \geq \sum_{i=1}^{n} c_i(x_i, s_i) \cdot w_i(\boldsymbol{s}).$$

Again, we have:

$$c_i(x_i, s_i) = c_i(0, s_i) \cdot (1 - x_i) + c_i(1, s_i) \cdot x_i,$$

which means the terms $c_i(x_i, s_i) \cdot w_i(\boldsymbol{s})$ can be written as

$$c_i(x_i, s_i) \cdot w_i(\boldsymbol{s}) = c_i(0, s_i) \cdot w_i(\boldsymbol{s}) + (c_i(1, s_i) - c_i(0, s_i)) \cdot x_i \cdot w_i(\boldsymbol{s}).$$

Similar to the reformulation of $x_i \cdot v_i$, the non-linear terms $x_i \cdot w_i(\boldsymbol{s})$ can be reformulated with the auxiliary decision variables $q_i(\boldsymbol{s})$ given that $0 \leq w_i(\boldsymbol{s}) \leq \bar{w}_i$. If $c_i(1, s_i) \geq c_i(0, s_i)$, we impose the following constraints:

$$\begin{cases} q_i(\boldsymbol{s}) \geq 0, \\ q_i(\boldsymbol{s}) \geq w_i(\boldsymbol{s}) + \bar{w}_i \cdot (x_i - 1). \end{cases}$$

If $c_i(1, s_i) < c_i(0, s_i)$, the following constraints will be imposed:

$$\begin{cases} q_i(\boldsymbol{s}) \leq w_i(\boldsymbol{s}), \\ q_i(\boldsymbol{s}) \leq \bar{w}_i \cdot x_i. \end{cases}$$

It shows that the robust problem (4) is equivalent to the mixed-integer linear optimization problem (9) with the introduction of additional decision variables $\boldsymbol{y}$, $\boldsymbol{w}(\boldsymbol{s})$, and $\boldsymbol{q}(\boldsymbol{s})$. □

Remark 1. *i) The mixed-integer linear optimization problem* (9) *requires the lower bounds $-\Delta f_i$ of v_i for all $i = 1, \ldots, n$. The reformulation would remain the same for other lower bounds; therefore, instead of using Δf_i defined in* (8)*, which is difficult to compute in general, we are going to use*

$$\Delta \bar{f} = \max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}, \boldsymbol{0}) - \min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}, \boldsymbol{e}), \tag{10}$$

where $\boldsymbol{e}$ is the vector of all ones, for numerical case studies later. Indeed,

$$\begin{aligned} \Delta f_i &= \max_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{s}_{-i} \in \mathcal{S}_0^{n-1}} \{ f(\boldsymbol{x}, (0, \boldsymbol{s}_{-i})) - f(\boldsymbol{x}, (1, \boldsymbol{s}_{-i})) \} \\ &\leq \max_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{s}_{-i} \in \mathcal{S}_0^{n-1}} f(\boldsymbol{x}, (0, \boldsymbol{s}_{-i})) - \min_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{s}_{-i} \in \mathcal{S}_0^{n-1}} f(\boldsymbol{x}, (1, \boldsymbol{s}_{-i})) \\ &\leq \max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}, \boldsymbol{0}) - \min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}, \boldsymbol{e}) = \Delta \bar{f}. \end{aligned}$$

The second inequality is due to the definition of $f(\boldsymbol{x}, \boldsymbol{s})$ given in (2) *and Assumption 1. $\Delta \bar{f}$ can be computed more easily as compared to Δf_i given that $\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}, \boldsymbol{s})$ and $\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}, \boldsymbol{s})$ can be solved using* (2) *together with linear duality and the reformulation of the non-linear terms $x_i \cdot w_i$.*

ii) The results in Theorem 1 (and Lemma 1) are showed for $\mathcal{X}_0 = \{0, 1\}$ and $\mathcal{S}_0 = \{0, 1\}$ for clarity of exposition. They can be generalized when $\mathcal{X}_0$ and $\mathcal{S}_0$ are extended to finite discrete sets under the assumption of the monotonicity of $c_i(x, \cdot)$ as shown in Assumption 1. The properties of dual decision variables $\boldsymbol{v}$ in Lemma 1 remain valid and the reformulation of non-linear terms can be handled by introducing additional binary decision variables to represent finite discrete sets.

The result in Theorem 1 shows that the robust problem (4) can be reformulated as a mixed-integer linear optimization problem. The number of decision variables $\boldsymbol{w}(\boldsymbol{s})$ is proportional to the number of scenarios, which is 2^n. If $c_i(x_i, s_i) \equiv c_i(s_i)$ for all $i = 1, \ldots, n$, there is no need to introduced $\boldsymbol{w}(\boldsymbol{s})$ if $f(\boldsymbol{s})$ can be computed upfront and used as the right-hand sides of the main constraints $u + \sum_{i:s_i \neq 0} v_i(s_i) \geq f(\boldsymbol{s})$ for all $\boldsymbol{s} \in \mathcal{S}_0^n$. Even in this case of decision-independent cost function $f(\boldsymbol{s})$, the high computational complexity of the robust problem (4) remains with the exponential number of these main constraints. In the next section, we are going to discuss how to solve this mixed-integer linear optimisation reformulation of the robust problem (4).

3. Computational Framework

The robust problem (4) has an exponential number of the main constraints, one for each scenario $\boldsymbol{s} \in \mathcal{S}_0^n$. Note that these main constraints are constructed based on the pair of primal-dual linear optimization problems (5) and (6). It is well-known that given a solution $\boldsymbol{x}$, there exist optimal distributions $P^*(\boldsymbol{x})$ with small supports based on the theory of linear programming (see, e.g., Dantzig [5]). It implies that one only requires the main constraints for a small number of scenarios $\boldsymbol{s} \in \mathcal{S}_0^n$. However, finding the active set of scenarios without knowing the solution $\boldsymbol{x}$ is difficult. A common approach to handle optimization problems with a large number of constraints is the constraint generation method. It has been used to solve large-scale robust optimization problems (see, e.g., [20, 31] and references therein). The key idea of the constraint generation method is to solve the *master problem* with a subset of scenarios and iteratively add violated scenarios by solving the *separation* problem. More precisely, in each iteration k, the master problem is a relaxed problem of (9) in which $\mathcal{S}_0^n$ is replaced by a subset $\mathcal{S}^{(k)} \subset \mathcal{S}_0^n$:

$$
\begin{aligned}
(\mathrm{MP})_k : \min_{\boldsymbol{x},u,\boldsymbol{v},\boldsymbol{y}} \quad & u + \sum_{i=1}^{n} \left[p_i(0,1) \cdot v_i + (p_i(1,1) - p_i(0,1)) \cdot y_i\right] \\
\text{s.t.} \quad & y_i \geq v_i, \quad \forall\, i = 1, \ldots, n \,:\, p_i(1,1) \geq p_i(0,1), \\
& y_i \geq -\Delta f_i \cdot x_i, \quad \forall\, i = 1, \ldots, n \,:\, p_i(1,1) \geq p_i(0,1), \\
& y_i \leq 0, \quad \forall\, i = 1, \ldots, n, \,:\, p_i(1,1) < p_i(0,1), \\
& y_i \leq v_i + \Delta f_i \cdot (1 - x_i), \forall\, i = 1, \ldots, n, \,:\, p_i(1,1) < p_i(0,1), \\
& -\Delta \boldsymbol{f} \leq \boldsymbol{v} \leq \boldsymbol{0}, \\
& u + \sum_{i: s_i \neq 0} v_i \geq f(\boldsymbol{x}, \boldsymbol{s}), \quad \forall\, \boldsymbol{s} \in \mathcal{S}^{(k)}, \\
& \boldsymbol{x} \in \mathcal{X}.
\end{aligned}
\tag{11}
$$

