Distributionally risk-receptive and risk-averse network interdiction problems with general ambiguity set*

Sumin Kang[1] and Manish Bansal[†1]

[1] *Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 24061, USA*

May, 2022

Abstract

In this paper, we introduce generalizations of stochastic network interdiction problem with distributional ambiguity. In particular, we consider a distributionally risk-averse (or robust) network interdiction problem (DRA-NIP) and a distributionally risk-receptive network interdiction problem (DRR-NIP) where a leader maximizes a follower's minimal expected objective value for either the worst-case or the best-case, respectively, probability distribution belonging to a given set of distributions (referred to as ambiguity set). The DRA-NIP arises in applications where a risk-averse leader is the main protagonist who interdicts a follower (opponent or evader) to cause delays in their supply convoy. In contrast, the DRR-NIP is applicable for network vulnerability analysis where a network user (or follower) also seeks to identify vulnerabilities in the network against potential disruptions by an adversary (or leader) who is receptive to risk for improving the expected objective values. We present exact and approximation algorithms for solving DRA-NIP and DRR-NIP with a general ambiguity set. We also provide conditions and family of ambiguity sets for which these approaches are finitely convergent. To evaluate the effectiveness and efficiency of the approaches for solving DRA-NIP and DRR-NIP, we provide results of our extensive computational experiments performed on instances known in the literature for (risk-neutral) stochastic NIP.

Keywords: stochastic network interdiction, distributionally robust optimization, distributionally risk-receptive, decomposition algorithm, general ambiguity set

1 Introduction

Network interdiction problem (NIP) is characterized as a game played on a network between two players: an interdictor, referred to as a leader, and a network user, referred to as a

*Distribution A: Approved for public release; distribution unlimited. OPSEC # 6023

[†]Corresponding author (bansal@vt.edu)

follower. The leader is a player who makes interdiction decisions for some network components to degrade the follower's performance on it. In other words, the leader attacks some arcs of the network to either increase cost to travel through them or remove them from the network. The follower observes the interdiction decision and finds a minimum cost (shortest) path between a given pair of source and destination nodes of the interdicted network. Since a given budget limits the leader's ability to interdict the network components, the leader has to make interdiction decisions with the objective of maximizing the minimum travel cost for the follower. The NIP has applications in military logistics where an interdictor attacks an opponent's logistics network to delay a supply convoy [13, 15] and in interdicting the smuggling of nuclear materials [25, 28, 39]. It also provides vulnerability analysis for critical infrastructure under adversarial attacks [6, 8, 35] by identifying vulnerable components in the network whose disruption will impose substantial damage.

In this paper, we focus on network interdiction under uncertainty where the success of interdiction attempts and the impact of interdiction on travel cost associated with each interdicted arc is uncertain. Previous studies have proposed (risk-neutral) stochastic programming models to formulate stochastic NIPs where the uncertain parameters are represented by random variables with a known probability distribution. However, a distinct aspect of uncertainty faced in many real-world applications, in particular, military operations, is the limited availability of relevant historical data, which leads to incomplete information of probability distribution associated with the uncertain parameters. One approach to address the ambiguity of the distribution is distributionally robust optimization (DRO) [12, 36]. In DRO, it is assumed that the probability distribution belongs to a set of distributions, known as an ambiguity set. Solving a DRO model gives a distributionally robust (risk-averse) solution by optimizing the expected value of the objective function for the worst-case distribution within the ambiguity set.

We propose distributionally risk-averse and risk-receptive models for the stochastic NIP (S-NIP) with distributional ambiguity. Applying the traditional DRO modeling approach to S-NIP renders the leader's objective as to maximize the expectation of the minimal objective values of the follower associated with the worst-case probability distribution. Throughout the paper, we refer to this DRO-based model as a distributionally risk-averse NIP (DRA-NIP). The DRA-NIP arises in applications where the leader is the main protagonist; for example, the leader interdicting an evader or delaying the supply convoy of an opponent. In contrast, for infrastructure vulnerability analysis, a network user also seeks to identify vulnerabilities of infrastructure against potential interdictions (or disruptions) by an adversary who is receptive to risk for improving the expected objective values. From the viewpoint of the leader, we refer to this problem as a distributionally risk-receptive NIP, denoted by DRR-NIP, where the leader makes decisions with respect to the best-case probability distribution from a given set of distributions.

The DRA-NIP and the DRR-NIP with a singleton ambiguity set reduce to the S-NIP. Therefore, the optimal objective values from the DRA-NIP and the DRR-NIP provide lower and upper bounds, respectively, of the objective value obtained from solving the S-NIP. In addition to generalizing the S-NIP, the DRA-NIP and DRR-NIP allow adjustments in the models based on the level of risk-aversion and risk-receptiveness, respectively, of a decision maker. These adjustments are managed by considering ambiguity sets ranging from a singleton set to a set of all probability distributions supported on Ω. Using the latter ambiguity

set (denoted by $\hat{\mathcal{P}}$), the DRA-NIP provides the most conservative solution for the leader, while we can obtain solutions reflecting the decreased level of risk-aversion by solving the DRA-NIP with the subsets of this set. Similarly, the DRR-NIP with $\hat{\mathcal{P}}$ as ambiguity set provides the most optimistic solution for the leader (or a most conservative solution for the follower) by considering the case when the uncertainty is realized in the most unfavorable way for the follower. We can also decrease the level of risk-receptiveness for the leader by using subsets of $\hat{\mathcal{P}}$ in the DRR-NIP.

1.1 Problem Definition: DRA-NIP and DRR-NIP

We consider a minimum cost path (or shortest path) network interdiction problem in which the success of interdiction and the impact of interdiction on each arc are assumed to be uncertain. This problem is mathematically defined as follows. Given a directed network $G = (N, A)$, the leader interdicts a set of arcs to maximize the expected shortest path traveling cost of the follower. A source node and a destination node along the follower's path are denoted by s and t, respectively. For each arc $a = (i, j) \in A$ between nodes $i, j \in N$, let traveling cost be denoted by c_a. Also, the penalty cost vector and the success of interdiction vector are denoted by random vectors $\tilde{d} \in \mathbb{R}_+^{|A|}$ and $\tilde{\xi} \in \{0, 1\}^{|A|}$, respectively. We assume that $(\tilde{d}, \tilde{\xi})$ have finite support Ω, and each possible realization $(d^\omega, \xi^\omega), \omega \in \Omega$ occurs with a probability p_ω, $\omega \in \Omega$. The leader's interdiction decision is denoted by a binary decision vector $\mathbf{x} \in \{0, 1\}^{|A|}$ where $x_a = 1$ or $x_a = 0$ implies that arc $a \in A$ is interdicted or not interdicted, respectively. A set of all feasible interdiction decision vectors is represented as $X := \{\mathbf{x} \in \{0, 1\}^{|A|} | \sum_{a \in A} x_a = b\}$ where $b > 0$ denotes the interdicting budget. Note that the follower is a "wait-and-see" player who makes a decision after observing the realizations of uncertainty and the leader's decision. Thus, the follower's shortest path problem in each scenario $\omega \in \Omega$ involves the deterministic traveling costs for all arcs which are represented as $(c_a + d_a^\omega \xi_a^\omega \hat{x}_a)$ for $a \in A$ where $\hat{x}_a$ is a given interdiction decision. Note that the penalty term $(d_a^\omega \xi_a^\omega \hat{x}_a)$ is activated when both $\hat{x}_a$ and ξ_a^ω have values of 1, which means the leader decides to interdict arc $a \in A$, and the interdiction is successful in scenario $\omega \in \Omega$. The follower's decision variable $y_a^\omega \in \{0, 1\}$ represents whether arc $a \in A$ is traversed in the shortest path in scenario $\omega \in \Omega$ or not. For each node $i \in N$, let $\delta^+(i)$ and $\delta^-(i)$ be the set of outgoing and incoming arcs for node i, respectively. We then formulate a risk-neutral stochastic (shortest or minimum cost path) network interdiction problem as follows:

$$\text{(S-NIP)} \quad \max_{\mathbf{x} \in X} \sum_{\omega \in \Omega} p_\omega \left(\min_{\mathbf{y}^\omega} \sum_{a \in A} (c_a + d_a^\omega \xi_a^\omega x_a) y_a^\omega \right), \tag{1a}$$

$$\text{s.t.} \sum_{a \in \delta^+(i)} y_a^\omega - \sum_{a \in \delta^-(i)} y_a^\omega = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases}, \quad \text{for all } i \in N, \quad \text{for all } \omega \in \Omega, \tag{1b}$$

$$y_a^\omega \in [0, 1], \quad \text{for all } a \in A, \quad \text{for all } \omega \in \Omega. \tag{1c}$$

Notice that since the constraint matrix associated with Constraints (1b), i.e., flow balance constraints, of the inner shortest path problem is totally unimodular, we can let $y_a \in [0, 1]$

instead of being binary [5]. Also, we denote the set of feasible solutions of the follower by $Y^\omega := \{\mathbf{y}^\omega |$ (1b) and (1c) hold$\}$ for $\omega \in \Omega$.

We now formulate the distributionally risk-averse and risk-receptive (shortest path) interdiction problems, i.e., DRA-NIP and DRR-NIP, associated with the different attitudes of the leader towards distributional ambiguity, as follows.

$$\text{(DRA-NIP)} \quad \max_{\mathbf{x}\in X} \min_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega \min_{\mathbf{y}^\omega\in Y^\omega} \sum_{a\in A} (c_a + d_a^\omega \xi_a^\omega x_a) y_a^\omega; \tag{2}$$

$$\text{(DRR-NIP)} \quad \max_{\mathbf{x}\in X} \max_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega \min_{\mathbf{y}^\omega\in Y^\omega} \sum_{a\in A} (c_a + d_a^\omega \xi_a^\omega x_a) y_a^\omega, \tag{3}$$

where $P = \{p_\omega\}_{\omega\in\Omega}$. In the DRA-NIP (2), the leader is risk-averse with the goal to maximize the expected shortest path traveling cost of the follower for the worst-case probability distribution among the ambiguity set $\mathcal{P}$, which is obtained by minimizing over $\mathcal{P}$. On the contrary, in the DRR-NIP (3), we consider a risk-receptive leader whose focus is on the best-case probability distribution in $\mathcal{P}$, i.e., maximizing over the ambiguity set, and the corresponding maximal expected shortest path traveling cost. Notice that for $|\mathcal{P}| = 1$, both of these models reduce to the (risk-neutral) S-NIP, and for $|\Omega| = 1$, they further reduce to the deterministic NIP. Likewise, when $\mathcal{P}$ is defined by all probability distributions supported on Ω, DRA-NIP and DRR-NIP reduce to robust optimization models where leader makes risk-averse decisions for the worst-case scenario and risk-receptive decisions for the best-case scenario, respectively. In addition to generalizing robust and stochastic optimization models, these models provide adjustable levels of risk-aversion or risk-receptiveness for an interdictor or a network user. This adjustability is tailor-made for military applications that have limited availability of relevant historical data. Lastly, we note the readers that the proposed approaches in this paper are applicable to the case where the traveling costs are also uncertain, i.e., $c_a^\omega, \omega \in \Omega$, even though we assume that the traveling costs of arcs $c_a, a \in A$ are deterministic.

1.2 Illustrative Example

Consider a network $G = (N, A)$ in Figure 1 where $N = \{1, 2, 3, 4\}$, $A = \{(1, 2), (1, 3), (2, 4), (3, 4)\}$, and nodes 1 and 4 are source s and destination t nodes, respectively. The traveling costs $c_{12} = 4$, $c_{13} = 2$, $c_{24} = 4$, and $c_{34} = 2$ are known. In this example, we assume the uncertain success of interdiction attempts and consider two realizations: $\xi^{\omega_1} = (1, 1, 0, 0)$ and $\xi^{\omega_2} = (0, 0, 1, 1)$ corresponding to the set A. For the sake of convenience, in this example, we assume penalty costs $d_{12}^\omega = 3$, $d_{13}^\omega = 8$, $d_{24}^\omega = 4$, and $d_{34}^\omega = 6$ for all $\omega \in \{\omega_1, \omega_2\}$, and consider a finite ambiguity set $\mathcal{P} = \{P_1, P_2, P_3\} = \{(0.7, 0.3), (0.3, 0.7), (0.5, 0.5)\}$. Notice that the shortest (or minimum cost) path between s and t on this network when there is no interdiction, is a path traversing along arcs $(1, 3)$ and $(3, 4)$, which we denote by SP^*, that costs 4 units.

Let the interdiction budget b be equal to 2 units. This implies that the leader has to interdict two arcs in A, and therefore, she has six possible choices: $\{(1, 2), (1, 3)\}$, $\{(2, 4), (3, 4)\}$, $\{(1, 3), (3, 4)\}$, $\{(1, 2), (2, 4)\}$, $\{(1, 2), (3, 4)\}$, and $\{(1, 3), (2, 4)\}$, i.e., the set $X = \{\mathbf{x}^i\}_{i=1}^6 = \{(1, 1, 0, 0), (0, 0, 1, 1), (0, 1, 0, 1), (1, 0, 1, 0), (1, 0, 0, 1), (0, 1, 1, 0)\}$. For a given $\mathbf{x} \in X$ and

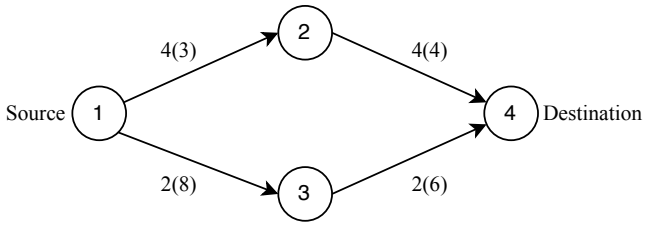


Figure 1: Example network with source node $s = 1$ and destination node $t = 4$.

Table 1: Expected travel cost for a given leader's action $\mathbf{x} \in X = \{\mathbf{x}^1, \ldots, \mathbf{x}^6\}$ and probability distribution $P \in \{P_1, P_2, P_3\}$; Optimal solution and objective value of the S-NIP for each distribution P, DRR-NIP, and DRA-NIP.

