Distributionally Ambiguous Multistage Stochastic Integer and Disjunctive Programs: Applications to Sequential Two-player Interdiction Games*

Sumin Kang† Manish Bansal†

June 18, 2023

Abstract

This paper studies the generalizations of multistage stochastic mixed-integer programs (MSIPs) with distributional ambiguity, namely distributionally risk-receptive and risk-averse multistage stochastic mixed-integer programs (denoted by DRR- and DRA-MSIPs). These modeling frameworks have applications in non-cooperative Stackelberg games involving two players, namely a leader and a follower, with uncertainty in the impact of the decisions made by the leader. We present cutting plane-based and reformulation-based approaches for solving DRR- and DRA-MSIPs to optimality. We showcase that these approaches are finitely convergent with probability one. We also introduce generalizations of MSIPs by considering multistage stochastic disjunctive programs with(out) distributional ambiguity and present algorithms for solving them. To assess the performance of the algorithms for MSIPs, we consider instances of multistage maximum flow and facility location interdiction problems that are important interdiction problems in their own right, and only their single-stage variants have been studied in the literature. Based on our computational results, we observe that the cutting plane-based approaches are 26.1 times (on average) faster than the reformulation-based approaches for the foregoing instances.

Keywords. distributionally robust optimization, multistage stochastic integer programs, multistage stochastic disjunctive programs, stochastic dual dynamic programming, distributionally risk-receptive optimization

1 Introduction

Multistage stochastic programming is a modeling framework for making a sequence of decisions over multiple stages while addressing the uncertainty of input data parameters. In a multistage stochastic program (MSP), the uncertain parameters are modeled as a stochastic process, and decisions are made at each stage based on the information observed up to the time these decisions are being made. The objective at each stage of MSP is to optimize the sum of the cost associated with the decisions of the current stage and the expected cost of the future stage. Each stage has *state* decision variables that directly affect the decisions of the next stage, and *local* decision variables belonging to the stage only. (Refer to Chapter 6 of [7] for more details about MSPs.) In stochastic programming, it is assumed that the probability distribution associated with uncertain data is

*Funding information: This work was funded by National Science Foundation, Grant Number: CMMI–1824897; U.S. Department of Defense, Grant Number: W56HZV-19-2-0001.

†Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA 24060 (suminkang@vt.edu, bansal@vt.edu).

known. However, in many real-world situations, the probability distribution is often unknown or challenging to estimate. Such an ambiguity in the probability distribution has been tackled by min-max stochastic programming [16, 38], which is lately known as distributionally robust optimization (DRO).

A DRO model considers a set of possible probability distributions, referred to as an *ambiguity* set, and evaluates an objective defined as the expected cost for a worst-case probability distribution within the ambiguity set. An optimal solution provided by this model attains the robustness with respect to the distributions, i.e., reflects the risk-aversion of the decision-maker. In this sense, we refer to a DRO problem as a *distributionally risk-averse* optimization problem throughout this paper (readers can refer to Chapters 6 and 7 of Shapiro et al. [40] for more discussions about the equivalence between a DRO problem and a risk-averse optimization problem with a coherent risk measure). Alternatively, one can consider an optimistic decision-maker under distributional ambiguity whose objective is to optimize the expected cost for a best-case probability distribution within the ambiguity set. We characterize the risk-appetite of such a decision-maker as *risk-receptiveness*, opposed to the risk-aversion. Accordingly, we refer to an optimization problem under the distributional ambiguity with a risk-receptive decision-maker as a *distributionally risk-receptive* optimization problem [24].

The concepts of distributional risk-aversion and risk-receptiveness have practical applications in game theoretic models involving two non-cooperative players, referred to as leader and follower, with uncertainty in the impact of the decisions made by the leader to the follower's objective, feasible region, or both. One class of problems that fall under this category is network interdiction problem (NIP), which involves a game played between two non-cooperative decision-makers—an interdictor (or attacker) and a network user (or defender). In NIP, the interdictor aims to maximize the disruption of the network, while the network user aims to optimize the network performance given the disruptions caused by the interdictor. Since NIP involves two decision-makers with conflicting objectives, it is important to address perspectives of both players. Specifically, from the network user's perspective who is interested in the vulnerability analysis of critical infrastructure, the distributionally risk-receptive NIP model can be used to demonstrate the worst disruption performance of the network, reflecting the interdictor's optimistic view towards the distributional ambiguity. Conversely, from the interdictor's perspective who intends to disrupt the network, the distributionally risk-averse NIP model can be used to obtain a robust plan for deploying interdiction resources to negatively impact the network user's objective (e.g., a robust plan of disrupting network components to impede traffickers [25, 29]).

In this paper, we introduce generalizations of MSPs to address distributional ambiguity, i.e., distributionally risk-averse multistage stochastic programs (DRA-MSPs) and distributionally risk-receptive multistage stochastic programs (DRR-MSPs). Our models consider binary state variables and the set of feasible solutions for each stage is defined by either a mixed-integer set or a disjunctive set that depends on state decisions made in the previous stage. We refer to these models as a distributionally risk-averse multistage stochastic mixed-integer or disjunctive linear program (denoted by DRA-MSIP or DRA-MSDP) and a distributionally risk-receptive multistage stochastic mixed-integer or disjunctive linear program (denoted by DRR-MSIP or DRR-MSDP). It is worth noting that if the ambiguity sets are singleton, then DRA- and DRR-MSIPs reduce to a (risk-neutral) MSIP.

For solving multistage stochastic linear programs (MSLPs), nested Benders decomposition (NBD), proposed by Birge [6], is one of the earliest work, which under-approximates the cost-to-go function by adding Benders cuts in a iterative manner. However, this approach could become computationally impractical for a large scenario tree. To address this issue, SDDP (stochastic dual dynamic programming) is developed by Pereira and Pinto [35] as a special case of NBD with stochastic sampling of scenario paths for MSLPs under the stage-wise independence assumption, i.e., random parameters at each stage are independent of the random process up to the previous stage. For MSIPs with the foregoing assumption, Zou et al. [46] present stochastic dual dynamic integer programming (SDDiP) embedded with Lagrangian cut. (Refer to Section 2 for an extensive literature review.) To solve DRA- and DRR-MSIPs, we present reformulation-based and cutting plane-based solution approaches that extend SDDP. We further generalize these algorithms for solving DRA- and DRR-MSDPs and present its application for solving DRA and DRR-MSIPs as well. We also introduce multistage maximum flow and facility location interdiction problems, formulate them as DRA- and DRR-MSIPs, and utilize instances of these problems to evaluate the effectiveness and efficiency of the proposed approaches.

In the remaining of this section, we present the generic formulations of DRA- and DRR-MSPs, motivating applications to aforementioned multistage two-player interdiction games, and a summary of our contributions along with the organization of this paper. Throughout the paper, we use $[d]$ to denote set $\{1, \dots, d\}$ for any positive integer d.

1.1 Problem Formulation: DRA-MSP and DRR-MSP

We first present the Bellman equation of an MSP with a planning horizon of T stages and then generalize it to the cases with distributional ambiguity, i.e., DRA- and DRR-MSPs. Assuming that the state decision vector and the stage decision vector for stage $t \in [T] := \{1, \dots, T\}$ are denoted by x_t and y_t, respectively, an (risk-neutral) MSP is formulated as follows:

$$\min_{(x_1,y_1)\in X_1} \Big\{ f_1(x_1, y_1) + \mathbb{E}_{P_2}\big[Q_2(x_1, \boldsymbol{\omega}_2)\big] \Big\}, \tag{1}$$

where $\boldsymbol{\omega}_t$ is a random vector that represents uncertain parameters for stage $t = 2, \dots, T$,

$$Q_t(x_{t-1}, \boldsymbol{\omega}_t) = \min_{(x_t,y_t)\in X_t(x_{t-1},\boldsymbol{\omega}_t)} \Big\{ f_t(x_t, y_t, \boldsymbol{\omega}_t) + \mathbb{E}_{P_{t+1}}\big[Q_{t+1}(x_t, \boldsymbol{\omega}_{t+1})\big] \Big\}, \tag{2}$$

for $t = 2, \dots, T$ and $Q_{T+1} = 0$. Set X_1 denotes the feasible region of the first stage, and for $t \in \{2, \dots, T\}$, set $X_t(x_{t-1}, \boldsymbol{\omega}_t)$ denotes the feasible region of each stage t that depends on a decision of the previous stage x_{t-1} and a realization of the random vector $\boldsymbol{\omega}_t$ belonging to sample space Ω_t. In MSPs, it is assumed that the probability distribution P_t associated with $\boldsymbol{\omega}_t$ is known.

Now, to present DRA- and DRR-MSPs, we consider a set of probability distributions $\mathcal{P}_t$ as the ambiguity set. Then, the bellman equation form of a DRA-MSP is given by

$$\min_{(x_1,y_1)\in X_1} \Big\{ f_1(x_1, y_1) + \max_{P_2\in\mathcal{P}_2} \mathbb{E}_{P_2}\big[Q_2^{RA}(x_1, \boldsymbol{\omega}_2)\big] \Big\}, \tag{3}$$

where

$$Q_t^{RA}(x_{t-1}, \boldsymbol{\omega}_t) = \min_{(x_t,y_t)\in X_t(x_{t-1},\boldsymbol{\omega}_t)} \Big\{ f_t(x_t, y_t, \boldsymbol{\omega}_t) + \max_{P_{t+1}\in\mathcal{P}_{t+1}} \mathbb{E}_{P_{t+1}}\big[Q_{t+1}^{RA}(x_t, \boldsymbol{\omega}_{t+1})\big] \Big\} \tag{4}$$

for $t = 2, \dots, T$, and $Q_{T+1}^{RA} = 0$. By minimizing the probability distribution over the ambiguity set in (3) and (4), instead of maximizing, we can formulate a DRR-MSP as follows:

$$\min_{(x_1,y_1)\in X_1} \Big\{ f_1(x_1, y_1) + \min_{P_2\in\mathcal{P}_2} \mathbb{E}_{P_2}\big[Q_2^{RR}(x_1, \boldsymbol{\omega}_2)\big] \Big\}, \tag{5}$$

where

$$Q_t^{RR}(x_{t-1}, \boldsymbol{\omega}_t) = \min_{(x_t, y_t) \in X_t(x_{t-1}, \boldsymbol{\omega}_t)} \left\{ f_t(x_t, y_t, \boldsymbol{\omega}_t) + \min_{P_{t+1} \in \mathcal{P}_{t+1}} \mathbb{E}_{P_{t+1}} \left[Q_{t+1}^{RR}(x_t, \boldsymbol{\omega}_{t+1}) \right] \right\} \tag{6}$$

for $t = 2, \ldots, T$, and $Q_{T+1}^{RR} = 0$.