Given the optimal solution $\left(\boldsymbol{x}^{(k)}, u^{(k)}, \boldsymbol{v}^{(k)}\right)$ obtained from $(\mathrm{MP})_k$, the separation problem is used to determine whether there is any violated constraint:

$$
(\mathrm{SP})_k : V_k = \min_{\boldsymbol{s} \in \mathcal{S}_0^n} \left\{ u^{(k)} + \sum_{i: s_i \neq 0} v_i^{(k)} - f(\boldsymbol{x}^{(k)}, \boldsymbol{s}) \right\}. \tag{12}
$$

If $V_k \geq 0$, there is no violated constraint, which means the current solution is optimal. Otherwise, add the optimal scenario $\boldsymbol{s}_k$ obtained from $(\mathrm{SP})_k$ to the set of scenarios, $\mathcal{S}^{(k+1)} = \mathcal{S}^{(k)} \cup \{\boldsymbol{s}_k\}$ and repeat. The detailed algorithm is written as Algorithm 1. In order to execute Algorithm 1, we need to

Algorithm 1 Constraint Generation Algorithm for Problem (9)

1: Initialize with $k \leftarrow 0$ and $\mathcal{S}^{(0)}$.
2: **loop**
3: Solve the master problem (11) with $\mathcal{S}^{(k)}$ to obtain an optimal solution $\left(\boldsymbol{x}^{(k)}, u^{(k)}, \boldsymbol{v}^{(k)}\right)$.
4: Solve the separation problem (12) with $\left(\boldsymbol{x}^{(k)}, u^{(k)}, \boldsymbol{v}^{(k)}\right)$ to obtain an optimal solution $\boldsymbol{s}_k$ and the optimal value V_k.
5: **if** $V_k < 0$ **then**
6: Update $k \leftarrow k + 1$ and $\mathcal{S}^{(k)} \leftarrow \mathcal{S}^{(k)} \cup \{\boldsymbol{s}_k\}$.
7: **else**
8: Stop. Return $\boldsymbol{x}^{(k)}$ as the optimal solution obtained.

solve the master problem (11), which is formulated as a mixed-integer linear optimization problem. The separation problem (12) can also be formulated as a mixed-integer linear optimization problem using linear duality. Without loss of generality, let us assume that $\mathcal{W}$ is explicitly represented as $\mathcal{W} = \{\boldsymbol{w} \mid \boldsymbol{A}\boldsymbol{w} = \boldsymbol{b}\}$. The following proposition shows how to reformulate the separation problem.

Proposition 1. *Given a solution $\left(\boldsymbol{x}^{(k)}, u^{(k)}, \boldsymbol{v}^{(k)}\right)$, the separation problem* (12) *can be reformulated as the following mixed-integer linear optimization problem:*

$$
\begin{aligned}
(SP)_k : V_k = \min_{\boldsymbol{s},\boldsymbol{d},\boldsymbol{e}} \quad & u^{(k)} + \sum_{i=1}^{n} v_i^{(k)} \cdot s_i - \left(\boldsymbol{b}^T\boldsymbol{d} + \bar{\boldsymbol{w}}^T\boldsymbol{e}\right) \\
\text{s.t.} \quad & \boldsymbol{A}_i^T\boldsymbol{d} + e_i \le c_i(x_i^{(k)}, 0) + (c_i(x_i^{(k)}, 1) - c_i(x_i^{(k)}, 0)) \cdot s_i, \\
& \quad \forall\, i = 1, \dots, n, \\
& \boldsymbol{e} \le \boldsymbol{0}, \\
& \boldsymbol{s} \in \mathcal{S}_0^n,
\end{aligned}
\tag{13}
$$

where $\boldsymbol{A}_i$ is the i-th column of the matrix $\boldsymbol{A}$, $i = 1, \dots, n$.

Proof. Given that $\boldsymbol{s} \in \mathcal{S}_0^n$, where $\mathcal{S}_0 = \{0, 1\}$, we can rewrite $\sum_{i:s_i \neq 0} v_i^{(k)}$ as $\sum_{i=1}^{n} v_i^{(k)} \cdot s_i$. The function $f(\boldsymbol{x}^{(k)}, \boldsymbol{s})$ is the optimal objective value of the linear optimization problem (2) whose dual problen can be written as follows if $\mathcal{W} = \{\boldsymbol{w} \mid \boldsymbol{A}\boldsymbol{w} = \boldsymbol{b}\}$:

$$
\begin{aligned}
f(\boldsymbol{x}^{(k)}, \boldsymbol{s}) = \max_{\boldsymbol{d},\boldsymbol{e}} \quad & \boldsymbol{b}^T\boldsymbol{d} + \bar{\boldsymbol{w}}^T\boldsymbol{e} \\
\text{s.t.} \quad & \boldsymbol{A}_i^T\boldsymbol{d} + e_i \le c_i(x_i^{(k)}, s_i), \quad \forall\, i = 1, \dots, n, \\
& \boldsymbol{e} \le \boldsymbol{0}.
\end{aligned}
\tag{14}
$$

Finally, we can compute$c_i(x_i^{(k)}, s_i) = c_i(x_i^{(k)}, 0) + (c_i(x_i^{(k)}, 1) - c_i(x_i^{(k)}, 0)) \cdot s_i$. The main constraint $\boldsymbol{A}_i^T\boldsymbol{d} + e_i \le c_i(x_i^{(k)}, s_i)$ in (14) can then be written as

$$
\boldsymbol{A}_i^T\boldsymbol{d} + e_i \le c_i(x_i^{(k)}, 0) + (c_i(x_i^{(k)}, 1) - c_i(x_i^{(k)}, 0)) \cdot s_i,
$$

which indicates that (13) is indeed a reformulation of the separation problem (12). $\square$

The result stated in Proposition 1 shows that the separation problem (12) can be reformulated as a mixed-integer linear optimization problem. Note that one can reformulate the separation problem using the same approach for general finite discret set $\mathcal{S}_0$ by introducing addition binary decision variables to represent $\mathcal{S}_0$. Next, we are going to discuss numerical experiments using Algorithm 1.