Probability $P \in \mathcal{P}$	$\mathbf{x}^1$	$\mathbf{x}^2$	$\mathbf{x}^3$	$\mathbf{x}^4$	$\mathbf{x}^5$	$\mathbf{x}^6$	**Optimal Solution**	**Optimal Value**
P_1	8.9	5.8	8.0	4.0	5.2	6.8	$\mathbf{x}^1$	8.9
P_2	6.1	8.2	8.0	4.0	6.8	5.2	$\mathbf{x}^2$	8.2
P_3	7.5	7.0	8.0	4.0	6.0	6.0	$\mathbf{x}^3$	8.0
Best-Case (DRR-NIP)	8.9	8.2	8.0	4.0	6.8	6.8	$\mathbf{x}^1$	8.9
Worst-Case (DRA-NIP)	6.1	5.8	8.0	4.0	5.2	5.2	$\mathbf{x}^3$	8.0

$P \in \mathcal{P}$, we solve the shortest path problem for each scenario and report the expected shortest path cost in rows 2-4 of Table 1. Using these expected costs, we also find an optimal interdiction solution corresponding to the following risk preference of the leader.

- **Risk-Averse (DRA-NIP)**: Consider a risk-averse interdictor who chooses to interdict arcs $(1, 3)$ and $(3, 4)$ in an optimal solution $\mathbf{x}^3$. Note that both of these arcs are on path SP^*. The interdiction will successfully affect cost to travel arc $(1, 3)$ in scenario ω_1 and arc $(3, 4)$ in scenario ω_2. As a result, the total traveling cost of SP^* increases by 8 and 6 units for scenario ω_1 and ω_2, respectively. Accordingly, an alternative path along arcs $(1, 2)$ and $(2, 4)$ that costs 8 units becomes a new shortest path in both scenarios. From the interdictor's perspective, this choice shows the risk-aversion as it guarantees the success of the interdiction on the prior shortest path SP^* and prevents the follower from traversing through this path for both scenarios.

- **Risk-Receptive (DRR-NIP)**: If we assume that the interdictor is risk-receptive, then the optimal strategy is to interdict arcs $(1, 2)$ and $(1, 3)$, i.e., $\mathbf{x}^1$. Both the arcs will be successfully interdicted in scenario ω_1, thereby resulting in a shortest-path cost of 11 units. In contrast, the interdiction attempts will fail in scenario ω_2, and thus SP^* remains the shortest path with a cost of 4 units. Observe that this interdiction solution is worse for the worst-case probability distribution in comparison to the DRA-NIP solution. However, from the follower's perspective, this result shows the most pessimistic scenario when the interdictor is most successful, which is not observable in the risk-averse model.

- **Risk-Neutral (S-NIP)**: Observe that the optimal interdiction solution for the S-NIP changes as we vary the known probability distribution, i.e., $\mathbf{x}^1$ (8.9 units) is optimal

for P_1, $\mathbf{x}^2$ (8.2 units) is optimal for P_2, and $\mathbf{x}^3$ (8 units) is optimal for P_3 distribution. The optimal solution values range from the DRA-NIP's optimal solution value to the DRR-NIP's optimal solution value.

1.3 Literature Review

In the 1970s, Fulkerson and Harding [13] studied the linear programming relaxation of the deterministic NIP, i.e., (1) with $|\Omega| = 1$, where interdiction variables $\mathbf{x}$ are continuous. In particular, they assumed that an interdiction action on an arc can increase the traveling cost of the arc by any fractional amount within a given budget. Golden [15] studied a variant of the foregoing problem where the objective is to minimize the total interdicting cost required to ensure a certain amount of increase in the shortest path traveling cost. At the beginning of the 21st century, the deterministic NIP with binary interdiction variable was studied by Israeli and Wood [21]. They proposed a Benders' decomposition approach and so-called super-valid inequalities that reduce the size of the feasible region X by cutting off non-optimal feasible solutions. In Israeli's Ph.D. dissertation [20], the author also utilized the decomposition approach for solving the S-NIP. Hemmecke et al. [17] studied the S-NIP and its variant where the objective is to maximize the probability that the shortest path traveling cost exceeds a given threshold, i.e., maximize the expectation of the indicator functions where an indicator function for a scenario takes a value of 1 when the threshold constraint for this scenario is met and 0, otherwise. To solve these problems, they dualized the inner minimization (or follower's) problem to derive a monolithic mixed-integer maximization program. In a follow-up paper, Held et al. [16] proposed a decomposition approach for the foregoing variant of the S-NIP. Nguyen and Smith [27] studied the S-NIP where the traveling costs (c_a and d_a) are assumed to be uniformly distributed on given intervals, thereby leading to a continuous sample space defined by a hyperrectangle. The authors presented a sequential approximation algorithm that iteratively reduces the gap between the bounds on the expected shortest path traveling cost by increasing the number of partitions of the given intervals in each iteration. Subsequently, Nguyen and Smith [26] studied a risk-averse variant of the aforementioned problem using Conditional-Value-at-Risk (CVaR) risk measure that an interdictor seeks to maximize. Song and Shen [37] proposed a risk-averse S-NIP where the risk aversion of a leader is modeled by a chance constraint to ensure that the follower's shortest path cost exceeds a given threshold with high probability. The objective of their model is to minimize the total interdiction cost subject to a given interdiction budget. Pay et al. [29] also considered a risk-averse S-NIP where a utility function represents a leader's risk preference which belongs to a set of functions defined using historical data. Its objective is to maximize the expected worst-case utility of the leader defined for a known probability distribution. Meanwhile, recent studies have also considered different aspects of NIP in the deterministic setting, such as information asymmetry between players [3, 4, 33], incomplete information [42], and randomized interdiction strategy [19]. Note that in the aforementioned studies on NIPs and S-NIPs, the distributional ambiguity in S-NIP has not been considered.

For the completeness of the literature review, we also briefly discuss the maximum-reliability path interdiction problem [25, 28] and the maximum flow interdiction problem [40]. We refer the readers to a survey paper by Smith and Song [37] for various types of network interdiction problems, such as knapsack interdiction problem [11] and maximum

clique interdiction problem [14]. The maximum-reliability path interdiction problem has been studied in the literature as an application of NIP in detecting evaders. In this problem, a leader installs sensors on a set of arcs to maximize the probability of detection. A follower (evader) seeks to find a maximum-reliability path from s to t, minimizing the probability of being detected. Pan and Morton [28] proposed a Benders' decomposition approach for the stochastic maximum-reliability path interdiction problem. In the maximum flow interdiction problem, a leader seeks to minimize the maximum network flow through interdiction actions that are characterized as removing [40] or diminishing capacities of a set of arc [41]. The stochastic variants of this problem are studied by Cormican et al. [9] and Janjarassuk and Linderoth [22], where the former provides a sequential approximation approach, and the latter presents a Benders' decomposition approach. Lei et al. [24] proposed risk-averse stochastic maximum flow interdiction problems in which they use CVaR measure in the objective function to incorporate risk preferences of both the leader and the follower. Additionally, in these problems, after a leader determines an interdiction plan, a follower can reinforce a set of arcs before the realization of the uncertainty. Similarly, Hien et al. [18] studied a network fortification problem where a network user has capabilities to fortify network components before an interdictor attacks the network. This problem can then be viewed as a defender-attacker-defender model, discussed in [8], where the network user aims to protect the network against a potential interdiction. They presented a stochastic maximal flow interdiction and fortification problem and proposed a robust stochastic approximation method to solve it. Recently, Sadana and Delage [32] studied a distributionally risk-averse (robust) maximum flow interdiction problem where the leader's decision is to determine a probability distribution defined over a finite set of all feasible interdiction plans with the objective of minimizing the worst-case CVaR of the flow. They utilized the distributionally robust CVaR measure to handle uncertain capacities where the ambiguity set is a budgeted uncertainty set (introduced in [7]). They first derived a nonlinear reformulation of the problem and then solved the reformulation using a spatial branch-and-bound algorithm.

1.4 Contributions and Organization of this Paper

In this paper, we introduce two modeling frameworks, i.e., DRR-NIP and DRA-NIP, that allow adjustments in the risk-receptiveness and risk-aversion, respectively, of a decision maker. These models generalize the deterministic NIP [21] and S-NIP [17] studied in the literature. The DRR-NIP also provides vulnerability analysis of the network from the perspective of the follower (or network user). In Sections 2 and 3, we present exact solution approaches for solving the DRR-NIP and the DRA-NIP with a general ambiguity set. We also provide conditions and a family of ambiguity sets for which the foregoing algorithms converge in a finite number of iterations. In Section 2, we also discuss an approach to efficiently produce a lower bound for the DRR-NIP. In Section 4, we present the results of our computational experiments performed to evaluate the effectiveness and efficiency of the proposed modeling frameworks and the algorithms. Specifically, we first show the computational efficacy of our approaches by analyzing results conducted on instances with a varying number of arcs having uncertainty. We then demonstrate the significance of solutions from the DRA-NIP and the DRR-NIP for moment matching and Wasserstein-distance ambiguity sets. Lastly, we provide concluding remarks in Section 5.

2 Solution Approaches for the DRR-NIP

In this section, we present two exact solution algorithms and an approximation algorithm for solving the DRR-NIP with a general ambiguity set. We redefine the DRR-NIP as follows:

$$z_{RR}^{opt} := \max_{\mathbf{x}\in X} \quad D_{max}(\mathbf{x}) := \max_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega Q_\omega(\mathbf{x}), \tag{4a}$$

$$\text{where } Q_\omega(\mathbf{x}) := \min_{\mathbf{y}^\omega\in Y^\omega} \sum_{a\in A} (c_a + d_a^\omega \xi_a^\omega x_a) y_a^\omega. \tag{4b}$$

Note that functions $Q_\omega(\mathbf{x}), \omega \in \Omega$ and $\sum_{\omega\in\Omega} p_\omega Q_\omega(\mathbf{x})$ for a given $P = \{p_\omega\}_{\omega\in\Omega} \in \mathcal{P}$ are concave on the convex hull of X. However, function $D_{max}(x)$ is nonconcave since it is a pointwise maximum of concave functions. To handle this nonconcavity of the objective function of DRR-NIP, we present: (A) Reformulation-based exact algorithm in which we reformulate the DRR-NIP and then solve the reformulation using an L-shaped method with a branch-and-cut framework; (B) Cuts-based decomposition exact approach in which we derive valid inequalities to approximate $D_{max}(\mathbf{x})$ and use them within a decomposition framework; and (C) Restrict-and-append algorithm to derive lower bounds for z_{RR}^{opt}.

2.1 Reformulation-based Approach for DRR-NIP

Combining the outer and inner maximization, formulation (4) is equivalent to

$$\max \quad \sum_{\omega\in\Omega} p_\omega Q_\omega(\mathbf{x}), \tag{5a}$$

$$\text{s.t.} \quad \mathbf{x} \in X, \{p_\omega\}_{\omega\in\Omega} \in \mathcal{P}, \tag{5b}$$

where we treat $\{p_\omega\}_{\omega\in\Omega}$ as first-stage variables. This leads to bilinear terms $p_\omega x_a$ in (5a) because

$$p_\omega Q_\omega(\mathbf{x}) = \min_{\mathbf{y}^\omega\in Y^\omega} \sum_{a\in A} (p_\omega c_a + d_a^\omega \xi_a^\omega p_\omega x_a) y_a^\omega, \quad \text{for all } \omega \in \Omega.$$

We linearize these bilinear terms as follows. Since the interdiction variables x_a, $a \in A$, are binary and $p_\omega \in [0,1], \omega \in \Omega$, we introduce a nonnegative continuous variable $\eta_a^\omega \in \mathbb{R}_+$ that satisfies $\eta_a^\omega \le p_\omega$ and $\eta_a^\omega \le x_a$ for $\omega \in \Omega, a \in A$, so that $p_\omega x_a$ can be replaced by η_a^ω. This results in the following reformulation of the DRR-NIP:

$$\max \quad \sum_{\omega\in\Omega} R_\omega(p_\omega, \eta^\omega), \tag{6a}$$

$$\text{s.t.} \quad \eta_a^\omega \le x_a, \quad \text{for all } \omega \in \Omega, a \in A, \tag{6b}$$

$$\eta_a^\omega \le p_\omega, \quad \text{for all } \omega \in \Omega, a \in A, \tag{6c}$$

$$\eta_a^\omega \ge 0, \quad \text{for all } \omega \in \Omega, a \in A, \tag{6d}$$

$$\mathbf{x} \in X, \{p_\omega\}_{\omega\in\Omega} \in \mathcal{P}, \tag{6e}$$

where $\eta^\omega := \{\eta_a^\omega\}_{a\in A}$ and $R_\omega(p_\omega, \eta^\omega) := \min\left\{ \sum_{a\in A} (p_\omega c_a + d_a^\omega \xi_a^\omega \eta_a^\omega) y_a^\omega : \mathbf{y}^\omega \in Y^\omega \right\}$ for $\omega \in \Omega$. Note that, for a given $\mathbf{y}^\omega$, $R_\omega(p_\omega, \eta^\omega)$ is an affine function of p_ω and η^ω. This indicates that

$R_\omega(p_\omega, \eta^\omega)$ is a point-wise minimum of affine functions and is thus a piecewise linear concave function. Formulation (6) can be solved using an L-shaped method with branch-and-cut framework [23, 38] for two-stage stochastic programs with continuous recourse, i.e., $R_\omega(\cdot)$ in our case, and mixed-binary first-stage variables $(x, \{p_\omega, \eta^\omega\}_\omega)$. In this approach, so-called optimality cuts for $R_\omega(p_\omega, \eta^\omega)$, i.e.,

$$R_\omega(p_\omega, \eta^\omega) \leq \sum_{a \in A} (p_\omega c_a + d_a^\omega \xi_a^\omega \eta_a^\omega) \hat{y}_a^\omega,$$

are derived using the extreme points $\hat{\mathbf{y}}^\omega$ of Y^ω to get outer approximations of R_ω for $\omega \in \Omega$. Note that this method is applicable for a general ambiguity set $\mathcal{P}$. For an ambiguity set $\mathcal{P}$ defined as a polyhedral set (for example, moment matching set [2, 10], Wasserstein ambiguity set [30], or total variation distance set [31]) or a mixed-binary set, the feasible region of (6) is a mixed-binary set. Reader can refer to Appendix A for more details on the implementation of this approach (used for our computational experiments).