When the feasible region $X_t(x_{t-1}, \boldsymbol{\omega}_t)$ is defined by a set of linear inequalities and disjunctive constraints, problems (3)-(4) and (5)-(6) are referred to as DRA- and DRR-MSDPs, respectively. Similarly, the feasible regions defined by mixed-integer sets lead to DRA- and DRR-MSIPs. Particularly, in DRA- and DRR-MSDPs, the feasible regions for $t \in [T]$ are defined as follows:

$$\begin{aligned} X_t(x_{t-1}, \boldsymbol{\omega}_t) := \Big\{ &(x_t, y_t) \in \mathbb{R}_+^{d_x} \times \mathbb{R}_+^{d_y} : \\ &\bigvee_{h \in H_t} \Big(A_t^h(\boldsymbol{\omega}_t) x_t + B_t^h(\boldsymbol{\omega}_t) y_t \geq b_t^h(\boldsymbol{\omega}_t) - C_t^h(\boldsymbol{\omega}_t) x_{t-1} \Big) \Big\}. \end{aligned} \tag{7}$$

Here, notation $\vee$ is used to denote disjunction ("or" logical operator). The disjunctive constraints generalize integrality constraints on variables. For example, a binary restriction on a variable, i.e., $x \in \{0, 1\}$, is equivalent to a disjunction: $(x = 0) \vee (x = 1)$. If the disjunctive constraints in (7) represent linear inequalities with integrality constraints on variables, then the DRA- and DRR-MSDPs reduce to DRA- and DRR-MSIPs. The main challenges encountered in solving these problems arises from the nonconvexity of feasible regions, caused by logical disjunctions or integer variables, as well as the nonlinearity and discontinuity of objective functions. Additionally, in the DRR problems, each stage problem's objective function is nonconvex even if all variables are continuous. It is worth noting that DRA- and DRR-MSPs are at least as hard as their special case, (risk-neutral) MSPs, which are in PSPACE (and conjectured in PSPACE-hard [18]).

1.2 Motivating Applications: Two-player Interdiction Problems

In this section, we formally present applications of DRR- and DRA-MSIPs in formulating two interdiction problems, namely, the maximum flow interdiction problem (MFIP) and the facility location interdiction problem (FLIP). The objective values of DRR- and DRA-MSPs provide a "confidence interval at distributional ambiguity" that showcases a range of the expected objective value, bounded by the optimistic and pessimistic estimates. For example, the vulnerability of a network can be evaluated by solving DRR and DRA variants of MFIP, computing the confidence interval of the network's performance, and comparing a specified performance target with the interval to determine whether it is likely to be achieved or not in the presence of disruptions.

1.2.1 Multistage maximum flow interdiction problem

In deterministic setting, Wollmer [42] and Wood [43] study single-stage MFIP, and Malaviya et al. [29] and Kosmas et al. [25] study multistage MFIP. Cormican et al. [13] and Janjarassuk and Linderoth [23] study stochastic (risk-neutral) variants of single-stage MFIP, and its DRA variant has been studied by Sadana and Delage [37]. In this paper, we introduce multistage stochastic MFIP (MS-MFIP) and its DRR and DRA models that incorporate varying risk-appetite of the interdictor.

Multistage Stochastic MFIP. Consider a directed and capacitated network, denoted by $G = (N, A)$ where N is a set of nodes and A is a set of directed arcs of the network. The interdictor's objective is to minimize the total flow from the source node s to the sink node r of the network G

by interdicting a subset of arcs in A. In contrast, the network user's objective is to maximize the total flow given the interdicted network. For each stage, both the interdictor and the network user make their decisions as follows: The interdictor removes a set of arcs from the network G given an interdiction budget, and the network user finds a maximum flow after observing the interdictor's decision. It is assumed that after an arc is interdicted, the network user cannot use it to the end of the time horizon.

Let x_t and y_t be the interdiction decision vector and the flow decision vector, respectively, for stage $t \in [T]$. An interdiction decision $x_{t,a}$, for each arc $a \in A$, is binary, i.e., $x_{t,a} = 1$, if an interdiction occurs on $a \in A$, and $x_{t,a} = 0$, otherwise. Each arc $a \in A$ is associated with the interdiction cost, denoted by $f_{t,a}$. The interdiction budget is denoted by b_t, for each stage $t \in [T]$. We assume that the capacity of arc $a \in A$ is uncertain and denoted by $c_{t,a}(\boldsymbol{\omega}_t)$. We denote the set of the outgoing arcs and the set of the incoming arcs of node $n \in N$ by $\delta^+(n)$ and $\delta^-(n)$, respectively. For the brevity, we assume there exists a dummy arc from r to s in A associated with infinite capacity and interdiction cost. Then, the bellman equation form of MS-MFIP is given by (1) and (2) where

$$\begin{aligned} Q_t(x_{t-1}, \boldsymbol{\omega}_t) := \min\ & \psi_t(x_t, \boldsymbol{\omega}_t) + \mathbb{E}_{P_{t+1}}\left[Q_{t+1}(x_t, \boldsymbol{\omega}_{t+1})\right] && \text{(8a)} \\ \text{s.t.}\ & \sum_{a \in A} f_{t,a} x_{t,a} \le b_t + \sum_{a \in A} f_{t,a} x_{t-1,a} && \text{(8b)} \\ & x_t \ge x_{t-1} && \text{(8c)} \\ & x_t \in \{0,1\}^{|A|}, && \text{(8d)} \end{aligned}$$

and

$$\begin{aligned} \psi_t(x_t, \boldsymbol{\omega}_t) := \max\ & y_{t,(r,s)} && \text{(9a)} \\ \text{s.t.}\ & \sum_{a \in \delta^+(n)} y_{t,a} - \sum_{a \in \delta^-(n)} y_{t,a} = 0, \quad \forall n \in N && \text{(9b)} \\ & y_{t,a} \le c_{t,a}(\boldsymbol{\omega}_t)(1 - x_{t,a}), \quad \forall a \in A && \text{(9c)} \\ & y_t \in \mathbb{R}_+^{|A|}, && \text{(9d)} \end{aligned}$$

for each $t \in [T]$. The minimization problem (8) is the interdictor's problem in which Constraint (8b) restricts the total interdiction cost within the given budget b_t. Constraint (8c) ensures that once the interdiction occurs on an arc, then its impact remains till the end of the time horizon. Given an interdiction solution x_t, the objective function of the interdictor's problem at stage $t \in [T]$, $\psi_t(x_t, \boldsymbol{\omega}_t)$, is a value function that provides a maximum flow over the interdicted network. The function $\psi_t(x_t, \boldsymbol{\omega}_t)$ is computed by solving the network user's problem (9) where decision variable $y_{t,a}$ represents a flow on arc a in A, Constraints (9b) enforce the flow balance on nodes in N, and Constraints (9c) restrict the capacity of arcs in A. Notice that because of Constraints (9c), $y_{t,a}$ is restricted to be zero if the interdiction occurs on arc $a \in A$, i.e., if $x_{t,a} = 1$.

1.2.2 Multistage facility location interdiction problem

Single-stage (deterministic) FLIP, also referred to as the r-interdiction median problem, is introduced by Church et al. [12]. The objective of the single-stage FLIP is to find a subset of r facilities that when removed, maximizes the network user's objective of minimizing total weighted distance. The FLIP we present in this section is multistage stochastic FLIP, denoted by MS-FLIP, which addresses uncertainty in the demand. In each stage of MS-FLIP, an interdiction decision is first made to remove the interdicted facilities from the network, and then each demand point is assigned to the closest facility to fulfill its demand value. We study its DRA and DRR models as well, which have not been studied in the literature.

Multistage Stochastic FLIP. Let L and M be the number of demand points and facilities, respectively. At stage $t \in [T]$, let $x_{tm} \in \{0,1\}$ be an interdiction decision variable, which equals 1, if the interdiction occurs on facility $m \in [M]$, or equals 0, otherwise. Variable $y_{tlm} \in \{0,1\}$ denotes an assignment decision that represents whether demand point l is assigned to facility m. We denote the random weighted distance between l and m by $c_{tlm}(\boldsymbol{\omega}_t) = a_{tl}(\boldsymbol{\omega}_t) d_{lm}$, where $d_{lm} > 0$ is the Euclidean distance between l and m, and $a_{tl}(\boldsymbol{\omega}_t)$ is an uncertain demand at point $l \in [L]$. The bellman equation of MS-FLIP is given by (1) and (2) with

$$Q_t(x_{t-1}, \boldsymbol{\omega}_t) := \max \sum_{l \in [L], m \in [M]} c_{tlm}(\boldsymbol{\omega}_t) y_{tlm} + \mathbb{E}_{P_{t+1}}\left[Q_{t+1}(x_t, \boldsymbol{\omega}_{t+1})\right] \tag{10a}$$

$$\text{s.t.} \sum_{m \in [M]} y_{tlm} = 1, \quad \forall l \in [L], \tag{10b}$$

$$x_t \geq x_{t-1}, \tag{10c}$$

$$\sum_{m \in [M]} (x_{tm} - x_{t-1,m}) = r_t, \tag{10d}$$

$$\sum_{n \in S_{lm}} y_{tln} \leq x_{tm}, \quad \forall l \in [L], m \in [M], \tag{10e}$$

$$x_t \in \{0,1\}^M, \; y_t \in \{0,1\}^{L \times M}. \tag{10f}$$

Here $S_{lm} := \{n \in [M] : d_{ln} > d_{lm}\}$ is the set of facilities that are farther than facility $m \in [M]$ is from demand point $l \in [L]$. The first term of the objective function (10a) represents the total weighted distance of the assignment of demand points to non-interdicted facilities. Constraints (10b) enforce each demand point to be assigned to a facility. Constraint (10c) ensures that the facilities interdicted from the previous stages remain interdicted for the current stage. As a consequence, Constraint (10d) ensures that the total number of interdictions occurred at the current stage equals to the budget r_t. Constraint (10e), for each $l \in [L]$ and $m \in [M]$, prevents the demand point l from being assigned to facilities farther than the facility m, unless the facility m is interdicted. It should be noted that problem (10) is a single maximization problem, but not a max-min problem, because of the assumption that each facility has enough capacity to cover all demand values and the demand point is always assigned to the closest facility through Constraints (10e).