4. Numerical Case Studies

We consider the retrofitting planning application studied by Peeta et al. [26]. As discussed in Section 2, there are $n = |\mathcal{E}|$ links in a transport network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, which are considered to be retrofitted ($x_i = 1$) or not ($x_i = 0$), where $\boldsymbol{x} \in \{0,1\}^n$ denote the decision variables. The retrofitting cost for each link i is b_i, which is used to formulate the budget constraint:

$$\sum_{i=1}^{n} b_i x_i \leq B,$$

where B is a retrofitting budget. The uncertainty is represented by whether a link i survives ($s_i = 1$) or fails ($s_i = 0$) after a natural disaster such as an earthquake. Given a scenario $\boldsymbol{s} \in \{0,1\}^n$, the total cost is a weighted sum of traversal costs on the shortest path between several OD pairs $j \in \mathcal{OD}$, $f(\boldsymbol{s}) = \sum_{j \in \mathcal{OD}} \omega_j \cdot f^j(\boldsymbol{s})$, where $\omega_j > 0$, $j \in \mathcal{OD}$, are the weights. Without loss of generality, we can assume that $\sum_{j \in \mathcal{OD}} \omega_j = 1$. The actual traversal cost on link i is $c_i(1)$, i.e., when $s_i = 1$. If the link i fails, i.e., $s_i = 0$, we set $c_i(0) = M$, where M is a large number. By adding a special link between the origin and destination of each OD pair with M as its fixed traversal cost, it is clear that if the origin and destination of an OD pair j is not connected in a scenario $\boldsymbol{s}$, $f^j(\boldsymbol{s}) = M$. Under the setting of multiple OD pairs, the weighted sum of traversal costs still can be computed as a network flow problem with the supply and demand of ω_j at the origin and destination of OD pair j, $j \in \mathcal{OD}$, respectively. When the network $\mathcal{G}$ is undirected as set in Peeta et al. [26], the network flow problem can be modified with $\boldsymbol{w} = |\boldsymbol{w}_u|$, where $\boldsymbol{w}_u$ are actual flow decision variables on undirected links which might be negative depending on the direction of the flows (see, e.g. [21] and references therein). The problem is still convex and it can be reformulated as an linear optimization problem, with which strong linear duality can be applied as normal.

4.1. Istanbul Networks

Similar to Peeta et al. [26], we consider the case studies based on highway networks in Istanbul, Turkey. We start with 9-link network on the Asian side of Istanbul with two OD pairs as shown in Figure 1. All information including retrofitting costs, traversal costs, and survival probabilities are provided in [26, Table 1] except for the survival probability of the added link 31 if it is not retrofit, which we will set at 0.5. The total retrofitting cost is 4260 and we will set the budget B to 50% of the total cost, which is $B = 2130$. The penalty cost M when there is no connection between an OD pair is set to be 30.

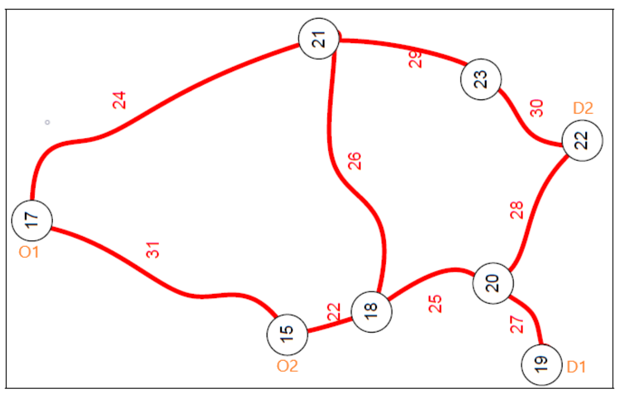

Figure 1: Highway network on the Asian side of Istanbul ([26])

We implement the robust formulation using IBM CPLEX 12.7.1 in C with Microsoft Visual Studio 16.1.5 on a Windows 10 computer with 3.70 GHz CPU and 32.0 GB RAM. As discussed, instead of Δf_i, we use $\Delta \bar{f}$ as defined in (10), which can be computed efficiently in this retrofitting planning application. The number of scenarios in this instance is $2^9 = 512$, which is small enough for us to solve the mixed-integer linear optimization formulation (9) directly. The optimal solution $\boldsymbol{x}^*$ of the robust problem is to retrofit 5 links, 22, 25, 27, 29, and 30 with the worst-case expected traversal cost $C(\boldsymbol{x}^*) = \mathbb{E}_{P_{wc}(\boldsymbol{x}^*)}[f(\boldsymbol{x}^*, \boldsymbol{\xi})] = 20.8505$, where $P_{wc}(\boldsymbol{x}^*)$ is the worst-case distribution given the optimal solution $\boldsymbol{x}^*$. The total computational time for this instance is 6.00 seconds with less than 0.01 seconds used to compute $\Delta \bar{f}$.

If the distribution is independent as assumed in [26], we solve the problem by enumerating all feasible solutions in 0.01 seconds given the size of the instance. The optimal solution $\boldsymbol{x}_{ind}$ is to retrofit 5 links, 22, 25, 26, 28, 29, and 30, which is different from $\boldsymbol{x}^*$. Table 1 shows the expected traversal costs for different distributions given these two solutions. It is clear that $\boldsymbol{x}_{ind}$ is better if the distribution is independent while $\boldsymbol{x}^*$ performs better in hedging against the worst-case scenario if the distribution is not know exactly.

Solution	Independent Distribution	Worst-Case Distribution
$\boldsymbol{x}^*$	19.5240	**20.8505**
$\boldsymbol{x}_{ind}$	**18.8201**	21.3435

Table 1: Expected traversal costs for different distributions

In order to demonstrate further the performance of the solution $\boldsymbol{x}^*$ of the robust problem (as compared to that of $\boldsymbol{x}_{ind}$), we conduct the stress test with contaminated probability distributions following the approach proposed by Dupačová [9]. Similar to the procedure discussed in Bertsimas et al. [2], we setup the stress-test experiment as follows.

1. Generate N_p scenarios from the contaminated probability distribution $P_\lambda = (1-\lambda)P_{ind} + \lambda P_{wc}(\boldsymbol{x}^*)$ for some $\lambda \in [0,1]$, where P_{ind} is the independent distribution given the original link survival probabilities (without retrofitting decisions) and $P_{wc}(\boldsymbol{x}^*)$ is the worst-case distribution given the optimal solution $\boldsymbol{x}^*$.

2. Compute the link survival probabilities, i.e., marginal distributions, from the generated scenarios and obtain the robust solution $\boldsymbol{x}^*_\lambda$ as well as $\boldsymbol{x}^\lambda_{ind}$ under the independence assumption together with the corresponding worst-case traversal costs $C(\boldsymbol{x}^*_\lambda)$ and $C(\boldsymbol{x}^\lambda_{ind})$ using the proposed model.

3. Generate N_o additional scenarios from P_λ and compute the average traversal costs $C_{avg}(\boldsymbol{x}^*_\lambda)$ and $C_{avg}(\boldsymbol{x}^\lambda_{ind})$ by considering all scenarios that could happen if $\boldsymbol{x}^*_\lambda$ and $\boldsymbol{x}^\lambda_{ind}$ are implemented, respectively.

With the described procedure, one can assume that data used in the proposed model is generated from the actual probability distribution, which is the contaminated distribution P_λ and the solutions obtained will be tested with out-of-samples data generated from the same distribution. In this experiment, we set $N_p = 100$ and $N_o = 10,000$, which allows us to obtain reliable average traversal costs. We start by setting $\lambda = 0.5$ and repeat the procedure 10 times to obtain the mean traversal costs and their corresponding standard deviations.

Solution	Worst-Case Cost	Average Cost
$\boldsymbol{x}^*_\lambda$	17.3760 (0.4414)	15.0422 (0.2240)
$\boldsymbol{x}^\lambda_{ind}$	17.6344 (0.8428)	15.5854 (0.7955)

Table 2: Means (standard deviations) of worst-case and average traversal costs for $\lambda = 0.5$

The results in Table 2 show that the mean worst-case cost obtained from $\boldsymbol{x}^*_\lambda$ is better than that from $\boldsymbol{x}^\lambda_{ind}$ as expected. Furthermore, the mean average cost from 10,000 scenarios drawn from the same contaminated P_λ for $\boldsymbol{x}^*_\lambda$ is also better than $\boldsymbol{x}^\lambda_{ind}$ in this case with $\lambda = 0.5$. Note that for these 10 runs, the average cost for $\boldsymbol{x}^*_\lambda$ is always strictly better than that of $\boldsymbol{x}^\lambda_{ind}$ when two solutions are different. Another observation is that standard deviations for both costs when $\boldsymbol{x}^*_\lambda$ is implemented are also smaller, which indicates that the performance $\boldsymbol{x}^*_\lambda$ is more stable in terms of traversal cost. We now vary λ from 0 to 1 and analyze the performance of these two solutions given different levels of contamination. The resulting worst-case and average traversal costs are plotted in Figure 2.