2.2 Cuts-Based Decomposition Approach for DRR-NIP

In this section, we present another decomposition approach for solving the DRR-NIP where instead of solving formulation (6) with $(\mathbf{x}, \{p_\omega, \eta^\omega\}_\omega)$ as variables, we utilize a distribution separation procedure and the following valid inequalities to obtain outer approximations of the nonconcave objective function $D_{max}(\mathbf{x})$ within a Benders' decomposition framework.

Theorem 1. *For a given interdiction solution $\hat{\mathbf{x}} = \{\hat{x}_a\}_{\omega \in \Omega}$ and the follower's optimal solutions $\hat{\mathbf{y}}^\omega = \{\hat{y}_a^\omega\}_{a \in A}$ for $Q_\omega(\hat{\mathbf{x}})$, for $\omega \in \Omega$, the following inequality is valid for all $\mathbf{x} \in X$.*

$$D_{max}(\mathbf{x}) \leq D_{max}(\hat{\mathbf{x}}) + \sum_{a \in A} \beta_a(\hat{x}_a)(x_a - \hat{x}_a), \tag{7}$$

where coefficients $\beta_a(\hat{x}_a), a \in A$, are defined as follows:

$$\beta_a(\hat{x}_a) = \begin{cases} \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega d_a^\omega \xi_a^\omega \hat{y}_a^\omega, & \text{if } \hat{x}_a = 0 \\ \min_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega d_a^\omega \xi_a^\omega \hat{y}_a^\omega, & \text{if } \hat{x}_a = 1 \end{cases}, \quad \text{for all } a \in A. \tag{8}$$

Proof. Consider any interdiction solution $\bar{\mathbf{x}} \in X$. Let $\bar{P}^* = \{\bar{p}_\omega^*\}_{\omega \in \Omega}$ be an optimal probability distribution obtained by solving $D_{max}(\bar{\mathbf{x}}) = \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega Q_\omega(\bar{\mathbf{x}})$. Then, for any feasible solution $\tilde{\mathbf{y}}^\omega \in Y^\omega$, $\omega \in \Omega$, the following inequalities hold.

$$\begin{aligned} D_{max}(\bar{\mathbf{x}}) &= \sum_{\omega \in \Omega} \bar{p}_\omega^* \min_{y^\omega \in Y^\omega} \sum_{a \in A} (c_a + d_a^\omega \xi_a^\omega \bar{x}_a) y_a^\omega, \\ &\leq \sum_{\omega \in \Omega} \bar{p}_\omega^* \sum_{a \in A} (c_a + d_a^\omega \xi_a^\omega \bar{x}_a) \tilde{y}_a^\omega, \\ &\leq \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega \sum_{a \in A} (c_a + d_a^\omega \xi_a^\omega \bar{x}_a) \tilde{y}_a^\omega. \end{aligned}$$

Now, consider a different interdiction solution $\hat{\mathbf{x}}(\in X) \neq \bar{\mathbf{x}}$. Let $\hat{\mathbf{y}}_\omega \in Y^\omega$ be an optimal solution corresponding to $Q_\omega(\hat{\mathbf{x}})$ for $\omega \in \Omega$. Then,

$$D_{max}(\bar{\mathbf{x}}) \leq \max_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega \sum_{a\in A} (c_a + d_a^\omega \xi_a^\omega \bar{x}_a)\hat{y}_a^\omega, \quad \text{from the above inequalities,} \tag{9a}$$

$$= \max_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega \sum_{a\in A} (c_a + d_a^\omega \xi_a^\omega \bar{x}_a + d_a^\omega \xi_a^\omega \hat{x}_a - d_a^\omega \xi_a^\omega \hat{x}_a)\hat{y}_a^\omega, \tag{9b}$$

$$= \max_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega \sum_{a\in A} \Big\{ (c_a + d_a^\omega \xi_a^\omega \hat{x}_a)\hat{y}_a^\omega + d_a^\omega \xi_a^\omega \bar{x}_a \hat{y}_a^\omega - d_a^\omega \xi_a^\omega \hat{x}_a \hat{y}_a^\omega \Big\}, \tag{9c}$$

$$\leq \max_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega \sum_{a\in A} (c_a + d_a^\omega \xi_a^\omega \hat{x}_a)\hat{y}_a^\omega + \max_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega \sum_{a\in A} d_a^\omega \xi_a^\omega \hat{y}_a^\omega (\bar{x}_a - \hat{x}_a), \tag{9d}$$

$$= D_{max}(\hat{\mathbf{x}}) + \max_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega \sum_{a\in A} d_a^\omega \xi_a^\omega \hat{y}_a^\omega (\bar{x}_a - \hat{x}_a), \tag{9e}$$

$$\leq D_{max}(\hat{\mathbf{x}}) + \sum_{a\in A} \max_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega d_a^\omega \xi_a^\omega \hat{y}_a^\omega (\bar{x}_a - \hat{x}_a). \tag{9f}$$

For $\hat{\mathbf{x}} = \bar{\mathbf{x}}$, these inequalities hold trivially. Also, in Inequality (9e), an optimal solution to the maximization ($\max_{P\in\mathcal{P}}(\cdot)$) depends on $\bar{\mathbf{x}}$, and therefore, it does not provide a linear inequality of the form (7). Next, we showcase that

$$\max_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega d_a^\omega \xi_a^\omega \hat{y}_a^\omega (\bar{x}_a - \hat{x}_a) = \beta_a(\hat{x}_a)(\bar{x}_a - \hat{x}_a), \tag{10}$$

where $\beta_a(\hat{x}_a)$ is given by (8). As the interdiction variables are binary, the value of $(\bar{x}_a - \hat{x}_a)$ for each arc $a \in A$ could be one of the three possible values: 0, 1, or -1.

Case I. For $(\bar{x}_a - \hat{x}_a) = 0$, i.e., $\bar{x}_a = \hat{x}_a \in \{0, 1\}$, equation (10) holds trivially.

Case II. For $(\bar{x}_a - \hat{x}_a) = -1$, i.e., $\bar{x}_a = 0$ and $\hat{x}_a = 1$,

$$\beta_a(\hat{x}_a) = -\max_{P\in\mathcal{P}} \Big\{ \sum_{\omega\in\Omega} (-d_a^\omega \xi_a^\omega \hat{y}_a^\omega) p_\omega \Big\} = \min_{P\in\mathcal{P}} \Big\{ \sum_{\omega\in\Omega} (d_a^\omega \xi_a^\omega \hat{y}_a^\omega) p_\omega \Big\}.$$

Case III. For $(\bar{x}_a - \hat{x}_a) = 1$, i.e., $\bar{x}_a = 1$ and $\hat{x}_a = 0$, coefficient $\beta_a(\hat{x}_a) = \max\big\{ \sum_{\omega\in\Omega} d_a^\omega \xi_a^\omega \hat{y}_a^\omega p_\omega : P \in \mathcal{P} \big\}$.

Therefore, using (9f) and (10), we conclude that Inequality (7) is valid for all $\mathbf{x} \in X$. □

Remark. *For a deterministic NIP, i.e., when $|\Omega| = 1$, coefficient $\beta_a(\hat{x}_a) = d_a^\omega \xi_a^\omega \hat{y}_a^\omega, a \in A$. In this special case, the proposed cut reduces to the so-called global Benders cut, which was introduced for a deterministic electric power grid interdiction problem [34].*

Inequalities (7) defined for a set of interdiction solutions provide a polyhedral outer approximation of the hypograph of the function $D_{max}(\mathbf{x})$. We now present a decomposition algorithm to solve DRR-NIP to optimality, in which we iteratively construct tighter outer approximations of $D_{max}(\mathbf{x})$. Algorithm 1 provides a pseudocode for this approach and it

works as follows. To initialize the algorithm, we set iteration counter L to 1, lower bound $z^{lb}_{RR} = -\infty$, and upper bound $z^{ub}_{RR} = \infty$. We also select an initial feasible solution $\hat{\mathbf{x}}^1 \in X$. At each iteration $L \geq 1$, we solve a master problem (denoted by $\mathcal{M}^L_{RR}$), which is defined as

$$z^{ub}_{RR} := \max_{\mathbf{x}\in X,\theta} \quad \theta \tag{11a}$$

$$\text{s.t.} \quad \theta \leq D_{max}(\hat{\mathbf{x}}^l) + \sum_{a\in A} \beta_a(\hat{x}^l_a)(x_a - \hat{x}^l_a), \quad \text{for } l = 1, , ..., L, \tag{11b}$$

where $\hat{\mathbf{x}}^l \in X$ is an optimal solution to the master problem $\mathcal{M}^{l-1}_{RR}$ for $l \in \{2, \ldots, L\}$. Constraints (11b) are referred to as optimality cuts for (hypograph of) $D_{max}(\mathbf{x})$. To derive an optimality cut (11b) and a lower bound, we solve the follower's shortest path problem (4b), referred to as subproblem, for the given $\hat{\mathbf{x}}^L$ to obtain an optimal solution $\hat{\mathbf{y}}^{\omega,L}$ and compute $Q_\omega(\hat{\mathbf{x}}^L)$, for $\omega \in \Omega$ (Line 3). In Line 4, we solve a distribution separation problem, which is defined as

$$z^{dsp,L}_{RR} := \max_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega Q_\omega(\hat{\mathbf{x}}^L), \tag{12}$$

to obtain an extremal (optimal) distribution $\{\hat{p}^L_\omega\}_{\omega\in\Omega}$ and to compute $D_{max}(\hat{\mathbf{x}}^L) = z^{dsp,L}_{RR}$. If $z^{dsp,L}_{RR}$ is greater than the best-known lower bound z^{lb}_{RR}, we update the best-known lower bound and associated feasible solution $(\hat{\mathbf{x}}^*, \{\hat{\mathbf{y}}^{\omega,*}\}_{\omega\in\Omega})$ in Lines 5-7).

Algorithm 1: A decomposition algorithm for the DRR-NIP

1 **Initialization:** $L \leftarrow 1$; $z^{lb}_{RR} \leftarrow -\infty$; $z^{ub}_{RR} \leftarrow \infty$; $\hat{\mathbf{x}}^1 \in X$ *(an initial feasible solution);*
2 **while** $(z^{ub}_{RR} - z^{lb}_{RR})/z^{lb}_{RR} > \epsilon$ **do**
3 Solve subproblem (4b) to obtain optimal solution $\hat{\mathbf{y}}^{\omega,L}$ and $Q_\omega(\hat{\mathbf{x}}^L)$ for all $\omega \in \Omega$;
4 Solve distribution separation problem (12) to obtain $\{\hat{p}^L_\omega\}_{\omega\in\Omega}$ and $z^{dsp,L}_{RR} = D_{max}(\hat{\mathbf{x}}^L)$;
5 **if** $(z^{dsp,L}_{RR} > z^{lb}_{RR})$ **then**
6 $(z^{lb}_{RR}, \hat{\mathbf{x}}^*, \{\hat{\mathbf{y}}^{\omega,*}\}_{\omega\in\Omega}) \leftarrow (z^{dsp,L}_{RR}, \hat{\mathbf{x}}^L, \{\hat{\mathbf{y}}^{\omega,L}\}_{\omega\in\Omega})$; */* Update best-known lower bound and feasible solution */*
7 **end**
8 Compute $\beta_a(\hat{x}^L_a)$, $a \in A$, using (8) and add an optimality cut, i.e., (7) with $\hat{\mathbf{x}} = \hat{\mathbf{x}}^L$, to $\mathcal{M}^{L-1}_{RR}$ to get $\mathcal{M}^L_{RR}$;
9 Solve master problem $\mathcal{M}^L_{RR}$ and obtain its optimal solution $(\hat{\mathbf{x}}^{L+1}, \hat{\theta}^{L+1})$;
10 Update upper bound $z^{ub}_{RR} \leftarrow \hat{\theta}^{L+1}$;
11 $L \leftarrow L + 1$;
12 **end**
13 **Return:** $\hat{\mathbf{x}}^*, \{\hat{\mathbf{y}}^{\omega,*}\}_{\omega\in\Omega}$ *and* z^{lb}_{RR}.

Then, in Line 8, we generate an optimality cut (7) for $\hat{\mathbf{x}}^L$ as proposed in Theorem 1. Using $\hat{\mathbf{y}}^{\omega,L}$, cut coefficients $\beta_a(\hat{x}_a)$, $a \in A$, are computed by solving (8) corresponding to $\hat{\mathbf{x}} = \hat{\mathbf{x}}^L$ using the distribution separation algorithm. This generated cut is added to $\mathcal{M}^{L-1}_{RR}$ to get an updated master problem $\mathcal{M}^L_{RR}$ (a tighter outer approximation) for $L \geq 1$ where

$\mathcal{M}^0_{RR}$ is defined by (11a). We solve the updated master problem (a mixed-binary linear program), store optimal solution $(\hat{\mathbf{x}}^{L+1}, \hat{\theta}^{L+1})$, and update the best-known upper bound z^{ub}_{RR} to $\hat{\theta}^{L+1}$ (Lines 9 and 10). After that, if the optimality gap between bounds $(z^{ub}_{RR} - z^{lb}_{RR})/z^{lb}_{RR}$ is not greater than a given tolerance level $\epsilon > 0$, we terminate the algorithm and return the best-known feasible solution and lower bound (Lines 2 and 13). Otherwise, we proceed to iteration $L + 1$.

Theorem 2. *Algorithm 1 solves the DRR-NIP to optimality in a finite number of iterations if there exists a finitely convergent oracle for the distribution separation problem* (12)*.*

Proof. In Steps 3-11, the algorithm solves $|\Omega|$ number of subproblems (linear programs), master problem (a mixed-binary program), and a distribution separation problem (4) corresponding to the ambiguity set $\mathcal{P}$. It also derives an optimality cut by calling a distribution separation algorithm at most $|A|$ number of times. Under the assumption of existence of a finitely convergent distribution separation algorithm, Steps 3-11 also takes finite iterations. Therefore, to ensure the finite convergence of Algorithm 1, we only need to show that these steps are performed a finite number of times before the algorithm converges to an optimal solution. Given an optimal solution $(\hat{\mathbf{x}}^L, \hat{\theta}^L)$ of a master problem, there are two possibilities.