Remark 1. In [24], the authors consider DRA and DRR variants of the shortest path interdiction problem. In the shortest path interdiction problem, the interdictor's objective is to maximize the traveling cost of the network user, whose objective is to find a shortest path on the interdicted network. It is important to note that their problems consider only a single stage for each player, i.e., $T = 2$, and the network user's problem is a linear program, i.e., $Q_2(x_1, \cdot)$ is piecewise linear and convex, which are special cases of the frameworks presented in this paper.

1.3 Contribution and Organization

In this paper, we present exact and finitely convergent algorithms for solving DRR-MSIPs, DRA-MSIPs, DRR-MSDPs, and DRA-MSDPs. We provide computational results for DRR- and DRA-MSIPs using instances of MS-MFIP and MS-FLIP. More specifically, the contributions of this paper are as follows.

- *DRR-MSIPs.* We present cutting plane-based and reformulation-based algorithms for DRR-MSIPs, referred to as DRR-SDDP-C and DRR-SDDP-R algorithms, respectively. For the DRR-SDDP-C algorithm, we derive a new class of valid inequalities to get lower-bound approximations in each iteration, and as a consequence, this algorithm is 24.1 times faster (on average) than the DRR-SDDP-R algorithm. As per our knowledge, DRR-MSIPs have not been studied in the literature even for $T = 2$.

- *DRA-MSIPs.* For DRA-MSIPs with Wasserstein ambiguity sets, we extend the approach proposed in [17] for DRA-MSLPs that is based on the strong duality-based reformulation (Theorem 1 in [21]). We denote this method by DRA-SDDP-R algorithm. Additionally, we present another separation-based algorithm for DRA-MSIPs, denoted by DRA-SDDP-C algorithm. Based on our computational experiments, we observe that the DRA-SDDP-C algorithm is 28 times faster (on average) than the DRA-SDDP-R algorithm.

- *DRR-MSDPs and DRA-MSDPs.* We introduce the first algorithm for solving MSDPs, DRR-MSDPs, and DRA-MSDPs by deriving tight extended formulations for parametric disjunctive constraints in each stage. Since MSIPs are special cases of MSDPs, we utilize the foregoing approach for solving MSIPs with(out) distributional ambiguity as well, where a hierarchy of relaxations of the feasible regions is obtained in each iteration.

- *MS-MFIP and MS-FLIP.* As mentioned in Section 1.2, MS-MFIP and MS-FLIP are important interdiction problems in their own right and have not been studied in the literature. This paper presents solution approaches for solving these problems and their distributionally ambiguous variants, thereby generalizing results of [12, 13, 23] that study single-stage version of these problems.

- By conducting out-of-sample tests, we demonstrate the significance of distributional risk-aversion and risk-receptiveness in the context of MS-MFIP. The results show that the distributional risk-aversion enhances the robustness of policies under uncertainty, and the distributional risk-receptiveness enables the identification of network vulnerabilities in situations where realizations of the uncertainty are unfavorable from the network user's perspective.

The rest of the paper is organized as follows: Section 2 provides a comprehensive review of previous studies on MSPs (with and without distributional ambiguity). In Section 3, we present a brief overview of SDDP and introduce its extension to address distributional ambiguity. We then proceed to propose solution algorithms for DRR- and DRA-MSIPs in Sections 4 and 5, respectively, along with the convergence analysis showing that they are finitely convergent. Subsequently, we further extend our approaches to handle DRR- and DRA-MSDPs in Section 6 and discuss application of these approaches for solving their special cases with integer variables by deriving a hierarchy of relaxations for the feasible set of each stage. In Section 7, we present our computational results, followed by the conclusions in Section 8.

Throughout the paper, we made the following assumptions:

Assumption 1. The random vectors are stage-wise independent; that is, $\boldsymbol{\omega}_t$ is independent of $\boldsymbol{\omega}_{[t-1]} = (\boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_{t-1})$ for all $t = 3, \ldots, T$.

Assumption 2. The supports Ω_t, for all $t = 2, \ldots, T$, are finite, i.e., $|\Omega_t| < \infty$. Accordingly, let $p_t^i = p_t(\omega_t^i)$, $i \in \mathcal{N}_t$, where $P_t = \{p_t(\omega_t^i)\}_{i \in \mathcal{N}_t}$ and $\mathcal{N}_t := \{1, \ldots, N_t\}$ is the index set associated with the support Ω_t, $t = 2, \ldots, T$.

Assumption 3. Variables $x_t \in \{0,1\}^{d_x}$ and $y_t \in \mathcal{Y}_t(x_t, \omega_t) \subseteq \mathbb{R}_+^{d_y}$ for every $t \in [T]$, where $\mathcal{Y}_t(x_t, \omega_t)$ is defined by either a mixed-integer set or a disjunctive set.

Remark 2. Note that, for general integer or discrete state variables x_t that is bounded, we can obtain its equivalent binary representation by using the binary expansion (e.g. see Bonami and Margot [8]). Also, even for mixed-integer state variables x_t, the binary expansion can be used to approximate the state variables. As discussed in Zou et al. [46], the use of the binary expansion is justified in MSIPs since the fraction of binary variables in the extensive formulation of an MSIP to all variables reaches zero as the number of stages or the size of support per stage increases.

Assumption 4. Sets X_1 and $X_t(x_{t-1}, \omega_t)$, given any $x_{t-1} \in \{0,1\}^{d_x}$ and $\omega_t \in \Omega_t$, for all $t = 2, \ldots, T$, are nonempty and compact.

Assumption 5. Functions $f_1(x_1, y_1)$ and $f_t(x_t, y_t, \omega_t)$, for $t = 2, \ldots, T$ and $\omega_t \in \Omega_t$, are linear.

For readers' convenience, below we list major abbreviations used in this paper:

DRA/DRR: Distributionally risk-averse/Distributionally risk-receptive.
MSLP/MSIP/MSDP: Multistage stochastic linear/integer/disjunctive program.
SDDP: Stochastic dual dynamic programming.
DA-SDDP: Our extension of SDDP that addresses distributional ambiguity.
DRR-SDDP-C: DA-SDDP for DRR-MSPs with a cutting plane-based approximation.
DRR-SDDP-R: DA-SDDP for DRR-MSPs with a reformulation-based approximation.
DRA-SDDP-C: DA-SDDP for DRA-MSPs with a cutting plane-based approximation.
DRA-SDDP-R: DA-SDDP for DRA-MSPs with a reformulation-based approximation.
MS-MFIP: Multistage stochastic maximum flow interdiction problem
MS-FLIP: Multistage stochastic facility location interdiction problem

2 Literature Review

In this section, we review literature related to solution approaches for MSLPs and MSIPs along with their distributionally robust variants.

Multistage Stochastic Linear Programs. As mentioned before, in order to solve MSLPs, NBD [6] approximates the expected cost-to-go function at each stage t, i.e., $\mathbb{E}_{P_{t+1}}[Q_{t+1}(\cdot)]$, by a piecewise linear function using Benders cutting planes, which are constructed by the dual solutions from the problem at the subsequent stage. For MSLPs under the stage-wise independence assumption, SDDP [35] is a NBD-like approach that harnesses scenario sampling to mitigate "curse of dimensionality" of dynamic programming without losing the (almost surely) finite convergence of the NBD algorithm. The (iteration) complexity and convergence of SDDP has also been further examined in [26, 39]; refer to [20] and references therein for recent advances for MSLPs.

Distributionally Robust (or Risk-averse) MSLPs. Recently, Philpott et al. [36] consider distributionally risk-averse multistage stochastic linear programs (denoted by DRA-MSLPs) where ambiguity sets are constructed based on χ^2 distance from a reference probability distribution. Their approaches are based on SDDP embedding separation algorithms that compute a worst-case probability distribution for the different reference probability distributions. Note that their problems consider only continuous variables. Duque and Morton [17] present an SDDP-based algorithm for DRA-MSLPs where the ambiguity set is defined using Wasserstein metric. Their approach is based on the dualization of the inner problem for finding a worst-case probability distribution. They present comparison analysis of the results from their algorithm and those from the modified algorithm of Philpott et al. [36] for Wasserstein metric. Park and Bayraksan [34] investigate DRA-MSLPs, where the ambiguity set is defined using ϕ-divergence, for an application to a water allocation problem. They propose a NBD-type algorithm relying on the dual reformulation of the inner problem for finding a worst-case probability distribution.

Multistage Stochastic Integer Programs. For solving MSIPs with binary state variables, an extension to SDDP, referred to as SDDiP, has been proposed by Zou et al. [46]. SDDiP uses a new class of cutting planes, constructed based on a Lagrangian relaxation where the strong duality holds for the resulting Lagrangian dual, to approximate the expected cost-to-go function in each stage. Zou et al. [46] provide the cut conditions under which SDDiP is finitely convergent. For a more general class of MSIPs, where all decision variables are allowed to be mixed-integer, there are several studies applying decomposition schemes directly to the deterministic equivalent formulation. For example, Carøe and Schultz [11] present a branch-and-bound algorithm based on a dual decomposition approach applied to the deterministic equivalent formulation. The approach uses a Lagrangian relaxation of *non-anticipativity* constraints which enforce the scenarios that follow the same history up to stage t to have the same decisions until stage t. As another example, Lulli and Sen [28] propose a branch-and-price algorithm, i.e., a branch-and-bound algorithm with column generation, for the same class of MSIPs. Recently, Büyüktahtakın [9] presents an approach for the MSIP with mixed-integer variables (in both risk-neutral and risk-averse settings), using cutting planes developed upon the concept of "scenario dominance" that is based on a partial ordering of scenarios. These cutting planes are generated by solving subproblems related to a subgroup of scenarios and added to the deterministic equivalent formulation within a branch-and-cut procedure. We also note that for multistage stochastic mixed-integer nonlinear programs, Zhang and Sun [45] present three decomposition algorithms—one based on NBD and two based on SDDP—which rely on a regularization of the expected cost-to-go function and a class of cutting planes called *generalized conjugacy cut* to approximate the function.