The worst-case cost $C(\boldsymbol{x}^*_\lambda)$ is indeed strictly smaller than $C(\boldsymbol{x}^\lambda_{ind})$ as expected in all cases when two solutions are different. Note that for these runs, $\boldsymbol{x}^*_\lambda = \boldsymbol{x}^\lambda_{ind}$ when $\lambda = 1$, i.e., $P_\lambda = P_{wc}(\boldsymbol{x}^*)$, and $\lambda = 0.6, 0.7$, and 0.8, which results in the same performance in both worst-case and average costs for $\boldsymbol{x}^*_\lambda$ and $\boldsymbol{x}^\lambda_{ind}$ as shown in Figure 2. With respect to the average cost, when P_{ind} is less contaminated with $\lambda \leq 0.2$, $\boldsymbol{x}^\lambda_{ind}$ performs better with $C_{avg}(\boldsymbol{x}^\lambda_{ind}) < C_{avg}(\boldsymbol{x}^*_\lambda)$. On the other hands, for $\lambda \geq 0.3$, $\boldsymbol{x}^*_\lambda$ becomes better, i.e., $C_{avg}(\boldsymbol{x}^*_\lambda) < C_{avg}(\boldsymbol{x}^\lambda_{ind})$), when the two solutions are different. The results show that the proposed model is suitable for the retrofitting planning application in which it is important to pay attention to the worst-case scenario given incomplete distributional information.

Next, we study the performance of Algorithm 1, which is also implemented in C with IBM CPLEX solver used to solve instances both master and separation problems. We use $N_{\max} = 100$ as the maximum number of iterations and $\epsilon = 10^{-6}$ as the numerical tolerance, i.e., the algorithm will

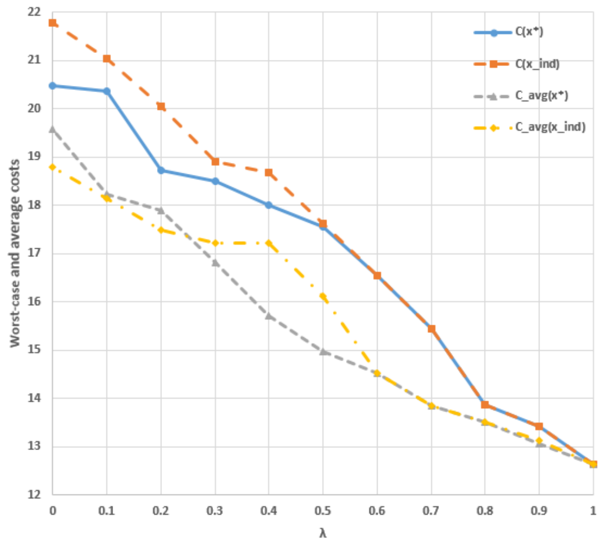

Figure 2: Worst-case and average traversal costs for different levels of contamination

stop if $V_k \geq -\epsilon$ instead of $V_k \geq 0$. We start the algorithm with $\mathcal{S}^{(0)} = \{\mathbf{0}\}$, i.e., the worst-case scenario when all links fail. Running the algorithm for the same instance, we obtain the optimal solution x^* after $N = 37$ iterations. It shows that one does not need to consider all 512 scenarios and the constraint generation method allows us to select a smaller number of necessary scenarios to find the optimal solution. For this instance, the computational time is 3.06 seconds as compared to 6.00 seconds needed to solve the complete formulation with 512 scenarios as mentioned previously.

We now consider the complete 30-link network with 5 OD pairs, which is shown in Figure 3. All information are again provided in [26, Table 1]. We are going to consider three budget levels, $B_1 = 1164$, $B_2 = 2328$, and $B_3 = 3492$, which are 10%, 20%, and 30%, respectively, of the total budget needed to invest in all links. To start with, the penalty cost M when there is no connection between an OD pair is set to 120 as in [26].

The total number of scenarios is 2^{30}, which makes it impossible to solve the mixed-integer linear optimization formulation (9) directly. In order to solve this instance, we use Algorithm 1 and limit the computational time to $T_{\max} = 1$ hour. The computational results with the budget B_1 show that the algorithm does not converge after 165 iterations within one hour limit, i.e., there are still violated constraints that need to be added to the

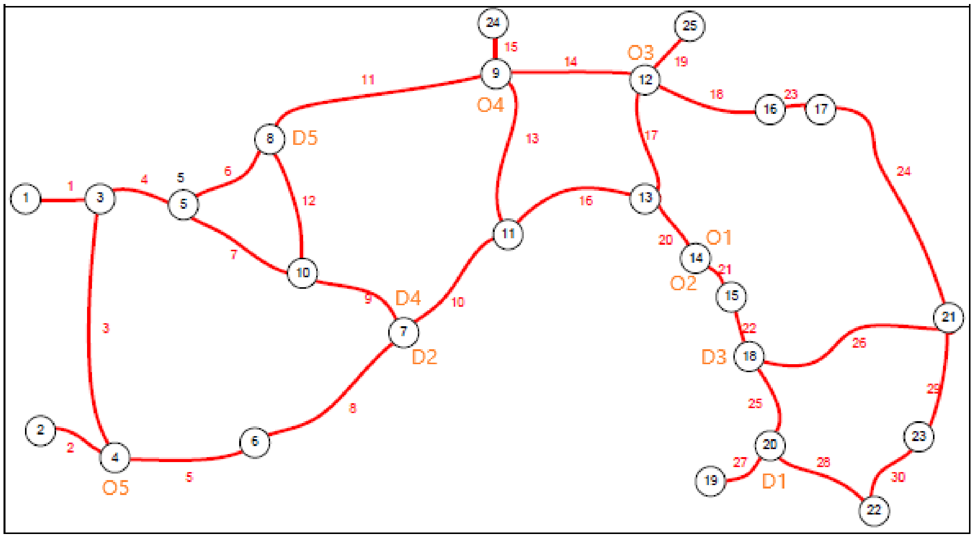

Figure 3: The complete 30-link highway network in Istanbul ([26])

master problem. One of the issues is that in mixed-integer optimization formulation (13) of the separation problem, the traversal costs are computed with a linked-based network flow problem in which all paths are considered and it requires substantial computational time. Facing the same issue, Peeta et al. [26] consider the practical setting when only a subset of (shortest) paths is considered. Let Π_j be the set of considered paths of the OD pair $j \in \mathcal{OD}$. Each path $\pi \in \Pi_j$ is represented by a subset of links and the path cost is $c_\pi(\boldsymbol{s}) = \sum_{i \in \pi} c_i(s_i)$. The cost function $f^j(\boldsymbol{s})$ is now can be written as $f^j(\boldsymbol{s}) = \min_{\pi \in \Pi_j} c_\pi(\boldsymbol{s})$, or equivalently,

$$
\begin{aligned}
f^j(\boldsymbol{s}) = \min_{\boldsymbol{w}} \quad & c_\pi(\boldsymbol{s}) \cdot w_\pi \\
\text{s.t.} \quad & \sum_{\pi \in \Pi_j} w_\pi = 1, \\
& 0 \leq w_\pi \leq 1, \; \forall \, \pi \in \Pi_j,
\end{aligned} \tag{15}
$$

which is in the form of (2). We can again consider (9) and Algorithm 1 with the path-based cost function. Given the path information provided in [26, Table 2] with 4, 6, 4, 4, and 6 paths, respectively, for five given OD pairs, we now test Algorithm 1 again for the 30-link network instance using the path-based cost function. We run Algorithm 1 with the budget B_1 and after 9584 iterations within one hour limit, it still does not converge. Even though it is easier to solve the sub-problem with the path-based formulation, it appears that there are several unnecessary scenarios added to the master problem in Algorithm 1, which makes the algorithm less efficient. We attempt to improve the convergence of the algorithm next by exploiting some structural properties of the underlying problem.