Case I. There does not exist any new optimality cut which indicates that the solution is feasible to the original DRR-NIP, and $\hat{\theta}^L = D_{max}(\hat{\mathbf{x}}^L) = z^{lb}_{RR} \leq z^{opt}_{RR}$. Since this solution is also an optimal solution to the master problem, $\hat{\theta}^L$ is an upper bound on z^{opt}_{RA}, i.e., $z^{ub}_{RR} = \hat{\theta}^L \geq z^{opt}_{RA}$. Therefore, the termination condition is satisfied, i.e., $z^{ub}_{RR} = z^{lb}_{RR}$, and the algorithm will terminate.

Case II. There exists a new optimality cut that is added to the master problem to cut off the violated solution $(\hat{\mathbf{x}}^L, \hat{\theta}^L)$. At iteration $K > L$, if $\hat{\mathbf{x}}^K = \hat{\mathbf{x}}^L$, then no new optimality cut is added to the master problem. Moreover, $\hat{\mathbf{x}}^K \neq \hat{\mathbf{x}}^L$ occurs finite times because the interdiction variables are binary and there are a finite number of feasible solutions in X. Therefore, the algorithm will terminate after finite iterations with an optimal solution. □

Remark. *There exists a finitely convergent oracle for the distribution separation problem associated with the ambiguity set defined as either a polyhedral set or a mixed-binary set. For a mixed-binary ambiguity set, the lift-and-project algorithm proposed by Balas et al. [1] can be used to solve the distribution separation problem.*

Remark. *According to Algorithm 1, a mixed-binary linear program $\mathcal{M}^L_{RR}$ has to be solved in each iteration $L \geq 1$. In our implementation for computational experiments, we have embedded this procedure in a branch-and-cut framework, which works as follows. We construct a single branch-and-cut tree. At each node of this tree, we solve a linear programming relaxation (node-LP) of the master problem* (11)*. If the values of the interdiction solution to the node-LP are integral, Steps 3-4, and 8 of Algorithm 1 are performed, and an optimality cut is added to the master problem if it is violated. Otherwise, we continue the typical branch-and-cut procedure, i.e., branching based on fractional values of the first-stage variables, node pruning based on bounds, and depth-first search traversal.*

2.3 Restrict-and-append Algorithm

We propose a "restrict-and-append" algorithm for the DRR-NIP where we iteratively solve a restriction of the DRR-NIP defined for a finite subset of $\mathcal{P}$ and then append the subset of distributions, if needed, by adding a new distribution. In iteration $L \geq 1$, we use Algorithm 1 in a branch-and-cut framework to solve the DRR-NIP with set $\bar{\mathcal{P}}^L$ as ambiguity set. Due to the finiteness of set $\bar{\mathcal{P}}^L$ and no closed-form representation, we solve the distribution separation problem by explicitly enumerating all distributions in $\bar{\mathcal{P}}^L$. Using $\bar{\mathcal{P}}^L$ as ambiguity set restricts the feasible region of the inner maximization in (4a), thereby leading to a restricted DRR-NIP,

$$z_{RR}^{lb} = \max_{\mathbf{x}\in X, P\in\bar{\mathcal{P}}^L} \sum_{\omega\in\Omega} p_\omega Q_\omega(\mathbf{x}). \tag{13}$$

Therefore, for each iteration, we have a lower bound on the optimal objective value of the original DRR-NIP. Let the lower bounding solution for iteration L be $\bar{\mathbf{x}}^L$. We solve the distribution separation problem using $\bar{\mathbf{x}}^L$ and the original ambiguity set $\mathcal{P}$, which we refer to as outer distribution separation problem, i.e., $D_{max}(\bar{\mathbf{x}}^L) = \max_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega Q_\omega(\bar{\mathbf{x}}^L)$, and store an optimal solution $\{\bar{p}_\omega^L\}_{\omega\in\Omega}$. If $D_{max}(\bar{\mathbf{x}}^L)$ is equal to z_{RR}^{lb}, then we terminate the algorithm and return the solution. Otherwise, we append $\bar{\mathcal{P}}^L$ by adding the new probability distribution $\{\bar{p}_\omega^L\}_{\omega\in\Omega}$ to get $\bar{\mathcal{P}}^{L+1}$ and then proceed to the next iteration of the algorithm. Notice that because of the finiteness of $|X|$ and the addition of the new probability distribution at the end of each iteration, this algorithm terminates in a finite number of iterations and provides a feasible solution and lower bound for the original DRR-NIP.

Remark. *Note that the restrict-and-append algorithm is a heuristic, and it is not guaranteed to produce an optimal solution.*

3 Exact Solution Approaches for the DRA-NIP

In this section, we present two exact solution approaches for the DRA-NIP with a general ambiguity set. We now redefine the DRA-NIP as follows:

$$z_{RA}^{opt} := \max_{\mathbf{x}\in X} \quad D_{min}(\mathbf{x}) := \min_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega Q_\omega(\mathbf{x}), \tag{14a}$$

$$\text{where } Q_\omega(\mathbf{x}) := \min_{\mathbf{y}^\omega\in Y^\omega} \sum_{a\in A} (c_a + d_a^\omega \xi_a^\omega x_a) y_a^\omega. \tag{14b}$$

Here the value functions $Q_\omega(\mathbf{x})$ provide the shortest path traveling cost for a given $\mathbf{x} \in X$ and $\omega \in \Omega$. Notice that function $Q_\omega(\mathbf{x})$ is concave on the convex hull of X. Therefore, for a given probability distribution $\{p_\omega\}_{\omega\in\Omega} \in \mathcal{P}$, the convex combination of the concave functions $Q_\omega(\mathbf{x})$ for $\omega \in \Omega$, i.e., $\sum_{\omega\in\Omega} p_\omega Q_\omega(\mathbf{x})$, is also concave. This concludes that the value function $D_{min}(\mathbf{x})$, is also concave because it is point-wise minimum of concave functions. Furthermore, problem (14) is also equivalent to solving the following semi-infinite optimization problem:

$$\max_{\mathbf{x}\in X,\theta} \quad \theta \tag{15a}$$

$$\text{s.t.} \quad \theta \leq \sum_{\omega\in\Omega} p_\omega \sum_{a\in A} (c_a + d_a^\omega \xi_a^\omega x_a)\hat{y}_a^\omega, \quad \text{for all } \hat{\mathbf{y}}^\omega \in \hat{Y}_E^\omega,\ \omega\in\Omega, \text{ and } \{p_\omega\}_{\omega\in\Omega} \in \mathcal{P}, \tag{15b}$$

where $\hat{Y}_E^\omega$ is the set of all extreme points of Y^ω for $\omega \in \Omega$.

3.1 Decomposition Approach for DRA-NIP

We present a decomposition algorithm for the DRA-NIP in which we iteratively obtain outer approximation of the function $D_{min}(\mathbf{x})$. A pseudocode of the proposed decomposition procedure is given in Algorithm 2, which works as follows. We start the algorithm using an initial feasible solution $\hat{\mathbf{x}}^1 \in X$. We also initialize iteration counter L, lower bound z_{RA}^{lb}, and upper bound z_{RA}^{ub} to 1, $-\infty$, and ∞, respectively. For each iteration $L \geq 1$, we compute $Q_\omega(\hat{\mathbf{x}}^L)$ by solving subproblem (14b) for all $\omega \in \Omega$ and store their optimal solutions $\hat{\mathbf{y}}^{\omega,L}, \omega \in \Omega$ (Line 3). Using $Q_\omega(\hat{\mathbf{x}}^L)$, we then solve a distribution separation problem $D_{min}(\hat{\mathbf{x}}^L)$, i.e.,

$$z_{RA}^{dsp,L} := \min_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega Q_\omega(\hat{\mathbf{x}}^L), \tag{16}$$

to get an extremal (optimal) distribution $\{\hat{p}_\omega^L\}_{\omega\in\Omega}$ (Line 4). If $z_{RA}^{dsp,L}$ is greater than the current lower bound z_{RA}^{lb}, we update the lower bound and store the current solution as $\hat{\mathbf{x}}^*$ and $\{\hat{\mathbf{y}}^{\omega,*}\}_{\omega\in\Omega}$ (Lines 5-7). Otherwise, we derive a so-called optimality cut using $\{\hat{p}_\omega^L\}_{\omega\in\Omega}$ and $\{\hat{\mathbf{y}}^{\omega,L}\}_{\omega\in\Omega}$, i.e.,

$$\theta \leq \sum_{\omega\in\Omega} \hat{p}_\omega^L \sum_{a\in A} (c_a + d_a^\omega \xi_a^\omega x_a)\hat{y}_a^{\omega,L},$$

and add it to the master problem $\mathcal{M}_{RA}^{L-1}$ (Line 8):

$$z_{RA}^{ub} := \max_{\mathbf{x}\in X,\theta} \quad \theta \tag{17a}$$

$$\text{s.t.} \quad \theta \leq \sum_{\omega\in\Omega} \hat{p}_\omega^l \sum_{a\in A} (c_a + d_a^\omega \xi_a^\omega x_a)\hat{y}_a^{\omega,l}, \quad \text{for all } l = 1,\ldots,L-1, \tag{17b}$$

where $\{\hat{p}_\omega^l\}_{\omega\in\Omega} \in \mathcal{P}$ and $\hat{\mathbf{y}}^{\omega,l} = \{\hat{y}_a^{\omega,l}\}_{a\in A} \in \hat{Y}_E^\omega$, $\omega \in \Omega$, for $l = 1,\ldots,L-1$. Afterwards, we solve the updated master problem $\mathcal{M}_{RA}^L$, obtain a new solution $(\hat{x}^{L+1}, \hat{\theta}^{L+1})$, and update the upper bound z_{RA}^{ub} (Lines 9-10). If the optimality gap $(z_{RA}^{ub} - z_{RA}^{lb})/z_{RA}^{lb}$ is below a given tolerance level $\epsilon > 0$, we return the best solution and terminate the algorithm (Lines 2 and 13). Otherwise, we proceed with the next iteration $L+1$.

Remark. *In Algorithm 2, $\mathcal{M}_{RA}^L$ is a mixed-binary program and solvable using any off-the-shelf solver for mixed-integer programs. However, we have embedded the decomposition procedure in a branch-and-cut framework for our computational experiments, which works as follows. At each node of the branch-and-cut tree, we solve a linear programming relaxation (node-LP) of the relaxed master problem* (17)*. If the values of the interdiction solution to the node-LP are integral, Steps 3-4 of Algorithm 2 are performed, and an optimality cut is added to the master problem if it is violated. Otherwise, we continue the typical branch-and-cut procedure.*

Theorem 3. *Algorithm 2 is finitely convergent to an optimal solution of the DRA-NIP if there exists a finitely convergent oracle for the distribution separation problem* (16)*.*

Proof. The proof of Theorem 3 follows the similar arguments as in the proof of Theorem 2, except that an optimality cut is defined by (17b), instead of (7). □

Algorithm 2: A decomposition algorithm for the DRA-NIP

Initialization: $L \leftarrow 1$; $z_{RA}^{lb} \leftarrow -\infty$; $z_{RA}^{ub} \leftarrow \infty$, $\hat{\mathbf{x}}^1 \leftarrow$ *any* $\mathbf{x} \in X$;
while $(z_{RA}^{ub} - z_{RA}^{lb})/z_{RA}^{lb} > \epsilon$ **do**
  Solve subproblem (14b) to obtain optimal solution $\hat{\mathbf{y}}^{\omega,L}$ and $Q_\omega(\hat{\mathbf{x}}^L)$ for all $\omega \in \Omega$;
  Solve distribution separation problem (16) to obtain $\{\hat{p}_\omega^L\}_{\omega\in\Omega}$ and $z_{RA}^{dsp,L} = D_{min}(\hat{\mathbf{x}}^L)$;
  if $(z_{RA}^{dsp,L} > z_{RA}^{lb})$ **then**
    $(z_{RA}^{lb}, \hat{\mathbf{x}}^*, \{\hat{\mathbf{y}}^{\omega,*}\}_{\omega\in\Omega}) \leftarrow (z_{RA}^{dsp,L}, \hat{\mathbf{x}}^L, \{\hat{\mathbf{y}}^{\omega,L}\}_{\omega\in\Omega})$;
    / Update best-known lower bound and feasible solution */*
  end
  Add an optimality cut to $\mathcal{M}_{RA}^{L-1}$ to get $\mathcal{M}_{RA}^{L}$;
  Solve master problem $\mathcal{M}_{RA}^{L}$ and obtain its optimal solution $(\hat{\mathbf{x}}^{L+1}, \hat{\theta}^{L+1})$;
  Update upper bound $z_{RA}^{ub} \leftarrow \hat{\theta}^{L+1}$;
  $L \leftarrow L + 1$;
end
Return: $\hat{\mathbf{x}}^*, \{\hat{\mathbf{y}}^{\omega,*}\}_{\omega\in\Omega}$ *and* z_{RA}^{lb}.

3.2 Relax-and-Append Algorithm for DRA-NIP

We present a "relax-and-append" algorithm to solve the DRA-NIP which is similar to the restrict-and-append algorithm for the DRR-NIP. More specifically in iteration $L \geq 1$, we use a branch-and-cut algorithm with Algorithm 2 to solve DRA-NIP with a finite set $\bar{\mathcal{P}}^L \subset \mathcal{P}$ as ambiguity set. Since $\bar{\mathcal{P}}^L$ is a finite set without any closed-form representation, we solve the distribution separation problem by explicitly enumerating all distributions. Note that solving DRA-NIP with $\bar{\mathcal{P}}^L$ will produce a relaxed solution for the original DRA-NIP which yields an upper bound on z_{RA}^{opt}:

$$z_{RA}^{ub,L} = \max_{\mathbf{x}\in X} \min_{P\in\bar{\mathcal{P}}^L} \sum_{\omega\in\Omega} p_\omega Q_\omega(\mathbf{x}). \tag{18}$$

Let an optimal solution for iteration L be $\bar{\mathbf{x}}^L$. We solve an outer distribution separation problem using $\bar{\mathbf{x}}^L$ and the original ambiguity set $\mathcal{P}$, i.e., $z_{RA}^{lb,L} = D_{min}(\bar{\mathbf{x}}^L) = \min_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega Q_\omega(\bar{\mathbf{x}}^L)$, which gives a lower bound on z_{RA}^{opt}. If $D_{min}(\bar{\mathbf{x}}^L) = z_{RA}^{ub,L}$, then the current interdiction solution $\bar{\mathbf{x}}^L$ is optimal, and we terminate the algorithm. Otherwise, we add the new probability distribution obtained by solving the outer distribution separation problem to $\bar{\mathcal{P}}^L$ and proceed to iteration $L + 1$.