Distributionally Robust MSIPs. For DRA-MSIPs, much less has been studied in the literature. Yu and Shen [44] investigate decision-dependent DRA-MSIPs, where state variables are binary, and the ambiguity set depends on the state decisions made at the previous stage. They consider three types of ambiguity sets constructed based on the decision-dependent moment information (e.g., mean and variance). They propose mixed-integer linear programming and mixed-integer semidefinite programming reformulations of the problems and solve them using SDDiP. Recently, in the dissertation of Nakao [31], a dual decomposition approach is presented for a DRA-MSIP where variables can be mixed-integer, and the ambiguity sets are defined using Wassertstein metric. The approach applies the dual reformulations to the inner problems over Wasserstein ambiguity sets in a consecutive manner for deriving a monolithic-minimization deterministic equivalent formulation of the DRA-MSIP. Then, they use a Lagrangian relaxation of the non-anticipativity constraints in the deterministic equivalent formulation to derive a Lagrangian dual. They provide computational results obtained by solving the Lagrangian dual using the branch-and-bound algorithm, proposed by Carøe and Schultz [11]. It is worth noting that for DRA-MSIPs, DRA-MSDPs, and DRA-MSLPs with $T = 2$, i.e., two stages, solution approaches are presented in [2, 4], [3], and [2, 19] respectively.

Distributionally Risk-Receptive Programs. Duchi et al. [15] employ DRR and DRA programs to construct a confidence interval for an optimal objective of a stochastic program with true probability distribution. Similarly, Cao and Gao [10] consider decision-making problems involving covariate data where uncertain parameters belong to a specific uncertainty set. They show that solving robust and optimistic optimization problems leads to worst-case and best-case rewards, respectively, which construct a confidence interval for the true reward. Gotoh et al. [22] investigate DRA and DRR programs in comparison to a sample average approximation. In particular, they show that by solving both DRA and DRR programs, one of their solutions always outperform the sample average approximation solutions in terms of out-of-sample test performance. Nakao et al. [32] consider a partially observable Markov decision process with distributional ambiguity. They solve a DRR model to obtain an upper bound on a true value function of a DRA partial observable Markov decision process. Moreover, DRR programs are studied in the context of reinforcement learning

[41] to find an optimistic policy and Bayesian statistics [33] to approximate a likelihood. However, these studies [41, 33] focus on only the problem of finding a best-case probability distribution and do not consider a decision-making procedure as in our DRR frameworks. Note that aforementioned studies have not considered multistage decision-making with integer variables and are not applicable for interdiction problems considered in this paper.

3 SDDP and its Extension to Distributional Ambiguity

In this section, we review SDDP developed for MSLPs and extend it to handle distributional ambiguity. This extension, referred to as *distributionally ambiguous SDDP* (DA-SDDP), serves as a broader architecture for our approaches to solve the DRA-MSP and DRR-MSP models.

The key ideas of SDDP are to approximate, rather than exactly evaluate, the expected values of the future costs, $\mathcal{Q}_{t+1}(x_t) := \mathbb{E}_{P_{t+1}}[Q_{t+1}(x_t, \boldsymbol{\omega}_{t+1})]$ for $t \in [T-1]$, using a set of cuts and to tighten this approximation in an iterative process that consists of two main steps: forward and backward steps. During a forward step, SDDP samples scenario paths, and along each scenario path it solves the stage problems (1) and (2) with approximations of the function $\mathcal{Q}_{t+1}(x_t)$, for all $t \in [T]$. This generates feasible solutions to the MSLP. In a backward step, SDDP solves the linear programming dual of the approximating problems at each stage t for all realizations given the feasible solutions obtained from the forward step. These dual solutions are then used to derive a valid cut that is added to the approximating problem at the previous stage $t-1$ to further tighten the approximation of the function $\mathcal{Q}_t(x_{t-1})$.

We propose an extension of SDDP, referred to as DA-SDDP, for both DRA-MSPs and DRR-MSPs, that approximates the pessimistic and optimistic expected cost-to-go functions: $\mathcal{Q}^{RA}_{t+1}(x_t) := \max\{\mathbb{E}_{P_{t+1}}[Q^{RA}_{t+1}(x_t, \boldsymbol{\omega}_{t+1})] : P_{t+1} \in \mathcal{P}_{t+1}\}$ and $\mathcal{Q}^{RR}_{t+1}(x_t) := \min\{\mathbb{E}_{P_{t+1}}[Q^{RR}_{t+1}(x_t, \boldsymbol{\omega}_{t+1})] : P_{t+1} \in \mathcal{P}_{t+1}\}$, for all $t \in [T-1]$, by constructing a convex lower approximation for each function. Note that these functions are not piecewise linear and convex for DRR- and DRA-MSIP as in the case of SDDP for MSLPs. This is because of the presence of integer variables in each stage. Moreover, even for a DRR-MSP with only continuous variables in each stage, $\mathcal{Q}^{RR}_{t+1}(x_t)$ is nonconvex.

Algorithm 1 Distributionally Ambiguous Stochastic Dual Dynamic Programming

```
Initialize l ← 1; x_0 ← initial state; ω_1 ← data at the first stage; Ω_1 := {ω_1}; K_t^l ← 0 for
  t = 1, ..., T − 1;
while (satisfying none of stopping conditions) do
  Sample a scenario path ξ^l ∈ Ξ := Ω_1 × ··· × Ω_T
  for t ∈ [T] do                                                      ▷ Forward Step
    Solve subproblem ℙ_t^l(x_{t−1}^l, ξ_t^l) and obtain (x_t^l, y_t^l) and Q̂_t^l(x_{t−1}^l, ξ_t^l)
  for t = T, ..., 2 do                                                ▷ Backward Step
    for i ∈ 𝒩_t do
      Solve relaxation ℙ̃_t^l(x_{t−1}^l, ω_t^i) and obtain cut (α_{t−1}^{i,l}, β_{t−1}^{i,l})
    Refine approximating function φ_{t−1}^l by using cuts (α_{t−1}^{i,l}, β_{t−1}^{i,l}), i ∈ 𝒩_t
    K_{t−1}^l ← K_{t−1}^l + 1
  Solve subproblem ℙ_1^l(x_0^l, ω_1) and obtain the bound LB
  K_t^{l+1} ← K_t^l for t = 1, ..., T − 1
  l ← l + 1
return Subproblems {ℙ_t^l}_{t∈[T−1]}, LB
```

A pseudocode of DA-SDDP is given in Algorithm 1. The algorithm is initialized with a predetermined initial state of the model, denoted by x_0, the input data for the first stage, denoted by ω_1, and the singleton set $\Omega_1 := \{\omega_1\}$, which are introduced to simplify the notation later. We also set the iteration counter l to 1, and the number of cuts K_t^l to 0. At iteration l, the algorithm samples a scenario path $\xi^l = (\xi_1^l, \cdots, \xi_T^l)$ from $\Xi := \Omega_1 \times \cdots \times \Omega_T$ (Line 3). Note that this can be readily extended to the sampling of multiple scenario paths per iteration. The remainder of the iteration consists of a forward step (Lines 4 and 5) and a backward step (Lines 6 to 11).

Forward Step. For each stage $t \in [T]$, DA-SDDP solves the following approximation of Problem (4) (or (6)), which we refer to as *subproblem* and denote by $\mathbb{P}_t^l(x_{t-1}^l, \xi_t^l)$ (Line 5):

$$\hat{Q}_t^l(x_{t-1}^l, \xi_t^l) := \min_{(x_t, y_t) \in X_t(x_{t-1}^l, \xi_t^l)} \left\{ f_t(x_t, y_t, \xi_t^l) + \phi_t^l(x_t) \right\}, \quad t \in [T], \tag{11}$$

where $\phi_T^l(\cdot) = 0$, and $x_{t-1}^l, t = 2, .., T$, is an optimal stage-$(t-1)$ solution. To simplify notation, we let $x_0^l := x_0$ and $X_1(x_0^l, \xi_1^l) := X_1$. Function $\phi_t^l(x_t)$, for each $t \in [T-1]$, is a piecewise linear and convex function that is constructed by K_t^l cuts and it serves as an under-approximation of the pessimistic and optimistic expected cost-to-go functions—$\mathcal{Q}_{t+1}^{RA}$ and $\mathcal{Q}_{t+1}^{RR}$—while solving DRA-MSP (3) and DRR-MSP (5), respectively. The details of methods to construct the function $\phi_t^l(x_t)$ will be presented in Sections 4 and 5.

Backward Step. For each $t = T, \ldots, 2$, the algorithm solves relaxations of the subproblems, denoted by $\tilde{\mathbb{P}}_t^l(x_{t-1}^l, \omega_t^i)$, and obtain affine cuts, denoted by their coefficients $(\alpha_{t-1}^{i,l}, \beta_{t-1}^{i,l})$ for $i \in \mathcal{N}_t$ (Line 8). These cuts (supporting hyperplanes) provide a lower-bounding approximation of the value function $\hat{Q}_t^l(\cdot, \cdot)$ such that

$$\hat{Q}_t^l(x_{t-1}, \omega_t^i) \geq (\alpha_{t-1}^{i,l})^\top x_{t-1} + \beta_{t-1}^{i,l}, \quad \forall x_{t-1} \in \{0,1\}^{d_x}, i \in \mathcal{N}_t, \tag{12}$$

and

$$\hat{Q}_t^l(x_{t-1}^l, \omega_t^i) = (\alpha_{t-1}^{i,l})^\top x_{t-1}^l + \beta_{t-1}^{i,l}. \tag{13}$$

It should be noted that the validity is defined for $\hat{Q}_t^l(\cdot, \cdot)$, yet it is sufficient to construct the function $\phi_{t-1}^l(\cdot)$ to yield a lower bound for the optimistic or pessimistic expected cost-to-go function, since $\hat{Q}_t^l(\cdot, \cdot)$ is always lower-bounding the exact value function $Q_t^{RR}(\cdot, \cdot)$ or $Q_t^{RA}(\cdot, \cdot)$, respectively. Using the cuts $\{(\alpha_{t-1}^{i,l}, \beta_{t-1}^{i,l})\}_{i \in \mathcal{N}_t}$, the algorithm tightens the approximating function $\phi_{t-1}^l(x_{t-1})$ and improves the lower bound (Line 9). In Line 11, the algorithm computes the lower bound on the overall optimal objective value by solving the subproblem associated with the first stage.