4.2. Constraint Generation Algorithm with Dominant Scenarios

We focus on Problem (7), which is the reformulation of the original problem (4) with $\mathcal{S} = \{0, 1\}$ considered in the application of retrofitting planning. The total number of main constraints is exponential but most of them are not needed. We would like to identify redundant constraints (or unnecessary scenarios) by considering different classes of scenarios based on their resulting shortest paths. Two scenarios $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ belong to a same class $\mathcal{C}$ of scenarios if $\Pi_j(\boldsymbol{s}_1) \cap \Pi_j(\boldsymbol{s}_2) \neq \emptyset$, where $\Pi_j(\boldsymbol{s}) \subseteq \Pi_j$ is the set of shortest paths of the OD pair j when the scenario $\boldsymbol{s}$ is realized. Clearly, if $\boldsymbol{s}_1, \boldsymbol{s}_2 \in \mathcal{C}$ for a class $\mathcal{C}$, $f^j(\boldsymbol{s}_1) = f^j(\boldsymbol{s}_2)$ for all $j \in \mathcal{OD}$. We can now define dominant scenarios as follows.

Definition 1. *A scenario $\boldsymbol{s}_2$* ***dominates*** *another scenario $\boldsymbol{s}_1$ in the same class $\mathcal{C}$ of scenarios if $\mathcal{I}(\boldsymbol{s}_1) \subsetneq \mathcal{I}(\boldsymbol{s}_2)$, where $\mathcal{I}(\boldsymbol{s}) = \{i : s_i = 1\}$. A scenario $\boldsymbol{s}_2$ is* ***dominant*** *in a class $\mathcal{C}$ if $\boldsymbol{s}_2 \in \mathcal{C}$ and there does not exist $\boldsymbol{s}_1 \in \mathcal{C}$ such that $\boldsymbol{s}_1$ dominates $\boldsymbol{s}_2$.*

The following claim shows that we only need to consider dominant scenarios.

Claim 1. *If a scenario $\boldsymbol{s}_1$ is not dominant in a class $\mathcal{S}$, then the constraint corresponding to $\boldsymbol{s}_1$ in* (7) *is redundant.*

Proof. Since scenario $\boldsymbol{s}_1$ is not dominant in $\mathcal{C}$, there exists a scenario $\boldsymbol{s}_2$ in the same class $\mathcal{C}$ such that $\boldsymbol{s}_2$ dominates $\boldsymbol{s}_1$. We shall show that the constraint for $\boldsymbol{s}_2$ in (7) implies the constraint for $\boldsymbol{s}_1$. Indeed, $\mathcal{I}(\boldsymbol{s}_1) \subsetneq \mathcal{I}(\boldsymbol{s}_2)$, thus

$$\begin{aligned} u + \sum_{i \in \mathcal{I}(\boldsymbol{s}_2)} v_i \geq f(\boldsymbol{x}, \boldsymbol{s}_2) &\Rightarrow u + \sum_{i \in \mathcal{I}(\boldsymbol{s}_1)} v_i \geq f(\boldsymbol{x}, \boldsymbol{s}_2) \\ &\Rightarrow u + \sum_{i \in \mathcal{I}(\boldsymbol{s}_1)} v_i \geq f(\boldsymbol{x}, \boldsymbol{s}_1). \end{aligned}$$

The first inequality is due to the non-positivity of v_i and the second inequality is due to $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are in a same class of scenarios, i.e., $f(\boldsymbol{x}, \boldsymbol{s}_1) = f(\boldsymbol{x}, \boldsymbol{s}_2)$. $\square$

Given Claim 1, we only need to consider constraints that correspond to dominant scenarios given that others are redundant. Even though the problem size is reduced, finding all dominant scenarios is still difficult. Let us now focus on classes of scenarios as discussed above. For each OD pair j, we assume that the paths in Π_j are ordered according to their traversal costs when all links are operational, the shortest first, as $\pi_j^1, \ldots, \pi_j^{|\Pi_j|}$. We define $\mathcal{C}_{\iota_1,\ldots,\iota_J}$, $J = |\mathcal{OD}|$, be the class of all scenarios $\boldsymbol{s}$ with which the ι_j-th path in Π_j is set as the shortest path for the OD pair j when $\boldsymbol{s}$ is realized. Given a scenario $\boldsymbol{s}$, it is straightforward to determine which class it belongs to by determining the shortest path(s) for each OD pair. The total number of these scenario classes is $N_c = \prod_{j \in \mathcal{OD}} |\Pi_j|$; however, in general, there will be many empty scenario classes that do not need to be considered as shown later in numerical examples. Note that a scenario $\boldsymbol{s}$ can belong to multiple classes if there are paths with the same traversal cost for some OD pairs. Given the definition of dominant scenarios in Definition 1, it is clear that $\boldsymbol{s}$ is a dominant scenario if $\mathcal{I}(\boldsymbol{s})$ is maximal with respect to inclusion in some scenario class $\mathcal{C}$, i.e, there is no scenario $\boldsymbol{s}' \in \mathcal{C}$ such that $\mathcal{I}(\boldsymbol{s}) \subsetneq \mathcal{I}(\boldsymbol{s}')$. Given a scenario class $\mathcal{C}_{\iota_1,\ldots,\iota_J}$, finding all dominant scenarios is still difficult but one can find a dominant scenario $\boldsymbol{s}$ by maximizing $|\mathcal{I}(\boldsymbol{s})|$. The following binary optimization problem can be used to find such a dominant scenario:

$$
\begin{aligned}
\boldsymbol{s}_{\iota_1,\ldots,\iota_J} \in \arg\max \quad & \sum_{i=1}^{n} s_i \\
\text{s.t.} \quad & s_i = 1, && \forall\, i \in \pi_j^{\iota_j},\ j \in \mathcal{OD}, \\
& \sum_{i \in \pi_j^{\iota}} s_i \leq |\pi_j^{\iota}| - 1, && \forall\, \iota = 1, \ldots, \iota_j - 1,\ j \in \mathcal{OD}, \\
& s_i \in \{0, 1\}, && \forall\, i = 1, \ldots, n.
\end{aligned} \tag{16}
$$

The first constraint is used to make sure that all links of the path $\pi_j^{i_j}$ are operational for all OD pairs $j \in \mathcal{OD}$. The second constraint implies that no path with smaller traversal cost is completely operational if the scenario $\boldsymbol{s}$ is realized. We are going to use these dominant scenarios in the constraint generation algorithm, i.e., $\mathcal{S}_0 = \{\boldsymbol{s}_{\iota_1,\ldots,\iota_J} \,:\, \iota_j = 1, \ldots, |\Pi_j|, j \in$