Theorem 4. *The relax-and-append algorithm solves the DRA-NIP to optimality in finitely many iterations, if there exists a finitely convergent oracle to solve the outer distribution separation problem* (16)*.*

Proof. In Iteration $L \geq 1$ of the relax-and-append algorithm, Algorithm 2 (within branch-and-cut framework) solves a relaxation of the DRA-NIP where a distribution separation

algorithm is to explicitly enumerate all probability distributions in a finite set $\bar{\mathcal{P}}^L \subset \mathcal{P}$. This takes a finite number of operations because of Theorem 3. Also, by the assumption, we can solve an outer distribution separation problem using a finitely convergent oracle. Next, we need to show that the relax-and-append algorithm satisfies the termination condition in finite iterations, i.e., $L < \infty$. Let an optimal solution for the relaxation of the DRA-NIP in iteration L be $\bar{\mathbf{x}}^L$ that yields an upper bound $z_{RA}^{ub,L}$. Also, assume that an optimal probability distribution provided by the outer distribution separation problem for $\mathbf{x} = \bar{\mathbf{x}}^L$ be $\bar{P}^L$. Now, one of the following two cases can occur: (I) $\bar{\mathbf{x}}^L = \bar{\mathbf{x}}^{L'}$ for some $L' < L$; or (II) $\bar{\mathbf{x}}^L \neq \bar{\mathbf{x}}^l$ for all $l < L$. In Case I, since the current relaxed ambiguity set $\bar{\mathcal{P}}^L$ already contains probability distribution $\bar{P}^{L'}$, we get $D_{min}(\bar{\mathbf{x}}^L) = D_{min}(\bar{\mathbf{x}}^{L'}) = z_{RA}^{ub,L}$. Hence, current solution $\bar{\mathbf{x}}^L$ is an optimal solution. In Case II, either of the two subcases can occur: (A) there exists an optimal distribution $\bar{P}^L$ that belongs to $\bar{\mathcal{P}}^L$, which again implies that $D_{min}(\bar{\mathbf{x}}^L) = z_{RA}^{ub,L}$ and the current solution is optimal; (B) if there does not exist an optimal distribution $\bar{P}^L$ that belongs to $\bar{\mathcal{P}}^L$, then we proceed to the next iteration with $\bar{\mathcal{P}}^{L+1} = \bar{\mathcal{P}}^L \cup \{\bar{P}^L\}$. Since $|X|$ is finite, there will be only finite iterations until the relax-and-append algorithm leads to Cases I or II(A), thereby returning an optimal solution. □

Remark. *In contrast to the restrict-and-append algorithm, the relax-and-append algorithm is guaranteed to produce an optimal solution.*

4 Computational Results

In this section, we present the results of the computational tests conducted for evaluating the effectiveness and efficiency of the proposed approaches for the DRR-NIP and the DRA-NIP. We implemented all algorithms in Java and Gurobi 9.0. All tests were conducted on a machine with an Intel Core i7 processor (3.8GHz) and 32GB of memory. We set the time limit for each experiment to 3600 seconds and terminate an algorithm when the optimality gap becomes less than 10^{-4}. For decomposition-based algorithms, we add optimality cuts as lazy constraints through Gurobi Callback feature. To this end, we disable presolve and set the number of threads to one. For the remaining Gurobi parameters, we use its default settings.

We consider two examples of ambiguity sets studied in the literature: the *moment matching set* and the *Wasserstein ambiguity set*. The moment matching set, denoted by $\mathcal{P}_M$, is defined by a polytope, restricting moments of random variables regarding some matching bounds. Let $\mathcal{F}$ be a sigma-algebra of Ω and f_q be real-valued measurable functions on $(\Omega, \mathcal{F})$ for $q = 1, ..., m$. Then, the moment matching set is defined as

$$\mathcal{P}_M := \Bigg\{ P = \{p_\omega\}_{\omega\in\Omega} : \underline{u}_q \leq \sum_{\omega\in\Omega} p_\omega f_q(\omega) \leq \bar{u}_q, \text{ for } q = 1, ..., m; \\ \sum_{\omega\in\Omega} p_\omega = 1; \ p_\omega \geq 0, \text{ for all } \omega \in \Omega \Bigg\}, \tag{19}$$

where $\underline{u}_q$ and $\bar{u}_q$ denote the predetermined component-wise lower and upper bounds, respec-

tively. For our experiments, we consider the component-wise moment inequalities, i.e.,

$$f_1(\omega) = \xi_1^\omega,\ f_2(\omega) = \xi_2^\omega,\ \ldots,\ f_{|A|}(\omega) = \xi_{|A|}^\omega,$$
$$f_{|A|+1}(\omega) = (\xi_1^\omega)^2,\ f_{|A|+2}(\omega) = (\xi_2^\omega)^2,\ \ldots,\ f_{2\times|A|}(\omega) = (\xi_{|A|}^\omega)^2.$$

Since we consider Bernoulli random variables, i.e., $\xi_a^\omega \in \{0,1\}$ for all $\omega \in \Omega, a \in A$, the constraints associated with the second moment are redundant as $f_{|A|+i}(\omega) = f_i(\omega)$ for $i \in \{1,\ldots,|A|\}$. We also consider the use of Wasserstein metric to specify the ambiguity set, which is known as Wasserstein ambiguity set (denoted by $\mathcal{P}_W$). Let $P^* = \{p_\omega^*\}_{\omega\in\Omega}$ be a reference probability distribution, which is given beforehand. Wasserstein ambiguity set $\mathcal{P}_W$ consists of all probability distributions within ϵ_W-Wasserstein distance from P^*, where $\epsilon_W > 0$ is given. Let $\|\cdot\|_1$ be L_1-norm on $\mathbb{R}^{|A|}$. Then, for $\Omega = \{\omega_1, ..., \omega_{|\Omega|}\}$, $\mathcal{P}_W$ is equivalent to

$$\begin{aligned}\mathcal{P}_W := \Bigg\{ P = \{p_\omega\}_{\omega\in\Omega} : \sum_{\omega\in\Omega} p_\omega = 1;\ \sum_{\omega_i\neq\omega_j\in\Omega} \|\omega_i - \omega_j\|_1 v_{\omega_i,\omega_j} \le \epsilon_W;\\ \sum_{\omega_j\in\Omega} v_{\omega_i,\omega_j} = p_{\omega_i},\ \text{ for all } \omega_i \in \Omega;\ \sum_{\omega_i\in\Omega} v_{\omega_i,\omega_j} = p^*_{\omega_j},\ \text{ for all } \omega_j \in \Omega;\\ p_\omega \ge 0,\ \text{for all } \omega\in\Omega;\ v_{\omega_i,\omega_j} \ge 0,\ \text{for all } \omega_i,\omega_j \in \Omega \Bigg\}.\end{aligned} \tag{20}$$

Since $\mathcal{P}_M$ and $\mathcal{P}_W$ are polyhedral sets, the corresponding distribution separation problems are linear programs, and to solve them, we use Gurobi's primary simplex algorithm because it performed better than the others in our preliminary tests for DRR-NIP and DRA-NIP instances.

For the restrict-and-append algorithm and the relax-and-append algorithm, we initialize a subset $\bar{\mathcal{P}}^1$ with a probability distribution $\bar{P}^0$ which is obtained by solving the following feasibility problems: $\text{argmax}_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega$ and $\text{argmin}_{P\in\mathcal{P}} \sum_{\omega\in\Omega} p_\omega$, respectively.

4.1 Instance Generation and Experiment Setup

We generate DRR-NIP and DRA-NIP test instances based on the instances from Nguyen and Smith [27], with some modifications. We consider networks with 40, 60, 80, and 100 nodes, and label them as N40, N60, N80, and N100, respectively. In the instances of [27], around 20% of all arcs in the network have the traveling cost uncertainty where $c_a^\omega \in [c_a^L, c_a^U]$ for $a \in A$. To modify these instances for DRR-NIP and DRA-NIP, we set $c_a = 0.5(c_a^L + c_a^U)$ and randomly generate interdiction successes $\xi_a^\omega, \omega \in \Omega$ (for arcs with uncertain cost in the original instances) from a Bernoulli distribution with the probability of success of 0.75. For the remaining 80% of the arcs, $\xi_a^\omega = 1$ for all $\omega \in \Omega$. The impact of successful interdiction on each arc, $d_a, a \in A$, is deterministic and directly taken from the instances of [27]. We use label US20 for these instances. In addition, we generate instances where $\xi_a^\omega, \omega \in \Omega$ is uncertain for all $a \in A$ and label them as US100. For each instance category, Nxx and USxx, we perform tests with different interdiction budgets $b \in \{2, 5, 10\}$, scenario sizes $|\Omega| \in \{100, 500, 1000\}$, and ambiguity sets $\mathcal{P} \in \{\mathcal{P}_M, \mathcal{P}_W\}$. For $\mathcal{P}_M$, we compute the first

moment of random variable, i.e., $u_a = \sum_{\omega \in \Omega} \xi_a^\omega / |\Omega|, a \in A$. Then, given tolerance level $\epsilon_M = 0.05$, we compute the bounds, $\underline{u}_a = (1 - \epsilon_M)u_a$ and $\bar{u}_a = (1 + \epsilon_M)u_a, a \in A$. Likewise, for $\mathcal{P}_W$, we set ϵ_W equal to ρ (= 0.1, by default) times the average of L_1-norm distance between all observations ω_i and ω_j in Ω.

4.2 Computational Results for the DRR-NIP

In Tables 2 and 3, we report the computational results of the proposed algorithms for US20 and US100 instances, respectively, of DRR-NIP with the moment matching set and the Wasserstein ambiguity set. The columns corresponding to results for the reformulation-based algorithm, the cut-based decomposition algorithm, and the restrict-and-append algorithm are labeled as "Reformulation," "Decomposition," and "RestrictAppend," respectively. Each row reports average results for ten instances. For each algorithm, we report the average computational time, T in seconds, for the instances solved with the time limit of 1 hour (3600 seconds). For the instances unsolved within the time limit, we report the average optimality gap (Gap %) and the number of unsolved instances out of ten in parenthesis. When all ten instances are solved within the time limit, we use label "ZG" to denote zero optimality gap. In contrast, if none of the ten instances is solved within the time limit, then we use label "3600+." Since the restrict-and-append algorithm is a heuristic, we present a relative gap between the best solution value obtained from the restrict-and-append algorithm and the upper bound obtained from the cut-based decomposition algorithm. In the column labeled "RelativeGap," we report the average of relative gaps that are greater than 10^{-4} and the number of corresponding instances out of ten in parenthesis. We use label ZRG—zero relative gap—when the relative gaps are less than 10^{-4} for all instances.

From the results in Table 2, we observe that the number of scenarios significantly impacts the performance of the reformulation-based algorithm, even for the relatively smaller-sized instances (N40 and N60), because the size of the reformulation (6) increases with the increase in the number of scenarios. In contrast, the cut-based decomposition approach is 395 times (on average) faster for the instances solved by both the algorithms. Interestingly, the restrict-and-append algorithm produces equal or better solutions than the cut-based decomposition algorithm for 541 out of 600 (around 90 %) instances, and the former is 4.2 times (on average) faster than the latter in solving the instances that the cut-based decomposition algorithm solves to optimality. Note that the restrict-and-append algorithm does not guarantee to provide an optimal solution, and we are able to evaluate the quality of its solution only because the cut-based decomposition algorithm provides an optimal solution in finite iterations.

In Table 3, we report the computational results for solving the DRR-NIP US100 instances using the proposed algorithms. The cut-based decomposition algorithm is 34 times (on average) faster than the reformulation-based algorithm for the optimally solved instances. Moreover, for the unsolved instances, the former provides the optimality gap that is 38% (on average) of the optimality gap reported by the latter. We can observe that even the cut-based decomposition algorithm cannot solve around 84 % (201 out of 240) of the larger-sized instances in N80 and N100 due to the higher level of uncertainty than the US20 instances. This is because generating an optimality cut is computationally more expensive for the US100 instances, since the algorithm has to compute the cut coefficients $\beta_a(x_a)$ for all arcs

Table 2: Computational results for solving DRR-NIP US20 instances using reformulation-based algorithm, cut-based decomposition algorithm, and restrict-and-append algorithm.