The algorithm repeats these iterations with the forward and backward steps until one of predetermined stopping conditions is satisfied. These conditions can include a maximum number of iterations, a limit on elapsed time, or convergence of the lower bound.

Note that there exist various ways of generating a cut $(\alpha_{t-1}^{i,l}, \beta_{t-1}^{i,l}), i \in \mathcal{N}_t$, which is a supporting hyperplane of the epigraph of $\hat{Q}_t^l(x_{t-1}, \omega_t^i)$, intersecting at $x_{t-1} = x_{t-1}^l$, i.e., a cut satisfying both (12) and (13). By solving the subproblem to optimality, we can obtain an integer optimality cut [27], given a lower bound for the value function $\hat{Q}_t^l(\cdot, \cdot)$, that satisfies (12) and (13). As another example, consider a Benders cut obtained by solving a linear programming relaxation of the subproblem when solving a DRA-MSIP (or DRR-MSIP). This cut satisfies (12), though not necessarily (13). However, we can derive a mixed-binary linear programming reformulation of the subproblem by adding binary variables replacing integer variables, use the hierarchy of relaxations (discussed in Section 6.2) to solve the reformulation to optimality, and obtain a Benders cut that satisfies both (12) and (13).

4 DA-SDDP Algorithms for DRR-MSIPs

In this section, we develop two DA-SDDP algorithms for DRR-MSIPs: DRR-SDDP-C and DRR-SDDP-R algorithms, where the approximation functions $\phi_t^l(x_t)$ are derived using cutting planes and reformulation techniques, respectively.

We define an approximating problem that is associated with the optimistic expected cost-to-go function $\mathcal{Q}_{t+1}^{RR}(x_t)$ and is constructed using cuts $\{(\alpha_t^{i,k}, \beta_t^{i,k}) : i \in \mathcal{N}_{t+1}, k \in [K_t^l]\}$, for each $t \in [T-1]$:

$$\phi_t^{l,B}(x_t) := \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \theta_t^i \tag{14a}$$

$$\text{s.t. } \theta_t^i \geq (\alpha_t^{i,k})^\top x_t + \beta_t^{i,k}, \quad \forall k \in [K_t^l],\ i \in \mathcal{N}_{t+1}, \tag{14b}$$

where $P_{t+1} = \{p_{t+1}^i\}_{i \in \mathcal{N}_{t+1}}$. By construction, solving this problem yields a lower bound on $\mathcal{Q}_{t+1}^{RR}(x_t)$. In the following sections, we describe how DRR-SDDP-C and DRR-SDDP-R algorithms construct the approximating functions $\phi_t^l(x_t)$, for $t \in [T-1]$, using cutting planes that utilize the binary property of the state variables and a mixed-integer linear programming reformulation of the problem (14), respectively. Each of these algorithm is followed by the convergence analysis.

4.1 DRR-SDDP-C algorithm with a cutting plane-based approximation

We define a cutting plane-based approximation function at iteration l for each $t \in [T-1]$ as follows:

$$\phi_t^{l,C}(x_t) := \min\ \phi \tag{15a}$$

$$\text{s.t. } \phi \geq (\delta_t^k)^\top (x_t - x_t^k) + \rho_t^k, \quad \forall k \in [K_t^l], \tag{15b}$$

where

$$\delta_{t,j}^k := \begin{cases} \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \alpha_{t,j}^{i,k}, & \text{if } x_{t,j}^k = 0, \\ \max_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \alpha_{t,j}^{i,k}, & \text{if } x_{t,j}^k = 1, \end{cases} \tag{16}$$

and

$$\rho_t^k := \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \big((\alpha_t^{i,k})^\top x_t^k + \beta_t^{i,k} \big),$$

for each $t \in [T-1]$ and $k \in [K_t^l]$. Recall that x_t^k and $(\alpha_t^{i,k}, \beta_t^{i,k})$ are an optimal stage-t solution and a cut generated in Line 8 at iteration $k \in [K_t^l]$. In Lemma 1, we prove that $\phi_t^{l,C}(x_t)$ provides a lower-bound approximation for $\mathcal{Q}_{t+1}^{RR}(x_t)$. As a consequence, we can replace $\phi_t^l(x_t)$ with $\phi_t^{l,C}(x_t)$ in Algorithm 1 to get the DRR-SDDP-C algorithm.

Lemma 1. *The cutting plane-based approximating function provides a lower bound for the optimistic expected cost-to-go function, i.e., $\phi_t^{l,C}(x_t) \leq \mathcal{Q}_{t+1}^{RR}(x_t)$ for all $l > 0, x_t \in \{0,1\}^{d_x}$ and $t \in [T-1]$.*

Proof. For any $k \in [K_t^l]$ and $x_t \in \{0,1\}^{d_x}$, it is satisfied that

$$\min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \hat{Q}_{t+1}^l(x_t, \omega_{t+1}^i) \geq \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \Big((\alpha_t^{i,k})^\top x_t + \beta_t^{i,k} \Big) \tag{17a}$$

$$= \rho_t^k + \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i (\alpha_t^{i,k})^\top (x_t - x_t^k) \tag{17b}$$

$$\geq \rho_t^k + \sum_{j \in [d_x]} \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \alpha_{t,j}^{i,k} (x_{t,j} - x_{t,j}^k). \tag{17c}$$

In the last inequality, if $x^k_{t,j} = 0$, then $(x_{t,j} - x^k_{t,j}) \in \{0, 1\}$. It follows that we can fix the coefficient of the term $(x_{t,j} - x^k_{t,j})$ to

$$\delta^k_{t,j} = \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p^i_{t+1} \alpha^{i,k}_{t,j}. \tag{18}$$

If $x^k_{t,j} = 1$, then $(x_{t,j} - x^k_{t,j}) \in \{0, -1\}$. In this case, we can fix the coefficient to

$$\delta^k_{t,j} = \max_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p^i_{t+1} \alpha^{i,k}_{t,j}. \tag{19}$$

Fixing the coefficients for all $j \in d_x$ and $k \in [K^l_t]$, we have

$$\min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p^i_{t+1} \hat{Q}^l_{t+1}(x_t, \omega^i_{t+1}) \geq (\delta^k_t)^\top (x_t - x^k_t) + \rho^k_t, \quad \forall k \in [K^l_t]. \tag{20}$$

Function $\phi^{l,C}_t(x_t)$ is constructed using the affine functions in the right-hand side of (20), and thus it follows that $\phi^{l,C}_t(x_t) \leq \mathcal{Q}^{RR}_{t+1}(x_t)$. □

Now, we show the finite convergence of the DRR-SDDP-C algorithm. To this end, we define a *policy* by a collection of decision functions $\{\bar{x}_t(\xi_{[t]}), \bar{y}_t(\xi_{[t]})\}_{t \in [T]}$, where $\xi_{[t]} = (\xi_1, \ldots, \xi_t)$, which serves as a decision rule given any scenario path $(\xi_1, \ldots, \xi_T)$. A policy is *optimal* for a DRR-MSIP if $(\bar{x}_t(\xi_{[t]}), \bar{y}_t(\xi_{[t]}))$ is optimal to the t-th stage problem (6) ((5) for $t = 1$) for $t \in [T]$ and all $\xi \in \Xi$.

Theorem 1. *The forward step of the DRR-SDDP-C algorithm defines an optimal policy for a DRR-MSIP in a finite number of iterations of its while loop with probability one. Furthermore, each iteration of the while loop is executed in a finite time if there exists a finite-time algorithm for computing the cut coefficients* (16)*.*

Proof. We first show that the finiteness of the while loop. Let $\{\bar{x}^l_t(\xi_{[t]}), \bar{y}^l_t(\xi_{[t]}) : t \in [T]\}$ be a policy that is defined by the forward step at iteration l of the DRR-SDDP-C algorithm. Then, to achieve the optimality of the policy, it suffices to show that $\phi^{l,C}_t(\bar{x}^l_t(\xi_{[t]})) = \mathcal{Q}^{RR}_{t+1}(\bar{x}^l_t(\xi_{[t]}))$ for $t \in [T-1]$ and all $\xi \in \Xi$.

Let $\mathcal{L}$ denote the set of iterations where the policy is non-optimal. For each $t \in [T-1]$, let $l_t \in \mathcal{L}$ be the largest iteration such that $\phi^{l,C}_t(\bar{x}^l_t(\xi_{[t]})) < \mathcal{Q}^{RR}_{t+1}(\bar{x}^l_t(\xi_{[t]}))$ for some $\xi \in \Xi$. Also, given iteration m, let $\bar{X}^m_t, t \in [T-1]$, be the set of $\bar{x}^l_t(\xi_{[t]})$ for all future iterations $l \in \mathcal{L} \setminus [m]$ and $\xi \in \Xi$, i.e., $\bar{X}^m_t = \{\bar{x}^l_t(\xi_{[t]}) : \xi \in \Xi, l > m, l \in \mathcal{L}\}$.

We first show that l_{T-1} is finite with probability one. At the forward step of iteration l, if we observe that $\phi^{l,C}_{T-1}(\bar{x}) < \mathcal{Q}^{RR}_T(\bar{x})$ for any $\bar{x} \in \bar{X}^1_{T-1}$, then for all future iterations $m > l$ we have

$$\phi^{m,C}_{T-1}(\bar{x}) \geq \min_{P_T \in \mathcal{P}_T} \sum_{i \in \mathcal{N}_T} p^i_T \hat{Q}^l_T(\bar{x}, \omega^i_T) = \mathcal{Q}^{RR}_T(\bar{x}). \tag{21}$$

The above holds since the cut added at iteration l (in the form of (15b)) satisfies both (12) and (13), and by definition $\hat{Q}^l_T$ is equivalent to Q^{RR}_T. It follows that $\phi^{m,C}_{T-1}(\bar{x}) = \mathcal{Q}^{RR}_T(\bar{x})$ for any $m > l$. Since $|\bar{X}^1_{T-1}| < \infty$ and every $\xi \in \Xi$ has a positive probability to be sampled, it holds with probability one that $\phi^{l,C}_{T-1}(\bar{x}) = \mathcal{Q}^{RR}_T(\bar{x})$ for all $\bar{x} \in \bar{X}^1_{T-1}$ after finitely many iterations. This shows that $l_{T-1} < \infty$ with probability one.