$\mathcal{OD}\}$. In addition, in each iteration of the algorithm, instead of finding only one violated scenario using (12), we now plan to find one violated scenario for each scenario class $\mathcal{C}_{\iota_1,\ldots,\iota_J}$ using the following binary optimization problem

$$\begin{aligned} \mathrm{SP}_k^{\iota_1,\ldots,\iota_J} : V_k^{\iota_1,\ldots,\iota_J} = \min \quad & u^{(k)} + \sum_{i=1}^{n} v_i^{(k)} \cdot s_i - \sum_{j \in \mathcal{OD}} \omega_j \cdot c_{\pi_j^{\iota_j}}(\boldsymbol{e}) \\ \text{s.t.} \quad & s_i = 1, \quad \forall\, i \in \pi_j^{\iota_j},\ j \in \mathcal{OD}, \\ & \sum_{i \in \pi_j^{\iota}} s_i \leq |\pi_j^{\iota}| - 1, \quad \forall\, \iota = 1, \ldots, \iota_j - 1,\ j \in \mathcal{OD}, \\ & s_i \in \{0, 1\}, \quad \forall\, i = 1, \ldots, n, \end{aligned} \tag{17}$$

where $\boldsymbol{e}$ is the vector of all one's. Similar to (16), the constraints in (17) implies that $\boldsymbol{s} \in \mathcal{C}_{\iota_1,\ldots,\iota_J}$ and the traversal cost is fixed, i.e., $f(\boldsymbol{s}) = \sum_{j \in \mathcal{OD}} \omega_j \cdot c_{\pi_j^{\iota_j}}(\boldsymbol{e})$. The resulting scenario obtained from (17) is likely to be dominant given that $v_i^{(k)} \leq 0$ for all $i = 1, \ldots, n$. More precisely, if $v_i^{(k)} < 0$ for all $i = 1, \ldots, n$, one can show that the resulting scenario is a dominant scenario within the scenario class $\mathcal{C}_{\iota_1,\ldots,\iota_J}$ given its optimality. We can now modify Algorithm 1 using dominant scenarios as Algorithm 2. The computational complexity of the algorithm depends on the number of feasible, i.e., non-empty, scenario classes, which can be pre-determined.

Algorithm 2 Constraint Generation Algorithm with Dominant Scenarios

1: Initialize with $k \leftarrow 0$ and $\mathcal{S}^{(0)}$.
2: **loop**
3: Solve the master problem (11) with $\mathcal{S}_k$ to obtain an optimal solution $\left(\boldsymbol{x}^{(k)}, u^{(k)}, \boldsymbol{v}^{(k)}\right)$.
4: Solve the separation problem (17) with $\left(\boldsymbol{x}^{(k)}, u^{(k)}, \boldsymbol{v}^{(k)}\right)$ to obtain optimal solution $\boldsymbol{s}_k^{\iota_1,\ldots,\iota_J}$ and the optimal value $V_k^{\iota_1,\ldots,\iota_J}$ for $\iota_j = 1, \ldots, |\Pi_j|$, $j \in \mathcal{OD}$. Set $V_k = \min_{\iota_1,\ldots,\iota_J} V_k^{\iota_1,\ldots,\iota_J}$.
5: **if** $V_k < 0$ **then**
6: Update $k \leftarrow k + 1$ and $\mathcal{S}^{(k)} \leftarrow \mathcal{S}^{(k)} \cup \{\boldsymbol{s}_k^{\iota_1,\ldots,\iota_J} : V_k^{\iota_1,\ldots,\iota_J} < 0\}$.
7: **else**
8: Stop. Return $\boldsymbol{x}^{(k)}$ as the optimal solution obtained.

Remark 2. *The notion of dominant scenarios reinforces the underlying motivation of constraint generation algorithms, i.e., to introduce only necessary constraints in each iteration. In addition to the common idea of finding violated constraints (necessary) using* (12) *to add to the master problem, here we incorporate the idea of explicitly excluding unnecessary/redundant constraints, i.e., constraints that can be satisfied automatically if others are included. This idea indeed can be applied to other applications. Having said that, the definition of dominant scenarios/constraints and how to find them are problem-dependent, which requires different analyses for different applications.*

Now, using Algorithm 2, we can solve the 30-link network with the budget B_1 after 14 iterations with the computational time of 36.08 seconds. We start with 980 dominant scenarios, one for each feasible scenario class (out of 6125 scenarios classes). The final number of scenarios used in the algorithm is 1657 as compared to the total number of scenarios of 2^{30}. Figure 4 shows the optimal objective value of the master problem $(\mathrm{MP})_k$ and the number of scenarios added in each iteration k. The results demonstrate the significant effect of dominant scenarios on the efficiency of the constraint generation algorithm.

The optimal solution $\boldsymbol{x}^*$ obtained from Algorithm 2 with the bugdet B_1 is to retrofit 7 links, 4, 7, 9, 12, 21, 22, and 25 with the worst-case expected traversal cost of 63.3535. Under the assumption of independent distributions, the optimal solution $\boldsymbol{x}_{ind}$ is to retrofit 6 links, 10, 17, 21, 22, 23, and 25 with the expected traversal cost of 42.4826 ([27]). We run Algorithm 2 again with the fixed solution $\boldsymbol{x}_{ind}$ and obtain its worst-case expected traversal cost of 67.6508. It is indeed higher than the worst-case expected traversal cost obtained from $\boldsymbol{x}^*$. This worst-case expected cost is also much higher than the expected traversal cost under the independence assumption.

We can run Algorithm 2 for different budgets and Table 3 show their results with N_i as the total number of iterations, N_s as the total number of scenarios used and T as the total computational time. These results show that the problem is less difficult to solve when the budget is increased.

Next we vary the penalty cost M when there is no connection between an OD pair from 40 to 120 given the budget B_1. The optimal solution is changed to to retrofit 5 links, 10, 17, 21, 22, and 25, when M is no more than 100 while the worst-case expected traversal cost decreases as expected (see Table 4). The total number of scenarios used is increased when M increases, which

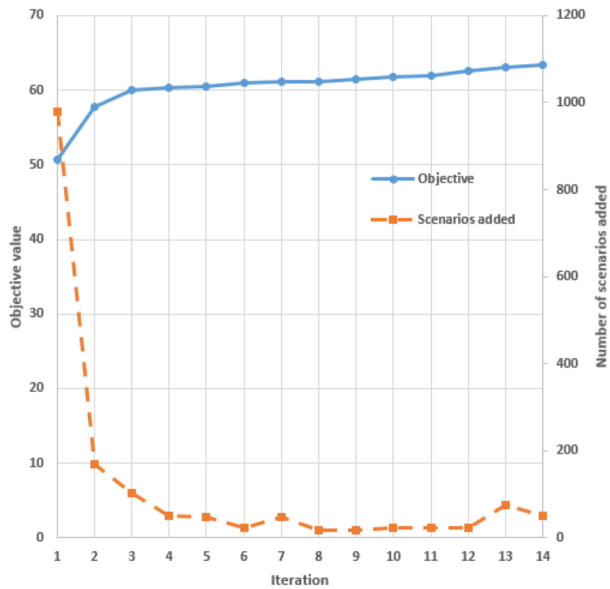

Figure 4: Optimal objective values of master problems and numbers of scenarios added

Budget	$\boldsymbol{x}^*$	$C(\boldsymbol{x}^*)$	N_i	N_s	T (secs)
B_1	4 7 9 12 21 22 25	63.3535	14	1657	36.08
B_2	4 10 12 17 20 21 22 25	37.5329	6	1207	20.64
B_3	3 4 7 10 12 13 17 20 21 22 23 25	20.4009	5	1137	17.82

Table 3: Computational results for different budgets

implies that the problem is likely more difficult when the penalty cost M is high.