Instances			Moment Matching Set						Wasserstein Ambiguity Set					
			Reformulation		Decomposition		RestrictAppend		Reformulation		Decomposition		RestrictAppend	
Category	b	$\|\Omega\|$	T(sec)	**Gap**	T(sec)	**Gap**	T(sec)	**RelativeGap**	T(sec)	**Gap**	T(sec)	**Gap**	T(sec)	**RelativeGap**
N40	2	100	19.7	ZG (0)	0.3	ZG (0)	0.2	ZRG (0)	29.5	ZG (0)	0.2	ZG (0)	0.2	ZRG (0)
		500	209	ZG (0)	1.0	ZG (0)	0.7	ZRG (0)	1713	ZG (0)	1.4	ZG (0)	0.8	ZRG (0)
		1000	1357	ZG (0)	1.7	ZG (0)	1.5	ZRG (0)	3285	4.79% (9)	4.0	ZG (0)	1.5	ZRG (0)
	5	100	506	ZG (0)	0.9	ZG (0)	0.6	0.01% (1)	227	ZG (0)	0.8	ZG (0)	0.7	0.02% (1)
		500	1923	1.04% (4)	2.9	ZG (0)	2.4	ZRG (0)	2975	1.55% (8)	5.3	ZG (0)	2.7	0.04% (1)
		1000	2775	2.07% (7)	6.4	ZG (0)	4.5	ZRG (0)	3279	4.38% (9)	15.7	ZG (0)	5.6	0.04% (1)
	10	100	2083	1.17% (5)	3.4	ZG (0)	2.0	0.08% (1)	1801	0.88% (4)	3.7	ZG (0)	2.0	0.1% (3)
		500	2284	1.90% (6)	13.3	ZG (0)	7.3	0.15% (2)	2552	1.79% (7)	73.8	ZG (0)	6.0	0.12% (2)
		1000	2439	3.10% (6)	23.4	ZG (0)	12.4	0.12% (3)	2976	5.25% (8)	387	ZG (0)	13.0	0.08% (3)
N60	2	100	97.3	ZG (0)	0.6	ZG (0)	0.4	ZRG (0)	49.6	ZG (0)	0.6	ZG (0)	0.4	0.11% (1)
		500	967	ZG (0)	2.7	ZG (0)	1.8	ZRG (0)	2579	1.27% (2)	3.1	ZG (0)	2.1	0.06% (1)
		1000	3200	2.25% (6)	4.8	ZG (0)	3.5	ZRG (0)	3600+	3.23% (10)	8.8	ZG (0)	3.9	0.05% (1)
	5	100	1995	0.61% (4)	3.1	ZG (0)	1.6	ZRG (0)	705	ZG (0)	2.5	ZG (0)	1.6	ZRG (0)
		500	3114	1.45% (7)	10.5	ZG (0)	5.6	ZRG (0)	3600+	1.44% (10)	12.8	ZG (0)	6.4	ZRG (0)
		1000	3600+	5.29% (10)	21.2	ZG (0)	11.6	ZRG (0)	3600+	5.65% (10)	34.6	ZG (0)	12.2	ZRG (0)
	10	100	3289	1.13% (8)	314	ZG (0)	11.0	0.06% (2)	3059	0.63% (7)	25.9	ZG (0)	15.7	0.09% (3)
		500	3600+	2.08% (10)	483	0.48% (1)	24.4	0.16% (5)	3600+	1.68% (10)	358	ZG (0)	35.4	0.07% (2)
		1000	3600+	4.33% (10)	612	0.95% (1)	45.6	0.26% (5)	3600+	5.34% (10)	1222	1.11% (2)	69.0	0.45% (4)
N80	5	100	3136	0.78% (5)	30.0	ZG (0)	4.2	0.01% (1)	1848	0.25% (1)	5.6	ZG (0)	4.0	0.03% (1)
		500	3600+	3.27% (10)	129	ZG (0)	13.7	0.06% (1)	3600+	1.76% (10)	80.7	ZG (0)	15.5	0.01% (1)
		1000	3600+	8.98% (10)	246	ZG (0)	28.2	0.06% (1)	3600+	11.48% (10)	373	ZG (0)	30.7	0.02% (1)
	10	100	3600+	1.40% (10)	515	0.14% (1)	72.3	0.14% (1)	3600+	0.85% (10)	111	ZG (0)	71.8	0.04% (1)
		500	3600+	4.02% (10)	1062	0.92% (2)	130	0.59% (3)	3600+	2.93% (10)	631	0.35% (1)	132	0.21% (2)
		1000	3600+	8.48% (10)	1331	1.13% (2)	206	1.09% (2)	3600+	8.25% (10)	1668	1.66% (1)	205	0.72% (2)
N100	5	100	3463	0.93% (9)	75.1	ZG (0)	13.3	ZRG (0)	2782	0.59% (4)	16.6	ZG (0)	12.6	0.05% (1)
		500	3600+	8.15% (10)	370	ZG (0)	43.6	ZRG (0)	3600+	3.11% (10)	166	ZG (0)	44.3	0.02% (2)
		1000	3600+	11.81% (10)	609	ZG (0)	92.1	ZRG (0)	3600+	13.81% (10)	730	ZG (0)	96.6	0.03% (1)
	10	100	3600+	1.86% (10)	771	0.64% (1)	221	0.66% (1)	3600+	1.25% (10)	393	ZG (0)	225	ZRG (0)
		500	3600+	9.82% (10)	1837	1.2% (4)	441	1.01% (4)	3600+	3.95% (10)	927	1.56% (1)	375	1.38% (1)
		1000	3600+	12.79% (10)	2132	1.71% (4)	695	1.28% (4)	3600+	13.21% (10)	2307	0.93% (4)	567	0.85% (4)

Table 3: Computational results for solving DRR-NIP US100 instances using reformulation-based algorithm, cut-based decomposition algorithm, and restrict-and-append algorithm.

Instances			Moment Matching Set						Wasserstein Ambiguity Set					
			Reformulation		Decomposition		RestrictAppend		Reformulation		Decomposition		RestrictAppend	
Category	b	$\|\Omega\|$	T(sec)	**Gap**	T(sec)	**Gap**	T(sec)	**RelativeGap**	T(sec)	**Gap**	T(sec)	**Gap**	T(sec)	**RelativeGap**
N40	2	100	75.5	ZG (0)	6.2	ZG (0)	0.9	0.02% (2)	20.5	ZG (0)	1.2	ZG (0)	0.9	0.02% (4)
		500	916	ZG (0)	31.9	ZG (0)	3.3	0.16% (1)	907	ZG (0)	29.6	ZG (0)	3.4	0.04% (1)
		1000	2470	1.89% (3)	66.7	ZG (0)	6.3	0.01% (1)	3152	2.31% (7)	155	ZG (0)	6.6	0.02% (1)
	5	100	2229	0.99% (3)	165	ZG (0)	5.2	0.05% (2)	1293	0.76% (1)	20.8	ZG (0)	4.9	0.06% (3)
		500	3241	2.18% (9)	1148	1.61% (1)	15.8	0.41% (5)	3243	2.11% (9)	819	0.46% (1)	20.9	0.13% (4)
		1000	3242	2.96% (9)	1812	2.40% (1)	29.0	0.47% (6)	3268	4.39% (9)	2425	0.95% (4)	37.7	0.76% (5)
	10	100	3240	2.06% (9)	2549	0.96% (6)	208	0.86% (7)	3124	1.10% (8)	1081	1.25% (1)	398	0.36% (4)
		500	3241	3.70% (9)	3244	2.65% (9)	189	2.53% (9)	3243	3.26% (9)	3236	1.57% (8)	534	1.52% (8)
		1000	3242	4.52% (9)	3245	3.14% (9)	466	2.97% (9)	3267	5.38% (9)	3246	2.38% (9)	639	2.22% (9)
N60	2	100	305	ZG (0)	21.1	ZG (0)	1.7	0.06% (1)	49.5	ZG (0)	2.1	ZG (0)	1.6	0.02% (2)
		500	3600+	1.54% (10)	178	ZG (0)	6.6	0.03% (2)	1671	ZG (0)	46.7	ZG (0)	7.7	0.01% (1)
		1000	3600+	4.75% (10)	493	ZG (0)	13.4	0.11% (1)	3600+	2.65% (10)	272	ZG (0)	15.0	0.02% (2)
	5	100	3474	1.52% (9)	694	ZG (0)	16.7	0.05% (3)	2354	0.77% (4)	52.8	ZG (0)	16.3	0.04% (4)
		500	3600+	4.05% (10)	3334	1.12% (8)	54.5	0.89% (10)	3600+	2.13% (10)	1517	0.31% (2)	61.4	0.14% (6)
		1000	3600+	7.14% (10)	3600+	1.56% (10)	103	1.52% (10)	3600+	3.87% (10)	3110	0.93% (5)	120	0.59% (8)
	10	100	3600+	2.80% (10)	3408	1.17% (9)	789	1.12% (9)	3600+	1.10% (10)	2411	0.70% (6)	844	0.59% (7)
		500	3600+	4.49% (10)	3600+	2.69% (10)	1177	2.54% (10)	3600+	3.28% (10)	3600+	1.36% (10)	1784	1.31% (10)
		1000	3600+	6.49% (10)	3600+	3.40% (10)	1829	3.13% (10)	3600+	4.80% (10)	3600+	1.79% (10)	2266	1.71% (10)
N80	5	100	3600+	2.25% (10)	2834	0.52% (6)	49.9	0.38% (8)	3600	1.11% (9)	174	ZG (0)	52.9	0.03% (4)
		500	3600+	8.18% (10)	3600+	1.68% (10)	192	1.58% (10)	3600+	2.64% (10)	2490	0.56% (4)	227	0.48% (5)
		1000	3600+	9.78% (10)	3600+	2.55% (10)	371	2.08% (10)	3600+	7.39% (10)	3600+	0.93% (10)	518	0.90% (10)
	10	100	3600+	3.41% (10)	3600+	1.72% (10)	2188	1.60% (10)	3600+	1.74% (10)	3507	0.73% (9)	2172	0.71% (9)
		500	3600+	7.96% (10)	3600+	3.31% (10)	2706	2.98% (10)	3600+	4.01% (10)	3600+	1.65% (10)	2954	1.56% (10)
		1000	3600+	10.19% (10)	3600+	5.19% (10)	3058	4.07% (10)	3600+	6.82% (10)	3600+	1.98% (10)	3375	1.86% (10)
N100	5	100	3600+	2.57% (10)	3456	0.71% (9)	74.6	0.68% (9)	3600+	1.09% (10)	142	ZG (0)	69.8	0.01% (3)
		500	3600+	8.63% (10)	3600+	2.07% (10)	306	1.68% (10)	3600+	2.85% (10)	2507	0.48% (4)	319	0.46% (4)
		1000	3600+	12.53% (10)	3600+	3.37% (10)	595	2.48% (10)	3600+	8.84% (10)	3506	0.97% (9)	657	0.93% (9)
	10	100	3600+	3.38% (10)	3600+	1.84% (10)	2768	1.69% (10)	3600+	2.04% (10)	3600+	0.74% (10)	2963	0.73% (10)
		500	3600+	10.82% (10)	3600+	4.57% (10)	3420	3.38% (10)	3600+	3.80% (10)	3600+	1.59% (10)	3577	1.48% (10)
		1000	3600+	11.44% (10)	3600+	6.23% (10)	3600+	4.98% (10)	3600+	9.69% (10)	3600+	1.94% (10)	3600+	1.76% (10)

Table 4: Computational results for solving DRA-NIP US20 instances using decomposition algorithm and relax-and-append algorithm.

Instances			Moment Matching Set					Wasserstein Ambiguity Set				
			Decomposition		RelaxAppend			Decomposition		RelaxAppend		
Category	b	$\|\Omega\|$	T(sec)	**Gap**	T(sec)	**Gap**	**#Iter**	T(sec)	**Gap**	T(sec)	**Gap**	**#Iter**
N40	2	100	0.3	ZG (0)	0.2	ZG (0)	1	0.2	ZG (0)	0.2	ZG (0)	1.1
		500	0.9	ZG (0)	0.9	ZG (0)	1.1	1.2	ZG (0)	0.8	ZG (0)	1.1
		1000	1.5	ZG (0)	1.5	ZG (0)	1.1	3.6	ZG (0)	1.6	ZG (0)	1.1
	5	100	0.6	ZG (0)	0.9	ZG (0)	1.5	0.6	ZG (0)	1.1	ZG (0)	1.8
		500	2.4	ZG (0)	2.9	ZG (0)	1.4	3.7	ZG (0)	4.2	ZG (0)	1.8
		1000	3.9	ZG (0)	6.3	ZG (0)	1.5	10.0	ZG (0)	8.1	ZG (0)	1.8
	10	100	1.8	ZG (0)	3.7	ZG (0)	2.2	1.7	ZG (0)	4.0	ZG (0)	2.1
		500	4.9	ZG (0)	10.3	ZG (0)	1.9	10.3	ZG (0)	18.8	ZG (0)	2.6
		1000	9.3	ZG (0)	16.5	ZG (0)	1.7	32.5	ZG (0)	33.6	ZG (0)	2.6
N60	2	100	0.6	ZG (0)	0.6	ZG (0)	1.2	0.4	ZG (0)	0.7	ZG (0)	1.3
		500	1.9	ZG (0)	2.1	ZG (0)	1.1	2.7	ZG (0)	2.7	ZG (0)	1.3
		1000	4.0	ZG (0)	4.3	ZG (0)	1.1	6.4	ZG (0)	4.8	ZG (0)	1.3
	5	100	1.7	ZG (0)	1.8	ZG (0)	1.3	1.3	ZG (0)	1.9	ZG (0)	1.4
		500	6.7	ZG (0)	6.0	ZG (0)	1.2	7.3	ZG (0)	6.8	ZG (0)	1.4
		1000	10.4	ZG (0)	11.4	ZG (0)	1.2	18.6	ZG (0)	14.5	ZG (0)	1.4
	10	100	9.2	ZG (0)	11.6	ZG (0)	1.6	9.3	ZG (0)	25.5	ZG (0)	2.4
		500	17.9	ZG (0)	32.9	ZG (0)	1.5	35.1	ZG (0)	66.2	ZG (0)	2.3
		1000	32.3	ZG (0)	59.7	ZG (0)	1.8	87.8	ZG (0)	136	ZG (0)	2.4
N80	5	100	5.6	ZG (0)	5.3	ZG (0)	1.4	3.9	ZG (0)	5.5	ZG (0)	1.4
		500	17.8	ZG (0)	16.7	ZG (0)	1.2	17.6	ZG (0)	20.5	ZG (0)	1.4
		1000	28.3	ZG (0)	32.7	ZG (0)	1.3	40.7	ZG (0)	38.3	ZG (0)	1.4
	10	100	61.5	ZG (0)	107	ZG (0)	1.9	48.1	ZG (0)	105	ZG (0)	1.9
		500	87.8	ZG (0)	176	ZG (0)	1.5	111	ZG (0)	219	ZG (0)	1.9
		1000	136	ZG (0)	309	ZG (0)	1.8	239	ZG (0)	351	ZG (0)	1.9
N100	5	100	14.6	ZG (0)	11.1	ZG (0)	1.1	11.1	ZG (0)	17.3	ZG (0)	1.4
		500	60.2	ZG (0)	42.8	ZG (0)	1.1	49.4	ZG (0)	59.8	ZG (0)	1.4
		1000	89.4	ZG (0)	98.0	ZG (0)	1.3	108	ZG (0)	118	ZG (0)	1.4
	10	100	237	ZG (0)	372	ZG (0)	1.3	275	ZG (0)	444	ZG (0)	1.4
		500	443	ZG (0)	522	0.02% (1)	1.2	526	ZG (0)	715	0.07% (1)	1.4
		1000	524	ZG (0)	585	0.09% (1)	1.2	682	0.08% (1)	755	0.07% (1)	1.4

$a \in A$ that have uncertainty. Similar to the results for the US20 instances, the restrict-and-append algorithm produces equal or better solutions than the cut-based decomposition algorithm for 484 out of 600 (around 81 %) instances. For the instances that the cut-based decomposition algorithm solves to optimality, the restrict-and-append algorithm is 22 times faster on average; whereas for the remaining instances, the restrict-and-append algorithm takes 1539 seconds (on average).