Next, we show that l_{T-2} is also finite with probability one. Suppose at iteration $l \geq l_{T-1}$ we observe that $\phi^{l,C}_{T-2}(\bar{x}) < \mathcal{Q}^{RR}_{T-1}(\bar{x})$ for any $\bar{x} \in \bar{X}^{l_{T-1}}_{T-2}$. Then, we have $\phi^{m,C}_{T-2}(\bar{x}) = \mathcal{Q}^{RR}_{T-1}(\bar{x})$ for all future iterations $m > l$ since

$$\phi^{m,C}_{T-2}(\bar{x}) \geq \min_{P_{T-1} \in \mathcal{P}_{T-1}} \sum_{i \in \mathcal{N}_{T-1}} p^i_{T-1} \hat{Q}^l_{T-1}(\bar{x}, \omega^i_{T-1}) \tag{22a}$$

$$= \min_{P_{T-1} \in \mathcal{P}_{T-1}} \sum_{i \in \mathcal{N}_{T-1}} p^i_{T-1} Q^{RR}_{T-1}(\bar{x}, \omega^i_{T-1}) = \mathcal{Q}^{RR}_{T-1}(\bar{x}). \tag{22b}$$

The first equality holds since $\phi^{m,C}_{T-1}(\bar{x}) = \mathcal{Q}^{RR}_{T}(\bar{x})$ for $m > l_{T-1}$. So, we have $\phi^{m,C}_{T-2}(\bar{x}) = \mathcal{Q}^{RR}_{T-1}(\bar{x})$ for any future iteration $m > l$. Since $|\bar{X}^{l_{T-1}}_{T-2}| < \infty$ and every $\xi \in \Xi$ has a positive probability, there exists a finite iteration l such that $\phi^{l,C}_{T-2}(\bar{x}) = \mathcal{Q}^{RR}_{T-1}(\bar{x})$ for all $\bar{x} \in \bar{X}^{l_{T-1}}_{T-2}$ with probability one, which implies $l_{T-2} < \infty$ with probability one. Similarly, we can prove by induction that l_t, for all $t \in [T-1]$, are finite. This proves that $|\mathcal{L}| < \infty$ with probability one.

Now, we show that each iteration terminates in finite time. This is followed by the facts that the subproblems are bounded and thus solvable in a finite time using a branch-and-cut algorithm and Line 9 is executed in a finite time by assumption. □

4.2 DRR-SDDP-R algorithm with a reformulation-based approximation

For each constraint in (14b), we multiply p^i_{t+1} to both sides of inequalities for each $i \in \mathcal{N}_{t+1}$, and replace $p^i_{t+1} x_t$ with a decision vector η^i_t in the right-hand side of the resulting inequalities. This yields the following system of inequalities:

$$p^i_{t+1}\theta^i_t \geq (\alpha^{i,k}_t)^\top \eta^i_t + \beta^{i,k}_t p^i_{t+1}, \quad \forall k \in [K^l_t],\ i \in \mathcal{N}_{t+1}, \tag{23a}$$

$$\eta^i_t \leq x_t, \qquad \forall i \in \mathcal{N}_{t+1}, \tag{23b}$$

$$\eta^i_t \leq p^i_{t+1}, \qquad \forall i \in \mathcal{N}_{t+1}, \tag{23c}$$

$$\eta^i_t \geq p^i_{t+1} + x_t - 1, \qquad \forall i \in \mathcal{N}_{t+1}, \tag{23d}$$

$$\eta^i_t \geq 0, \qquad \forall i \in \mathcal{N}_{t+1}. \tag{23e}$$

Notice that a system of inequalities (23b) to (23e) ensure that a feasible η^i_t equals to $p^i_{t+1} x_t$, given any $x_t \in \{0,1\}^{d_x}$ and $\{p^i_{t+1}\}_{i \in \mathcal{N}_{t+1}} \in [0,1]^{N_{t+1}}$. We introduce an additional variable $\bar{\theta}^i_t$ to replace $p^i_{t+1}\theta^i_t$. This yields a reformulation-based approximating function of $\mathcal{Q}^{RR}_{t+1}(x_t)$ at iteration l for each $t \in [T-1]$ as follows:

$$\phi^{l,R}_t(x_t) := \min \sum_{i \in \mathcal{N}_{t+1}} \bar{\theta}_i \tag{24a}$$

$$\text{s.t. } \bar{\theta}_i \geq (\alpha^{i,k}_t)^\top \eta^i_t + \beta^{i,k}_t p^i_{t+1},\ \forall k \in [K^l_t], i \in \mathcal{N}_{t+1}, \tag{24b}$$

$$\text{(23b)–(23e)}, \tag{24c}$$

$$P_{t+1} \in \mathcal{P}_{t+1}. \tag{24d}$$

Since θ^i_t is not restricted, we can readily show that the equivalence between (23a) and (24b). Therefore, by construction, $\phi^{l,R}_t(x_t)$ equals to the value function $\phi^{l,B}_t(x_t)$ (14), which is a lower bound for $\mathcal{Q}^{RR}_{t+1}(x_t)$. The DRR-SDDP-R algorithm is defined as Algorithm 1 with the function $\phi^l_t(x_t)$ being replaced by $\phi^{l,R}_t(x_t)$ for every iteration l and stage t. We note that the above formulation becomes a linear program when the ambiguity set $\mathcal{P}_{t+1}$ is defined by a polytope (e.g., Wasserstein ambiguity set with a finite support).

We show the finite convergence of the DRR-SDDP-R algorithm as follows.

Theorem 2. *The forward step of the DRR-SDDP-R algorithm defines an optimal policy to a DRR-MSIP in a finite number of iterations of its while loop with probability one. Furthermore, each iteration of the while loop is executed in a finite time if the ambiguity set at every stage is defined by a polytope or a mixed-binary linear set.*

Proof. By following a similar argument to the proof of Theorem 1, we can show that the number of the while-loop iterations required to define an optimal policy is finite with probability one and each while-loop iteration is executed in a finite time. □

5 DA-SDDP Algorithms for DRA-MSIPs

In this section, we develop two DA-SDDP algorithms for DRA-MSIPs: DRA-SDDP-C and DRA-SDDP-R algorithms, where the approximation function $\phi_t^l(x_t)$ is constructed using cutting planes derived by a separation approach and a reformulation derived by utilizing the strong duality, respectively.

5.1 DRA-SDDP-C algorithm with a cutting plane-based approximation

Recall that we consider the cuts, $\{(\alpha_t^{i,k}, \beta_t^{i,k}) : i \in \mathcal{N}_{t+1}, k \in [K_t^l]\}$, for each $t \in [T-1]$ to get approximations in the backward steps of DA-SDDP. For DRA-MSIPs, we identify a worst-case probability distribution in each iteration k for the given cuts $(\alpha_t^{i,k}, \beta_t^{i,k}), i \in \mathcal{N}_{t+1}$, and the given solution x_t^k by solving the following problem, referred to as *distribution separation problem*:

$$\max_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \big((\alpha_t^{i,k})^\top x_t^k + \beta_t^{i,k} \big), \tag{25}$$

for each $t \in [T-1]$. Let $\{p_{t+1}^{i,k}\}_{i \in \mathcal{N}_{t+1}}$ be an optimal solution to the distribution separation problem (25). Then, we define a cutting plane-based approximating function for iteration l and $t \in [T-1]$ as follows:

$$\phi_t^{l,S}(x_t) := \min \Bigg\{ \phi : \phi \geq (\pi_t^k)^\top x_t + \gamma_t^k, \ \forall k \in [K_t^l] \Bigg\}, \tag{26}$$

where

$$\pi_t^k = \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^{i,k} \alpha_t^{i,k}, \tag{27a}$$

$$\gamma_t^k = \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^{i,k} \beta_t^{i,k}. \tag{27b}$$

Consequently, we define the DRA-SDDP-C algorithm as Algorithm 1 with $\phi_t^l(x_t)$ being replaced by $\phi_t^{l,S}(x_t)$ for every iteration l and $t \in [T-1]$. The following theorem shows the finite convergence of the DRA-SDDP-C algorithm.

Theorem 3. *The forward step of the DRA-SDDP-C algorithm provides an optimal policy for DRA-MSIP in a finite number of iterations of its while loop with probability one. Furthermore, each iteration of the while loop is executed in a finite time if there exists a finite-time algorithm for solving the distribution separation problem* (25)*.*

Proof. Let $\{\bar{x}_t^l(\xi_{[t]}), \bar{y}_t^l(\xi_{[t]})\}_{t \in [T]}$ be a policy that is defined by the forward step at iteration l. To show its optimality, it suffices to show $\phi_t^{l,S}(\bar{x}_t^l(\xi_{[t]})) = \mathcal{Q}_{t+1}^{RA}(\bar{x}_t^l(\xi_{[t]}))$ for $t \in [T-1]$ and all $\xi \in \Xi$. We can prove the statement by following a similar argument to the proof of Theorem 1. □

Remark 3. There are various types of ambiguity sets for which we can solve problem (25) and compute the coefficients (16) using a finite-time algorithm; e.g., ambiguity sets constructed using Wasserstein metric [21], moment information [44], total variation distance [5] and χ^2 distance [36], where their supports are finite.

5.2 DRA-SDDP-R algorithm with dual-based reformulation

We present a reformulation-based approach, referred to as DRA-SDDP-R algorithm, that relies on the dualization of the inner maximization problems in (3) and (4), i.e., $\max\{\mathbb{E}_{P_{t+1}}[Q^{RA}_{t+1}(x_t, \boldsymbol{\omega}_{t+1})] : P_{t+1} \in \mathcal{P}_{t+1}\}$. We note that similar reformulation-based approaches are studied in recent papers for DRA-MSLPs with ϕ-divergence-based ambiguity sets [34], DRA-MSLPs with Wasserstein ambiguity sets [17], and DRA-MSIPs with moment-based ambiguity sets [44]. In the following, we demonstrate the DRA-SDDP-R algorithm for DRA-MSIPs with Wasserstein ambiguity sets, but we note that this approach can be applied to DRA-MSIPs with a general family of ambiguity sets for which dual formulations of DRA models are available.