M	40	60	80	100	120
$C(\boldsymbol{x}^*)$	22.0506	29.0506	36.0506	43.0506	63.3535
N_s	1341	1376	1380	1470	1657

Table 4: Computational results for different penalty costs

We now consider the effect of number of paths considered in the path-based formulation. Using the k-shortest path algorithms proposed by Yen [33], one can compute all (loop-less) shortest paths for each OD pairs. For this 30-link network instance, the numbers of shortest paths for five given

OD pairs are 14, 30, 10, 12, and 12, respectively. We run Algorithm 2 using 8 different options of numbers of paths for five given OD pairs used in the path-based formulation of the 30-link network instance.

Path option	Number of paths	$C(\boldsymbol{x}^*)$	N_s	T (secs)
1	[1 1 1 1 1]	67.6508	92	0.74
2	[1 3 1 1 3]	66.1442	457	4.89
3	[2 4 2 2 4]	63.3535	589	5.46
4	[3 5 3 3 5]	63.3535	918	11.33
5	[4 6 4 4 6]	63.3535	1657	36.08
6	[8 10 8 8 10]	63.3535	2203	165.43
7	[10 12 10 10 12]	63.3535	2262	353.68
8	[14 30 10 12 12]	63.3535	2490	1234.83

Table 5: Computational results for different penalty costs

Table 5 shows that the worst-case expected traversal cost $C(\boldsymbol{x}^*)$ remains the same for most path options. It implies that in the worst case, scenarios with which the OD pairs are disconnected are likely more important than scenarios with longer paths as their shortest paths in determining the total expected traversal cost. The problem is more difficult when the numbers of paths considered increase, which reflects in the total number of scenarios used. The total computational time is increased exponentially; however, the increase is mainly due to the pre-processing time that one needs to find all feasible classes out of all scenario classes generated by different paths from the given OD pairs. Figure 5 shows that the number of potential scenario classes increases exponentially from 32 to almost 900,000 while the number of feasible classes, which is more important for the implementation of Algorithm 2, only increases to 1500. It also shows that the pre-processing time to find all feasible scenario classes increases exponentially from 0.08 to 1188.46 seconds while the execution time of Algorithm 2 only increases from 0.66 to 46.37 seconds. Note that with the path-based formulation, the computational time used to compute $\Delta\bar{f}$ is actually reduced significantly, which is less than 10^{-4} seconds for these 30-link network instances.

4.3. Random Networks

In order to test the proposed algorithm further, we now generate networks randomly as follows. We start with m nodes and generate an $m \times m$ adjacency

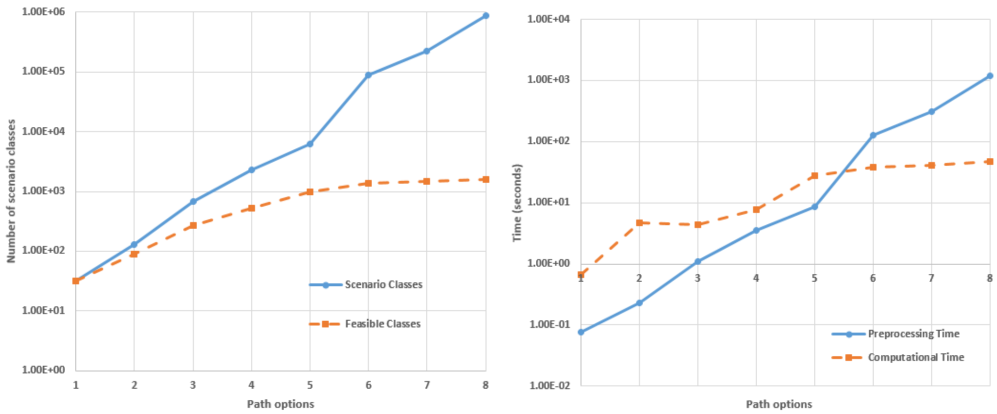

Figure 5: Numbers of scenario classes and computational times for different path options

matrix with approximately $\alpha \times m^2$ (undirected) links, where $\alpha \in (0, 1)$ is the link density. The actual traversal costs on operational links are randomly generated using the uniform distribution on the interval $(0, 1)$. The penalty cost M is set to twice the total traversal cost of all links. The survival probabilities of the links (with and without being retrofitted) are randomly set between 0.5 and 1. The retrofitting costs of the links are also generated randomly using the uniform distribution on the interval $(0, 1)$. The budget is set at 10% of the total retrofitting cost of all links. Similar to 30-link network instance, we generate randomly five OD pairs and for each OD pair, we generate five (loop-less) shortest paths to be used in the path-based information. We shall focus mainly on the computational performance of Algorithm 2 with respect to the network size.

We set $m = 25$ as in the previous 30-link instance. We vary α from 5% to 25% and generate 10 instances for each value of α. Figure 6 shows computational results for these different network densities. The average number of links increases more or less linearly as expected from 35 (for $\alpha = 5\%$)to 119 (for $\alpha = 25\%$). It means that the total number of scenarios can be around 2^{100}. All instances are solved within the one-hour limit with the maximum average computational time of approximately 900 seconds for $\alpha = 15\%$. The largest computation time is approximately 2500 seconds for a network instance with $\alpha = 20\%$. As discussed previously, computational time might depend on the number of feasible scenario classes, which affects the total number of scenarios used to find optimal solutions. It shows in Figure 6 that both the maximum average number of feasible classes (3600) and the maximum average number of scenarios used (6800) happen when $\alpha = 15\%$. The highest number of scenarios used to find optimal solutions is approximately 14000 for an instance with $\alpha = 20\%$, which is still a tiny fraction of the total number of scenarios of $2^{104} \sim 2 \times 10^{31}$ for that particular instance.

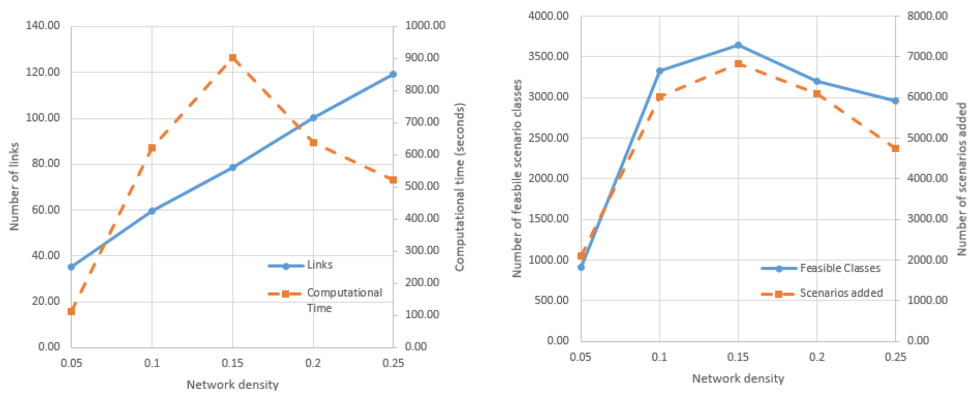

Figure 6: Computational results for different network densities

We also generate network instances with larger m, $m = 50, 75$, and 100 while keeping $\alpha = 5\%$. For $m = 50$, 6 out of 10 instances reach the maximum computational time of one hour. In order to check whether Algorithm 2 can handle larger networks, we remove the one-hour limit and run the algorithm again with one random network instance for each value of m. The computational results for these instances are shown in Table 6. Even though the computational time is high (almost 20 hours for the network instance of 462 links), the number of scenarios needed to find the optimal solution is again only a tiny fraction of the total number of scenarios (32,000 versus $2^{462} \sim 10^{139}$), which shows the efficiency of Algorithm 2.