4.3 Computational Results of the DRA-NIP

In Table 4 and 5, we present the computational results obtained by solving the DRA-NIP instances. Labels "Decomposition" and "RelaxAppend" in the table denote the decomposition algorithm and the relax-and-append algorithm, respectively. In each row, we report the average value of the results for ten instances. The definitions of labels T, Gap, ZG, 3600+ are the same as presented in the previous section. Note that we define the optimality gap for the relax-and-append algorithm as $(z_{RA}^{ub,L} - z_{RA}^{lb,L})/z_{RA}^{lb,L}$ where L is the last iteration that solves the relaxation of DRA-NIP to optimality. In case the relax-and-append algorithm reaches the time limit before terminating the branch-and-cut algorithm for the first

Table 5: Computational results for solving DRA-NIP US100 instances using decomposition algorithm and relax-and-append algorithm.

Instances			Moment Matching Set					Wasserstein Ambiguity Set				
			Decomposition		RelaxAppend			Decomposition		RelaxAppend		
Category	b	$\|\Omega\|$	T(sec)	**Gap**	T(sec)	**Gap**	**#Iter**	T(sec)	**Gap**	T(sec)	**Gap**	**#Iter**
N40	2	100	0.8	ZG (0)	2.1	ZG (0)	3.5	0.6	ZG (0)	1.9	ZG (0)	3.4
		500	3.7	ZG (0)	8.9	ZG (0)	4.2	4.3	ZG (0)	8.5	ZG (0)	4.0
		1000	6.6	ZG (0)	20.6	ZG (0)	4.7	15.3	ZG (0)	18.0	ZG (0)	4.2
	5	100	10.6	ZG (0)	32.3	ZG (0)	6.1	4.8	ZG (0)	18.0	ZG (0)	4.0
		500	131	ZG (0)	932	0.01% (2)	18.1	43.9	ZG (0)	91.9	ZG (0)	4.2
		1000	229	ZG (0)	1207	0.06% (2)	18.0	144	ZG (0)	163	ZG (0)	4.3
	10	100	382	ZG (0)	1071	0.10% (2)	8.0	400	0.08% (1)	569	0.14% (1)	4.2
		500	1481	0.38% (2)	3240	0.22% (9)	17.1	602	0.30% (1)	1206	0.23% (1)	4.0
		1000	2018	0.63% (3)	3240	0.34% (9)	12.9	1240	0.77% (1)	1626	0.20% (2)	4.3
N60	2	100	1.9	ZG (0)	2.4	ZG (0)	3.1	1.2	ZG (0)	2.5	ZG (0)	3.2
		500	10.4	ZG (0)	9.3	ZG (0)	3.2	7.7	ZG (0)	9.9	ZG (0)	2.9
		1000	23.9	ZG (0)	17.7	ZG (0)	3.2	23.8	ZG (0)	21.3	ZG (0)	3.2
	5	100	31.0	ZG (0)	112	ZG (0)	7.5	12.5	ZG (0)	49.8	ZG (0)	4.3
		500	514	ZG (0)	1639	0.03% (4)	22.7	109	ZG (0)	457	ZG (0)	6.4
		1000	1317	ZG (0)	2139	0.07% (4)	21.7	293	ZG (0)	800	ZG (0)	7.0
	10	100	1147	0.16% (1)	2252	0.09% (5)	6.0	914	0.07% (1)	2035	0.16% (4)	4.4
		500	2779	0.50% (6)	3600+	0.26% (10)	6.3	1781	0.17% (3)	2517	0.16% (6)	3.5
		1000	3600+	0.63% (10)	3600+	0.37% (10)	4.4	2346	0.42% (5)	2817	0.24% (6)	3.1
N80	5	100	165	ZG (0)	327	ZG (0)	6.4	38.3	ZG (0)	255	ZG (0)	5.2
		500	2110	0.29% (4)	2728	0.05% (6)	13.9	284	ZG (0)	1145	0.02% (2)	5.4
		1000	2999	0.45% (6)	2894	0.10% (7)	9.0	864	ZG (0)	1669	0.07% (2)	5.0
	10	100	2534	0.32% (6)	2881	0.22% (7)	2.1	2232	0.24% (5)	2875	0.29% (7)	2.2
		500	3600+	0.82% (10)	3600+	0.47% (10)	1.6	2855	0.46% (6)	3330	0.41% (8)	1.6
		1000	3600+	1.10% (10)	3600+	0.65% (10)	1.3	3512	0.58% (8)	3556	0.45% (9)	1.4
N100	5	100	304	ZG (0)	296	ZG (0)	4.9	68.1	ZG (0)	338	ZG (0)	4.8
		500	2670	0.31% (5)	2288	0.03% (3)	7.9	452	ZG (0)	1914	0.01% (2)	5.7
		1000	3281	0.52% (8)	2726	0.04% (6)	6.0	1154	ZG (0)	2597	0.02% (5)	4.6
	10	100	3600+	0.36% (10)	3600+	0.30% (10)	1.6	2922	0.35% (6)	3600+	0.32% (10)	1.7
		500	3600+	1.04% (10)	3600+	0.55% (10)	1.3	3600+	0.47% (10)	3600+	0.48% (10)	1.2
		1000	3600+	1.22% (10)	3600+	0.73% (10)	1.2	3600+	0.65% (10)	3600+	0.66% (10)	1.0

iteration, we get $z_{RA}^{ub,1}$ from the upper bound of the branch-and-cut algorithm and $z_{RA}^{lb,1}$ by solving the outer distribution separation problem using the best incumbent solution. For the relax-and-append algorithm, we additionally report the average number of iterations until the termination under label "$\#Iter$."

According to Table 4, the decomposition algorithm is 1.2 times (on average) faster than the relax-and-append algorithm in solving the instances to optimality. The decomposition algorithm solved all 600 instances, except one, within 63 seconds (on average), whereas the relax-and-append algorithm could not solve four instances to optimality. Also, the optimality gaps for the unsolved instances are less than 0.1 % for both algorithms. The relax-and-append algorithm requires 1.5 iterations (on average), which implies some instances are solved immediately after the first iteration of the algorithm. We observe that for the US20 instances, the relax-and-append algorithm terminates after the first iteration when the interdictor's optimal decision is to interdict the arcs with no parameter uncertainty, so that any probability distribution leads to the same objective value. Note that the relax-and-append algorithm utilizes the explicit enumeration for the distribution separation instead of solving a linear program as done in the decomposition algorithm. For a small-sized subset

of ambiguity set, this significantly reduces the time required for solving the distribution separation problem, and therefore, the relax-and-append algorithm is likely to be faster than the decomposition algorithm when it terminates within a small number of iterations.

In Table 5, we observe that the relax-and-append algorithm requires 5.7 iterations on average because of the higher level of uncertainty in the US100 instances than the US20 instances. This results in a larger performance difference between the algorithms; the decomposition approach is 2.5 times (on average) faster than the relax-and-append algorithm.

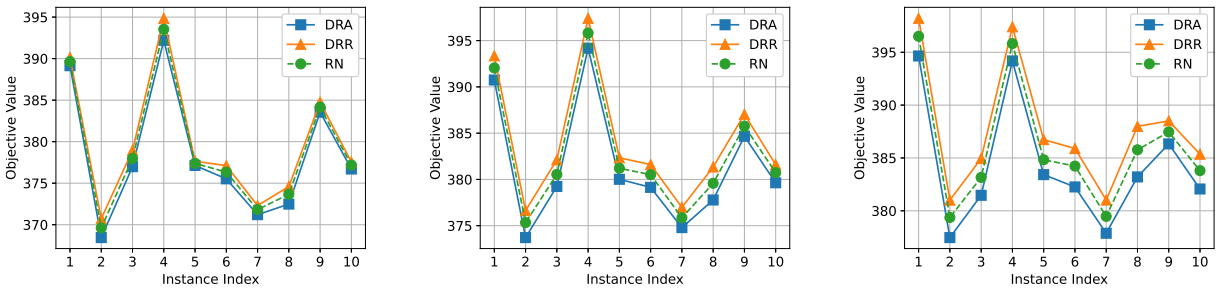
We report, however, the relax-and-append algorithm attains smaller optimality gaps in some instances even though it solves less number of instances to optimality. For example, it reduces the gap to 61% (on average) of the optimality gap obtained by the decomposition algorithm for the instances with the moment matching set that were not solved within the time limit. This shows that the relax-and-append algorithm may be better in providing a feasible solution for larger-sized instances that the decomposition algorithm cannot solve to optimality.

Lastly, we discuss the impact of the choice of ambiguity set on the solution time. For the DRA-NIP with the moment matching set, the size of the distribution separation problem is highly dependent on the number of arcs associated with uncertain parameters since it impacts the number of constraints. This causes the proposed algorithms to spend more time in solving the distribution separation problem for the US100 instances than the US20 instances. To show this, we compute the ratio of the time for solving a distribution separation problem to the total solution time taken for solving each of the US20 and US100 instances using the decomposition algorithm for DRA-NIP. The average of the ratios for the US20 instances is computed to be 9.1%, while the same for the US100 instances is 67.7%. On the other hand, for the Wasserstein ambiguity set, the average of the ratios for the US20 instances and the US100 instances are 20% and 29.2%, respectively. This is because the number of arcs having uncertainty does not impact the number of variables and constraints in the Wasserstein ambiguity set (20); instead, it is largely impacted by the number of scenarios $|\Omega|$. For example, the average of the ratios for US100 instances for $|\Omega| = 100, 500$, and 1000 are 3.6%, 21%, and 36.8%, respectively.

For the DRR-NIP, we also consider the time spent in computing cut coefficients (8), which involves solving the corresponding distribution problem. Similar to the DRA-NIP, we observe that the average of the time taken in solving distribution separation problems and in computing the cut coefficients for the cut-based decomposition algorithm is largely impacted by the number of arcs having uncertainty with the moment-matching set and by the number of scenarios with the Wasserstein ambiguity set.

4.4 Comparison between Risk-Neutral, Distributionally Risk-Averse, and Distributionally Risk-Receptive Solutions

In this section, we evaluate the solutions obtained from the DRA-NIP, DRR-NIP, and (risk-neutral) S-NIP (1) frameworks. For S-NIP instances, we assume that each scenario is realized with equal probability. For test instances, we use the US100 instances so that there is no trivial optimal solution that interdicts the arcs having no uncertainty. Also, we fix the number of scenarios to 100, as we observe in our preliminary tests that the number of



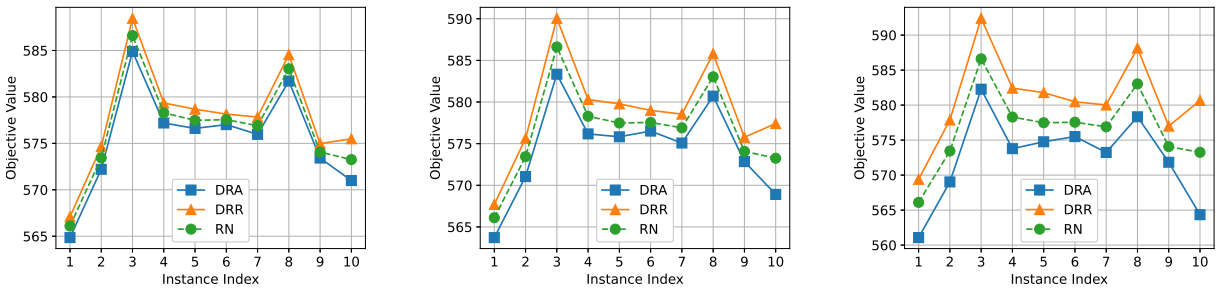
(a) N40, $b = 2$ (b) N40, $b = 5$ (c) N40, $b = 10$

Figure 2: Optimal objective values for instances of N40 (US100) obtained by solving DRA-NIP, DRR-NIP, and RN-NIP with Wasserstein ambiguity set and $b \in \{2, 5, 10\}$.

scenarios has only a minor impact on the difference among distributionally risk-receptive, distributionally risk-averse, and risk-neutral solution values. We plot and compare optimal objective values for different risk preference in Figures 2, 3, and 4.

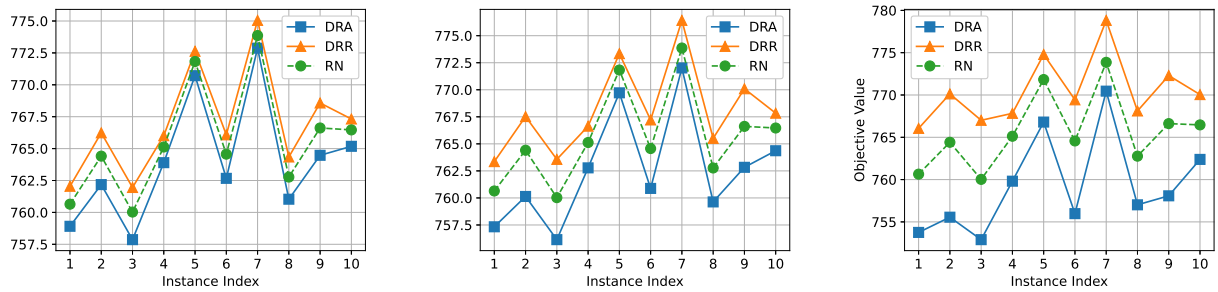
In Figure 2, we present the optimal objective values for ten instances from N40 category with $b = 2$, 5, and 10. Comparing three graphs 2a-2c, we can see that the gap between risk-receptive and risk-averse solution values grows as b increases. Next, in Figures 3 and 4, we present the optimal objective values for N60 instances with the moment matching set and N80 instances with the Wasserstein set, respectively. Given the moments of random variables $u_q, q = 1, ..., m$, the size of the moment matching set increases as ϵ_M increases. Likewise, the Wasserstein ambiguity set's size increases as the Wasserstein distance limit (ϵ_W) increases. In Figures 3 and 4, we can see that the gap between the risk-averse and risk-receptive solution values becomes larger as we increase ϵ_M and the multiplication factor ρ of ϵ_W. These results show that the outcome under distributional uncertainty may vary significantly when the interdictor is capable of interdicting many arcs, and the decision makers are less confident about the ambiguity set (in other words, have a larger ambiguity set). However, in such a case, the DRR-NIP and DRA-NIP frameworks provide optimal solutions for the worst-case and best-case probability distributions, which establish the bounds on the outcome of the S-NIP, respectively, that can help the decision makers in conducting the worst-case and best-case analysis.