The Wasserstein ambiguity set is defined for $t \in \{2, \dots, T\}$ as

$$\mathcal{P}^W_t := \Bigg\{ P_t \in \mathbb{R}^{N_t}_+ : \sum_{i \in \mathcal{N}_t} p^i_t = 1,\ \sum_{j \in \mathcal{N}_t} v_{ij} = p^i_t,\ i \in \mathcal{N}_t,\ \sum_{i \in \mathcal{N}_t} v_{ij} = \bar{p}^j_t,\ j \in \mathcal{N}_t,$$
$$\sum_{i \neq j \in \mathcal{N}_t} \|\omega^i_t - \omega^j_t\| v_{ij} \le \epsilon,\ v_{ij} \ge 0,\ \forall i, j \in \mathcal{N}_t \Bigg\},$$

where $\{\bar{p}^i_t\}_{i \in \mathcal{N}_t}$ is a reference probability distribution on Ω_t for $t \in \{2, \dots, T\}$, and $\|\cdot\|$ is the notation for any norm. For a given $\epsilon > 0$, each ambiguity set $\mathcal{P}^W_t, t \in \{2, \dots, T\}$, is a Wasserstein ball containing all probability distributions within ϵ-Wasserstein distance from the reference probability distribution. The dual of the maximization in (3) and (4) for $t \in [T-1]$ is given as follows:

$$\min_{\gamma \ge 0} \Bigg\{ \epsilon\gamma - \sum_{i \in \mathcal{N}_{t+1}} \bar{p}^i_{t+1} \nu^i : \nu^i + \|\omega^i_{t+1} - \omega^j_{t+1}\| \gamma \ge Q^{RA}_{t+1}(x_t, \omega^j_{t+1}),\ \forall i, j \in \mathcal{N}_{t+1} \Bigg\}. \tag{28}$$

Here, the strong duality holds by Theorem 1 in [21]. Then, given cuts $(\alpha^{i,k}_t, \beta^{i,k}_t), i \in \mathcal{N}_{t+1}, k \in [K^l_t]$, we define a reformulation-based approximating function for iteration l and $t \in [T-1]$ as follows:

$$\phi^{l,D}_t(x_t) := \min \Bigg\{ \epsilon\gamma - \sum_{i \in \mathcal{N}_{t+1}} \bar{p}^i_{t+1} \nu^i : \gamma \ge 0,$$
$$\nu^i + \|\omega^i_{t+1} - \omega^j_{t+1}\| \gamma \ge (\alpha^{i,k}_t)^\top x_t + \beta^{i,k}_t,\ \forall i, j \in \mathcal{N}_{t+1}, k \in [K^l_{t+1}] \Bigg\}.$$

Consequently, the DRA-SDDP-R algorithm for DRA-MSIPs with Wasserstein ambiguity set are defined as Algorithm 1 in which $\phi^l_t(x_t)$ is being replaced with $\phi^{l,D}_t(x_t)$.

6 Extensions to Multistage Stochastic Disjunctive Programs with Distributional Ambiguity

In this section, we present extensions of the DA-SDDP algorithms to DRR- and DRA-MSDPs defined in Section 1.1, under the following assumption: For $h \in H_t$, set $\{(x_t, y_t) \in \mathbb{R}^{d_x}_+ \times \mathbb{R}^{d_y}_+ : A^h_t(\omega_t)x_t + B^h_t(\omega_t)y_t \ge b^h_t(\omega_t) - C^h_t(\omega_t)x_{t-1}\}$ is nonempty and compact for any $x_{t-1} \in \{0,1\}^{d_x}$ and $\omega_t \in \Omega_t$. Also, its constraints include $x_{t-1} \ge 0, x_{t-1} \le 1, x_t \ge 0$, and $x_t \le 1$.

6.1 DA-SDDP algorithms for DRR- and DRA-MSDPs

Let us consider a set of cuts, $\{(\pi_t^k, \gamma_t^k)\}_{k\in[K_t^l]}$, for iteration l and stage t, constructing an approximation of the pessimistic (risk-averse) or optimistic (risk-receptive) expected cost-to-go function. Then, with the same definition of x_0 and ω_1 as in Algorithm 1, the subproblem at iteration l is given by

$$\hat{Q}_t^l(x_{t-1}, \omega_t) = \min \quad f_t(x_t, y_t, \omega_t) + \phi_t \tag{29a}$$

$$\text{s.t.} \quad \phi_t \geq (\pi_t^k)^\top x_t + \gamma_t^k, \ \forall k \in [K_t^l]; \qquad (x_t, y_t) \in X_t(x_{t-1}, \omega_t), \tag{29b}$$

for $t \in [T]$ and $\omega_t \in \Omega_t$, where $\phi_T = K_T^l = 0$ and $X_t(x_{t-1}, \omega_t)$ is defined by a disjunctive set (7). We define the feasible set of the foregoing subproblem by

$$\begin{aligned}\mathcal{D}_t^l(x_{t-1}, \omega_t) := \Big\{&(x_t, y_t, \phi_t) \in \mathbb{R}_+^{d_x} \times \mathbb{R}_+^{d_y} \times \mathbb{R}_+ : \\ &\bigvee_{h\in H_t} \Big(\phi_t - (\pi_t^k)^\top x_t \geq \gamma_t^k, \ k \in [K_t^l], \ A_t^h(\omega_t)x_t + B_t^h(\omega_t)y_t \geq b_t^h(\omega_t) - C_t^h(\omega_t)x_{t-1}\Big)\Big\},\end{aligned}$$

and derive the convex hull of $\mathcal{D}_t^l(x_{t-1}, \omega_t)$ in the following proposition.

Proposition 1. *For any $x_{t-1} \in \{0,1\}^{d_x}$ and $\omega_t \in \Omega_t$, the convex hull of the set $\mathcal{D}_t^l(x_{t-1}, \omega_t), t \in [T]$, is equivalent to the projection of polyhedral set $\tilde{\mathcal{D}}_t^l(x_{t-1}, \omega_t)$ onto the (x_t, y_t, ϕ_t)-space where*

$$\begin{aligned}\tilde{\mathcal{D}}_t^l(x_{t-1}, \omega_t) := \Bigg\{ &\sum_{h\in H_t} \zeta_{t,0}^h = 1, \sum_{h\in H_t} \zeta_{t,1}^h - x_t = 0, \sum_{h\in H_t} \zeta_{t,2}^h - y_t = 0, \\ &\sum_{h\in H_t} \zeta_{t,3}^h = x_{t-1}, \sum_{h\in H_t} \zeta_{t,4}^h - \phi_t = 0, \\ &A_t^h(\omega_t)\zeta_{t,1}^h + B_t^h(\omega_t)\zeta_{t,2}^h + C_t^h(\omega_t)\zeta_{t,3}^h - b_t^h(\omega_t)\zeta_{t,0}^h \geq 0, \quad h \in H_t, \\ &\zeta_{t,4}^h - (\pi_t^k)^\top \zeta_{t,1}^h - \gamma_t^k \zeta_{t,0}^h \geq 0, \quad h \in H_t, k \in K_t^l, \\ &x_t \in \mathbb{R}_+^{d_x}, y_t \in \mathbb{R}_+^{d_y}, \phi_t \in \mathbb{R}_+, \\ &\zeta_{t,0}^h \in \mathbb{R}_+, \zeta_{t,1}^h \in \mathbb{R}_+^{d_x}, \zeta_{t,2}^h \in \mathbb{R}_+^{d_y}, \zeta_{t,3}^h \in \mathbb{R}_+^{d_x}, \zeta_{t,4}^h \in \mathbb{R}_+, h \in H_t \Bigg\}.\end{aligned} \tag{30}$$

Proof. Refer to Appendix A. $\square$

Using Proposition 1, we derive an extension of DA-SDDP for DRR- and DRA-MSDPs, namely DA-SDDP-DP. Its pseudocode is provided in Algorithm 2.

DA-SDDP-DP shares a similar structure to DA-SDDP, but note that it involves distinct subproblems and a special subroutine for adding cuts to the subproblems. In particular, it solves the linear programming equivalents of subproblems (29), derived using Proposition 1 and referred to as *LP-subproblems*:

$$\min \Big\{ f_t(x_t, y_t, \omega_t) + \phi_t : (x_t, y_t, \omega_t) \in Proj_{x_t, y_t, \phi_t}\big(\tilde{\mathcal{D}}_t^l(x_{t-1}, \omega_t)\big) \Big\}, \quad t \in [T], \tag{31}$$

where $\phi_T = 0$. In this algorithm, let x_t^l be an optimal solution obtained by solving the LP-subproblem (31) during a forward step (Line 5) for iteration l and stage t. In a backward step (Line 8), the algorithm solves the LP-subproblem (31) given x_t^l and ω_t^i to obtain a cut in the form of

Algorithm 2 DA-SDDP-DP

```
1: Initialize l ← 1; x_0 ← initial state; ω_1 ← data at the first stage; Ω_1 := {ω_1}; K_t^l ← 0 for t = 1, ..., T − 1;
2: while (satisfying none of stopping conditions) do
3:   Sample a scenario path ξ^l ∈ Ξ := Ω_1 × ··· × Ω_T
4:   for t ∈ [T] do                                                    ▷ Forward Step
5:     Solve t-stage LP-subproblem (29) given x_{t−1} = x_{t−1}^l and ω_t = ξ_t^l
6:   for t = T, ..., 2 do                                              ▷ Backward Step
7:     for i ∈ N_t do
8:       Solve t-stage LP-subproblem (29) given x_{t−1} = x_{t−1}^l and ω_t = ω_t^i and obtain cut (σ_{t−1,1}^{i,l}, σ_{t−1,0}^{i,l})
9:     Add cuts (π_{t−1}^l, γ_{t−1}^l) to (t − 1)-stage LP-subproblem (29) by using cuts (σ_{t−1,1}^{i,l}, σ_{t−1,0}^{i,l}), i ∈ N_t
10:    K_{t−1}^l ← K_{t−1}^l + 1
11:  Solve LP-subproblem (29) for t = 1 to obtain the bound LB
12:  K_t^{l+1} ← K_t^l for t = 1, ..., T − 1
13:  l ← l + 1
14: return LP-subproblems, LB
```