m	25	50	75	100
n	35	124	280	462
N_s	4253	10369	37048	32539
T (seconds)	401.63	2967.74	27776.60	71124.30

Table 6: Computational results for different network sizes

5. Conclusion

We propose a marginal-based distributionally robust optimization framework to handle probability dependence of decision-dependent discrete distributions which can be applied for the retrofitting planning application. The proposed constraint generation algorithm with the notion of dominant scenarios works well with several case studies. As future research directions, one can investigate multivariate marginal ambiguity models of decision-dependent distributions for relevant applications.

Acknowledgement

We would like to thank all three reviewers for their helpful comments and suggestions, which have helped us significantly improve the paper.

References

[1] Ben-Tal, A., Ghaoui, L.E., Nemirovski, A., 2009. Robust Optimization. Princeton University Press, Princeton, NJ.

[2] Bertsimas, D., Doan, X.V., Natarajan, K., Teo, C.P., 2010. Models for minimax stochastic linear optimization problems with risk aversion. Mathematics of Operations Research 35, 580–602.

[3] Birge, J., Louveaux, F., 2011. Introduction to stochastic programming. Springer, New York, NY.

[4] Chang, L., Peng, F., Ouyang, Y., Elnashai, A., Jr, B.S., 2012. Bridge seismic retrofit program planning to maximize postearthquake transportation network capacity. Journal of Infrastructure Systems 18, 75–88.

[5] Dantzig, G., 1963. Linear programming and extensions. Princeton University Press.

[6] Diwekar, U., 2020. Optimization under uncertainty, in: Introduction to Applied Optimization. Springer, pp. 151–215.

[7] Doan, X.V., Li, X., Natarajan, K., 2015. Robustness to dependency in portfolio optimization using overlapping marginals. Operations Research 63, 1468–1488.

[8] Doan, X.V., Natarajan, K., 2012. On the complexity of nonoverlapping multivariate marginal bounds for probabilistic combinatorial optimization problems. Operations Research 60, 138–149.

[9] Dupačová, J., 2006. Optimization under exogenous and endogenous uncertainty, in: Lukáš, L. (Ed.), Proceedings of the 24th Mathematical Methods in Economics International Conference, University of West Bohemia in Pilsen. pp. 131–136.

[10] Embrechts, P., Puccetti, G., 2006. Bounds for functions of dependent risks. Finance and Stochastics 10, 341–352.

[11] Flach, B., Poggi, M., 2010. On a class of stochastic programs with endogenous uncertainty: theory, algorithm and application. Technical Report no. 05/10. PUC Rio.

[12] Goel, V., Grossmann, I., 2004. A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves. Computers & Chemical Engineering 28, 1409–1429.

[13] Goel, V., Grossmann, I., 2006. A class of stochastic programs with decision dependent uncertainty. Mathematical Programming 108, 355–394.

[14] Gupta, V., Grossmann, I., 2011. Solution strategies for multistage stochastic programming with endogenous uncertainties. Computers & Chemical Engineering 35, 2235–2247.

[15] Gupta, V., Grossmann, I., 2014. A new decomposition algorithm for multistage stochastic programs with endogenous uncertainties. Computers & Chemical Engineering 62, 62–79.

[16] Hellemo, L., 2016. Managing Uncertainty in Design and Operation of Natural Gas Infrastructure. Ph.D. thesis. Norwegian University of Science and Technology.

[17] Jonsbråten, T., Wets, R., Woodruff, D., 1998. A class of stochastic programs with decision dependent random elements. Annals of Operations Research 82, 83–106.

[18] Lappas, N., Gounaris, C., 2018. Robust optimization for decision-making under endogenous uncertainty. Computers & Chemical Engineering 111, 252–266.

[19] Luo, F., Mehrotra, S., 2020. Distributionally robust optimization with decision dependent ambiguity sets. Optimization Letters , 1–30.

[20] Mahmoudzadeh, H., Purdie, T., Chan, T., 2016. Constraint generation methods for robust optimization in radiation therapy. Operations Research for Health Care 8, 85–90.

[21] McBride, R., Mamer, J., 2001. Solving the undirected multicommodity flow problem using a shortest path-based pricing algorithm. Networks: An International Journal 38, 181–188.

[22] Nohadani, O., Sharma, K., 2018. Optimization under decision-dependent uncertainty. SIAM Journal on Optimization 28, 1773–1795.

[23] Noyan, N., Rudolf, G., Lejeune, M., 2018. Distributionally robust optimization with decision-dependent ambiguity set. Optimization Online E-Print 2018/09/6821 .

[24] Pan, F., 2005. Stochastic network interdiction: models and methods. Ph.D. thesis. The University of Texas at Austin.

[25] Pass, B., 2011. Uniqueness and Monge solutions in the multimarginal optimal transportation problem. SIAM Journal on Mathematical Analysis 43, 2758–2775.

[26] Peeta, S., Salman, S., Gunnec, D., Viswanath, K., 2010. Pre-disaster investment decisions for strengthening a highway network. Computers & Operations Research 37, 1708–1719.

[27] Prestwich, S., Laumanns, M., Kawas, B., 2014. Distribution shaping and scenario bundling for stochastic programs with endogenous uncertainty, in: Stochastic Programming E-Print Series 5, Institut für Mathematik, HU Berlin. pp. 1–24.

[28] R. Gao, R., Kleywegt, A., 2017. Data-driven robust optimization with known marginal distributions. Technical Report. Georgia Institute of Technology.

[29] Royset, J., Wets, R., 2017. Variational theory for optimization under stochastic ambiguity. SIAM Journal on Optimization 27, 1118–1149.

[30] Rüschendorf, L., 1991. Fréchet-bounds and their application, in: Aglio, G.D., Kotz, S.I., Salinetti, G. (Eds.), Advances in Probability Distributions with Given Marginals. Mathematics and Its Applications Volume 67, pp. 151–187.

[31] Simchi-Levi, D., Wang, H., Wei, Y., 2018. Constraint generation for two-stage robust network flow problems. INFORMS Journal on Optimization .

[32] Vayanos, P., Kuhn, D., Rustem, B., 2011. Decision rules for information discovery in multi-stage stochastic programming, in: Decision and Control and European Control Conference (CDC-ECC), IEEE. pp. 7368–7373.

[33] Yen, J., 1971. Finding the k shortest loopless paths in a network. Management Science 17, 712–716.

[34] Žáčková, J., 1966. On minimax solutions of stochastic linear programming problems. Časopis pro pěstování matematiky 91, 423–430.

[35] Zhang, J., Xu, H., Zhang, L., 2016. Quantitative stability analysis for distributionally robust optimization with moment constraints. SIAM Journal on Optimization 26, 1855–1882.