From the perspective of the follower, an optimal risk-receptive solution gives information of the most vulnerable arcs, whose failure can significantly increase the minimum cost path. Therefore, the DRR-NIP is an appropriate model for the follower in terms of vulnerability analysis. On the other hand, for the leader, the DRR-NIP and DRA-NIP frameworks provide a range of possible outcomes of interdiction attempts. This range provides the leader with the comparative analysis of the expected outcome according to the risk appetite of the decision maker. Suppose the situation where the leader has relatively small budget or high confidence on the distributional uncertainty like in Figures 2a, 3a and 4a. Then, the gap between risk-averse and risk-receptive objective values will be minor, and the leader can realize that there is no incentive to choose an optimal risk-receptive interdiction strategy in comparison to an optimal risk-averse strategy. Hence, both the DRR-NIP and the DRA-NIP can be useful from the leader's perspective.



(a) N60, $\epsilon_M = 0.05$ (b) N60, $\epsilon_M = 0.1$ (c) N60, $\epsilon_M = 0.25$

Figure 3: Optimal objective values for instances of N60 (US100) obtained by solving DRA-NIP, DRR-NIP, and RN-NIP with moment matching set defined using $\epsilon_M \in \{0.05, 0.1, 0.25\}$.



(a) N60, $\rho = 0.1$ (b) N60, $\rho = 0.2$ (c) N60, $\rho = 0.5$

Figure 4: Optimal objective values for instances of N80 (US100) obtained by solving DRA-NIP, DRR-NIP, and RN-NIP with Wasserstein ambiguity set defined using $\rho \in \{0.1, 0.2, 0.5\}$.

5 Conclusion

In this paper, we introduced the distributionally risk-receptive and distributionally risk-averse network interdiction problems (denoted by DRR-NIP and DRA-NIP, respectively), which are the generalizations of the stochastic network interdiction problem with distributional ambiguity. The DRR-NIP that involves a risk-receptive leader provides the best-case analysis from the leader's perspective, which can also be utilized for the vulnerability analysis of the network by the follower. The DRA-NIP considers a risk-averse leader and provides a robust interdiction solution over distributional ambiguity. For the DRR-NIP, we proposed two exact approaches: the first approach is based on the reformulation technique that linearizes the bilinear terms in the objective function, and the second approach is based on the valid cut utilizing the binary property of interdiction variables and the distribution separation procedure. We also proposed an approximation algorithm that uses the restricted feasible set of probability distributions. For the DRA-NIP, we proposed a decomposition approach and a relax-and-append algorithm to exactly solve it. We presented convergence analysis for these approaches along with the results of extensive computational tests to demonstrate the effectiveness and efficiency of these approaches. We also analyzed the factors that impact the difference between distributionally risk-receptive and risk-averse optimal solution values

and discussed the significance of the DRR-NIP and the DRA-NIP frameworks for both the leader and the follower with varying levels of risk appetite.

ACKNOWLEDGMENTS

This research is funded by Automotive Research Center (ARC) of University of Michigan, Ann Arbor in accordance with Cooperative Agreement W56HZV-19-2-0001 U.S. Army CCDC Ground Vehicle Systems Center (GVSC) Warren, MI.

References

[1] Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical Programming*, 58(1):295–324, 1993.

[2] Manish Bansal, Kuo-Ling Huang, and Sanjay Mehrotra. Decomposition algorithms for two-stage distributionally robust mixed binary programs. *SIAM Journal on Optimization*, 28(3):2360–2383, 2018.

[3] N. Orkun Baycik and Kelly M. Sullivan. Robust location of hidden interdictions on a shortest path network. *IISE Transactions*, 51(12):1332–1347, 2019.

[4] Halil Bayrak and Matthew D. Bailey. Shortest path network interdiction with asymmetric information. *Networks*, 52(3):133–140, 2008.

[5] Mokhtar S Bazaraa, John J Jarvis, and Hanis D Sherali. *Linear programming and network flows*. John Wiley & Sons, Hoboken, NJ, 2008.

[6] Michael G.H. Bell, U. Kanturska, J.-D. Schmöcker, and Achille Fonzone. Attacker–defender models and road network vulnerability. *Philosophical Transactions of the Royal Society A*, 366(1872):1893–1906, 2008.

[7] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[8] Gerald Brown, Matthew Carlyle, Javier Salmerón, and Kevin Wood. Defending critical infrastructure. *Interfaces*, 36(6):530–544, 2006.

[9] Kelly J. Cormican, David P. Morton, and R. Kevin Wood. Stochastic network interdiction. *Operations Research*, 46(2):184–197, 1998.

[10] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

[11] Federico Della Croce and Rosario Scatamacchia. An exact approach for the bilevel knapsack problem with interdiction constraints and extensions. *Mathematical Programming*, 183(1):249–281, 2020.

[12] Jitka Dupačová. The minimax approach to stochastic programming and an illustrative application. *Stochastics*, 20(1):73–88, 1987.

[13] D. R. Fulkerson and Gary C. Harding. Maximizing the minimum source-sink path subject to a budget constraint. *Mathematical Programming*, 13(1):116–118, 1977.

[14] Fabio Furini, Ivana Ljubić, Sébastien Martin, and Pablo San Segundo. The maximum clique interdiction problem. *European Journal of Operational Research*, 277(1):112–127, 2019.

[15] Bruce Golden. A problem in network interdiction. *Naval Research Logistics Quarterly*, 25(4):711–713, 1978.

[16] Harald Held, Raymond Hemmecke, and David L Woodruff. A decomposition algorithm applied to planning the interdiction of stochastic networks. *Naval Research Logistics*, 52(4):321–328, 2005.

[17] Raymond Hemmecke, Rüdiger Schultz, and David L. Woodruff. Interdicting stochastic networks with binary interdiction effort. In David L. Woodruff, editor, *Network interdiction and stochastic integer programming*, pages 69–84. Kluwer, Boston, MA, 2003.

[18] Le Thi Khanh Hien, Melvyn Sim, and Huan Xu. Mitigating interdiction risk with fortification. *Operations Research*, 68(2):348–362, 2020.

[19] Tim Holzmann and J. Cole Smith. The shortest path interdiction problem with randomized interdiction strategies: Complexity and algorithms. *Operations Research*, 69(1):82–99, 2020.

[20] Eitan Israeli. *System Interdiction and Defense*. PhD thesis, Naval Postgraduate School, Monterey, CA, 1999.

[21] Eitan Israeli and R. Kevin Wood. Shortest-path network interdiction. *Networks*, 40(2):97–111, 2002.

[22] Udom Janjarassuk and Jeff Linderoth. Reformulation and sampling to solve a stochastic network interdiction problem. *Networks*, 52(3):120–132, 2008.

[23] Gilbert Laporte and François V. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142, 1993.

[24] Xiao Lei, Siqian Shen, and Yongjia Song. Stochastic maximum flow interdiction problems under heterogeneous risk preferences. *Computers & Operations Research*, 90:97–109, 2018.

[25] David P Morton, Feng Pan, and Kevin J Saeger. Models for nuclear smuggling interdiction. *IIE Transactions*, 39(1):3–14, 2007.

[26] Di H. Nguyen and J. Cole Smith. Asymmetric stochastic shortest-path interdiction under conditional value-at-risk. *IISE Transactions*, pages 1–32, 2022.

[27] Di H. Nguyen and J. Cole Smith. Network interdiction with asymmetric cost uncertainty. *European Journal of Operational Research*, 297(1):239–251, 2022.

[28] Feng Pan and David P. Morton. Minimizing a stochastic maximum-reliability path. *Networks*, 52(3):111–119, 2008.

[29] Babak Saleck Pay, Jason R. W. Merrick, and Yongjia Song. Stochastic network interdiction with incomplete preference. *Networks*, 73(1):3–22, 2019.

[30] Georg Pflug and David Wozabal. Ambiguity in portfolio selection. *Quantitative Finance*, 7(4):435–442, 2007.

[31] Hamed Rahimian, Güzin Bayraksan, and Tito Homem-de Mello. Identifying effective scenarios in distributionally robust stochastic programs with total variation distance. *Mathematical Programming*, 173(1):393–430, 2019.

[32] Utsav Sadana and Erick Delage. The value of randomized strategies in distributionally robust risk averse network interdiction games. *arXiv preprint arXiv:2003.07915*, 2020.

[33] Javier Salmerón. Deception tactics for network interdiction: A multiobjective approach. *Networks*, 60(1):45–58, 2012.

[34] Javier Salmerón, Kevin Wood, and Ross Baldick. Worst-case interdiction analysis of large-scale electric power grids. *IEEE Transactions on Power Systems*, 24(1):96–104, 2009.

[35] Maria P Scaparra and Richard L Church. A bilevel mixed-integer program for critical infrastructure protection planning. *Computers & Operations Research*, 35(6):1905–1923, 2008.

[36] Herbert Scarf. A min-max solution of an inventory problem. In Kenneth Joseph Arrow, Samuel Karlin, and Herbert Scarf, editors, *Studies in the Mathematical Theory of Inventory and Production*, pages 201–209. Stanford University Press, Stanford, CA, 1958.

[37] Yongjia Song and Siqian Shen. Risk-averse shortest path interdiction. *INFORMS Journal on Computing*, 28(3):527–539, 2016.

[38] Richard M Van Slyke and Roger Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969.

[39] Alan Washburn and Kevin Wood. Two-person zero-sum games for network interdiction. *Operations Research*, 43(2):243–251, 1995.

[40] Richard Wollmer. Removing arcs from a network. *Operations Research*, 12(6):934–940, 1964.

[41] R. Kevin Wood. Deterministic network interdiction. *Mathematical and Computer Modelling*, 17(2):1–18, 1993.

[42] Jing Yang, Juan S. Borrero, Oleg A. Prokopyev, and Denis Sauré. Sequential shortest path interdiction with incomplete information and limited feedback. *Decision Analysis*, 18(3):218–244, 2021.

Appendix A L-Shaped Method with branch-and-cut for DRR-NIP

For the sake of completeness of this paper, we also provide a discussion on our implementation of the L-shaped method with a branch-and-cut framework to solve reformulation (6) of the DRR-NIP, which is equivalent to

$$\max \quad \theta, \tag{21a}$$

$$\text{s.t.} \quad \theta \leq \sum_{\omega\in\Omega}\sum_{a\in A}(p_\omega c_a + d_a^\omega \xi_a^\omega \eta_a^\omega)\hat{y}_a^\omega, \quad \text{for all } \hat{\mathbf{y}}^\omega \in \hat{Y}_E^\omega,\ \omega\in\Omega, \tag{21b}$$

$$\eta_a^\omega \leq x_a, \quad \text{for all } \omega\in\Omega, a\in A, \tag{21c}$$

$$\eta_a^\omega \leq p_\omega, \quad \text{for all } \omega\in\Omega, a\in A, \tag{21d}$$

$$\eta_a^\omega \geq 0, \quad \text{for all } \omega\in\Omega, a\in A, \tag{21e}$$

$$\mathbf{x}\in X, \{p_\omega\}_{\omega\in\Omega}\in\mathcal{P}, \theta\in\mathbb{R}. \tag{21f}$$

where $\hat{Y}_E^\omega$ is the set of all extreme points of Y^ω for $\omega\in\Omega$. At the root node of the branch-and-cut tree, we construct an upper bound approximation by relaxing constraints (21b), which is referred to as current problem. For the boundedness of the root-node problem (in particular, θ), we randomly select a feasible interdiction solution $\bar{\mathbf{x}}_0\in X$, solve subproblems (4b) with $\mathbf{x}=\bar{\mathbf{x}}_0$ to obtain $\bar{\mathbf{y}}_0^\omega\in\hat{Y}_E^\omega$ for all $\omega\in\Omega$, and derive an optimality cut of the form (21b).

Similar to a generic branch-and-cut approach, at each node in the branch-and-cut tree, the algorithm solves the LP relaxation of the current problem (node-LP) with fixed values for a subset of the $\mathbf{x}$ variables, i.e., $x_i=1$ for $i\in S_1$ and $x_i=0$ for $i\in S_2$ where S_1, $S_2\subseteq\{1,\ldots,|A|\}$ and $S_1\cap S_2=\emptyset$. Denote $(\bar{\mathbf{x}},\bar{\theta},\bar{\eta})$ as the solution to the node-LP, where $\bar{\eta}=\{\bar{\eta}^\omega\}_{\omega\in\Omega,a\in A}$ and $\bar{\theta}$ gives a node upper bound on the optimal objective value of (21). Among all leaf nodes in the current tree, we collect the node upper bounds and take the maximum as the best upper bound (UB). In case $\bar{\mathbf{x}}$ is integral, the algorithm computes $R_\omega(\bar{p}_\omega,\bar{\eta}^\omega)$ and associated optimal solution $\bar{\mathbf{y}}_{node}^\omega\in\hat{Y}_E^\omega$ for all $\omega\in\Omega$, where $\{\bar{p}_\omega\}$ is obtained by solving $\max_{P\in\mathcal{P}}\sum_{\omega\in\Omega}p_\omega Q(\bar{\mathbf{x}})$. If $\bar{\theta}>\sum_{\omega\in\Omega}R_\omega(\bar{p}_\omega,\bar{\eta}^\omega)$, it uses $\bar{\mathbf{y}}_{node}^\omega$ to derive an optimality cut of the form (21b), adds the cut to the current problem (and node-LP), and resolve the node-LP. The foregoing step is repeated until there is no violated optimality cut. Again, in case the latest node-LP solution has integral $\bar{\mathbf{x}}$, the algorithm updates the best-known lower bound, $LB=\max\{LB,\sum_{\omega\in\Omega}R_\omega(\bar{p}_\omega,\bar{\eta}^\omega)\}$, and check if the optimality gap, $(UB-LB)/LB$, is less than a given tolerance level. If the current node-LP solution is not integral, the algorithm proceeds with a conventional branch-and-cut procedure, such as adding cutting planes (and resolving the node-LP), branching, and adding two nodes to the branch-and-bound tree.