Benders cut, $(\sigma^l_{t-1,1}(x^l_{t-1},\omega^i_t), \sigma^l_{t-1,0}(x^l_{t-1},\omega^i_t))$, where $\sigma^l_{t-1,1}(x^l_{t-1},\omega^i_t)$ and $\sigma^l_{t-1,0}(x^l_{t-1},\omega^i_t)$ are the optimal dual multipliers, associated with the constraints $\sum_{h\in H_t}\zeta^h_{t,0} = 1$ and $\sum_{h\in H_t}\zeta^h_{t,3} = x^l_{t-1}$, respectively. These cuts $(\sigma^l_{t-1,1}(x^l_{t-1},\omega^i_t), \sigma^l_{t-1,0}(x^l_{t-1},\omega^i_t))$ for $i \in \mathcal{N}_t$ are used to derive a cut $(\pi^l_{t-1}, \gamma^l_{t-1})$; the cut $(\pi^l_{t-1}, \gamma^l_{t-1})$ takes the form of (15b), if we solve a DRR-MSDP, and it takes the form of (26), if we solve a DRA-MSDP. Then, its copies for $h \in H_t$ are added to the $(t-1)$-stage LP-subproblem (Line 9). In Line 11, it computes the lower bound by solving the first-stage LP-subproblem. DA-SDDP-DP repeats this procedure until a stopping condition is met. Both the DRR- and DRA-SDDP-DP algorithms, defined as Algorithm 2 with cut $(\pi^l_{t-1}, \gamma^l_{t-1})$ obtained as above for DRR-MSDPs and DRA-MSDPs, respectively, have the finite convergence that can be proved using Theorems 1 and 3, respectively.

Remark 4. In the implementation of Algorithm 2, we can establish the LP-subproblems once and reuse them in each iteration, without the need for repeated construction, by adding constraints $\zeta^h_{t,4} - (\pi^k_t)^\top \zeta^h_{t,1} - \gamma^k_t \zeta^h_{t,0}$ to the LP-subproblems as needed (in Line 9).

6.2 Application of Proposition 1 for solving DRR- and DRA-MSIPs using Hierarchical Relaxations

In this section, we present a hierarchy of relaxations ranging from linear relaxation to tight extended formulations for each stage to solve DRR- and DRA-MSIPs by applying Proposition 1. For the ease of exposition, let y_t be continuous. Then, the subproblem (11) of DRR- and DRA-MSIPs for iteration l and stage t can be rewritten as

$$\min\Big\{ f_t(x_t, y_t, \xi^l_t) + \phi^l_t : \Big(x_{t,j} = 0 \vee x_{t,j} = 1, j = 1, \ldots, d_x\Big) \\ \bigwedge \Big(\phi_t - (\pi^k_t)^\top x_t \geq \gamma^k_t, k \in [K^l_t],\ A_t(\xi^l_t)x_t + B_t(\xi^l_t)y_t \geq b_t(\xi^l_t) - C_t(\xi^l_t)x_{t-1}\Big)\Big\}. \quad (32)$$

We use $\mathcal{D}_t^l(x_{t-1}, \xi_t^l)$ and $\mathcal{D}_t^{l,LP}(x_{t-1}, \xi_t^l)$ to denote the feasible region of the subproblem (32) and its linear programming relaxation, respectively. A relaxation of $\mathcal{D}_t^l(x_{t-1}, \xi_t^l)$ can be defined as follows.

$$\mathcal{D}_t^{l,s}(x_{t-1}, \xi_t^l) := \mathcal{D}_t^{l,LP}(x_{t-1}, \xi_t^l) \cap \{x_{t,j} = 0 \vee x_{t,j} = 1,\ j \in [s]\},\ \text{for } s \in [d_x]. \tag{33}$$

It is easy to see that $\mathcal{D}_t^{l,s}(x_{t-1}, \xi_t^l)$ for $s = d_x$ is equivalent to the original set. Moreover,

$$\mathcal{D}_t^{l,LP}(x_{t-1}, \xi_t^l) \supseteq \mathcal{D}_t^{l,1}(x_{t-1}, \xi_t^l) \supseteq \cdots \supseteq \mathcal{D}_t^{l,d_x}(x_{t-1}, \xi_t^l) = \mathcal{D}_t^l(x_{t-1}, \xi_t^l),$$

and thus $conv(\mathcal{D}_t^{l,1}(x_{t-1}, \xi_t^l)) \supseteq \cdots \supseteq conv(\mathcal{D}_t^{l,d_x}(x_{t-1}, \xi_t^l)) = conv(\mathcal{D}_t^l(x_{t-1}, \xi_t^l))$. This provides a hierarchy of relaxations of the feasible region $\mathcal{D}_t^l(x_{t-1}, \xi_t^l)$ of DRR- and DRA-MSIPs. The tight extended formulation of the convex hull of the relaxations can be obtained using Proposition 1.

Proposition 2. *The convex hull of the set $\mathcal{D}_t^{l,s}(x_{t-1}, \omega_t)$ is the projection of the following set onto the (x_t, y_t, ϕ_t)-space for any $x_{t-1} \in \{0,1\}^{d_x}$ and $\omega_t \in \Omega_t$:*

$$\begin{aligned}
\Bigg\{ &\sum_{h \in [|\mathcal{J}_t^s|]} \zeta_{t,0}^h = 1,\ \sum_{h \in [|\mathcal{J}_t^s|]} \zeta_{t,1}^h - x_t = 0,\ \sum_{h \in [|\mathcal{J}_t^s|]} \zeta_{t,2}^h - y_t = 0,\ \sum_{h \in [|\mathcal{J}_t^s|]} \zeta_{t,3}^h = x_{t-1}, \\
&A_t(\omega_t)\zeta_{t,1}^h + B_t(\omega_t)\zeta_{t,2}^h + C_t(\omega_t)\zeta_{t,3}^h \geq b_t(\omega_t), \quad h \in [|\mathcal{J}_t^s|], \\
&\zeta_{t,1,j}^h = 0,\ j \in J_1^h,\ \zeta_{t,1,j}^h = \zeta_{t,0}^h,\ j \in J_2^h, \quad h \in [|\mathcal{J}_t^s|], \\
&\zeta_{t,4}^h - (\pi_t^k)^\top \zeta_{t,1}^h - \gamma_t^k \zeta_{t,0}^h \geq 0, \quad h \in [|\mathcal{J}_t^s|], k \in [K_t^l], \\
&x_t \in \mathbb{R}_+^{d_x}, y_t \in \mathbb{R}_+^{d_y}, \phi_t \in \mathbb{R}_+, \\
&\zeta_{t,0}^h \in \mathbb{R}_+, \zeta_{t,1}^h \in \mathbb{R}_+^{d_x}, \zeta_{t,2}^h \in \mathbb{R}_+^{d_y}, \zeta_{t,3}^h \in \mathbb{R}_+^{d_x}, \zeta_{t,4}^h \in \mathbb{R}_+,\ h \in [|\mathcal{J}_t^s|] \Bigg\},
\end{aligned} \tag{34}$$

where $\mathcal{J}_t^s := \{(J_1^h, J_2^h) : h \in [|\mathcal{J}_t^s|]\}, s \in [d_x], t \in [T]$, be a set of all pairs of disjoint sets (J_1, J_2) such that $J_1, J_2 \subseteq [d_x], J_1 \cap J_2 = \emptyset$, and $|J_1 \cup J_2| = s$.

Proof. For $s \in [d_x]$, the set $\mathcal{D}_t^{l,s}(x_{t-1}, \omega_t), t \in [T]$, is given by the following disjunctive constraints:

$$\bigvee_{(J_1,J_2) \in \mathcal{J}_t^s} \Bigg(x_{t,j} = 0,\ j \in J_1,\ x_{t,j} = 1,\ j \in J_2,\ \phi_t - (\pi_t^k)^\top x_t \geq \gamma_t^k,\ k \in [K_t^l], \\
A_t(\omega_t)x_t + B_t(\omega_t)y_t \geq b_t(\omega_t) - C_t(\omega_t)x_{t-1} \Bigg), \tag{35}$$

where $\mathcal{J}_t^s = \{(J_1, J_2) : J_1, J_2 \subseteq [d_x], J_1 \cap J_2 = \emptyset, |J_1 \cup J_2| = s\}$. By applying Proposition 1 to this disjunctive set, we obtain the tight extended formulation (34). □

Proposition 2 provides the following relaxation that can be used to generate cuts in Line 8 of Algorithm 1:

$$\tilde{Q}_t^{l,s}(x_{t-1}^l, \xi_t^l) = \min\{f_t(x_t, y_t, \xi_t^l) + \phi_t^l(x_t) : (x_t, y_t) \in Proj_{x_t,y_t}(\mathcal{D}_t^{l,s}(x_{t-1}^l, \xi_t^l))\},$$

where $\tilde{Q}_t^{l,1}(x_{t-1}^l, \xi_t^l) \leq \cdots \leq \tilde{Q}_t^{l,d_x}(x_{t-1}^l, \xi_t^l) = \hat{Q}_t^l(x_{t-1}^l, \xi_t^l)$. For $s = d_x$, a Benders cut obtained by solving this relaxation is a supporting hyperplane satisfying (13). For the smaller values of s, the Benders cut does not necessarily support the value function $\hat{Q}_t^l(\cdot)$, but the relaxations are computationally easier to solve. It is worth noting that the value of s can be adjusted either before or during the execution of the algorithm to address this trade-off between the computational effort and the effectiveness of the cuts.

Remark 5. The above relaxations can be readily extended to the case where y_t is mixed-binary variables. Furthermore, if y_t is mixed-integer, a hierarchy of relaxations can be derived by employing the binary expansion as discussed in Remark 2.

7 Computational Tests

In this section, we present computational results of utilizing the presented algorithms, the DRR-SDDP-C, DRR-SDDP-R, DRA-SDDP-C, and DRA-SDDP-R algorithms, to solve multistage stochastic maximum flow interdiction problem (MS-MFIP) and multistage stochastic facility location interdiction problem (MS-FLIP), discussed in Section 1.2, with distributional ambiguity. These algorithms are implemented in Julia 1.8 where subproblems, coefficient-computing problems (16), and distribution separation problem (25) are solved using Gurobi 9.5 with an optimality tolerance of 10^{-4}. We also integrate our implementation of the inner functionalities of our DA-SDDP algorithms with `SDDP.jl` [14] package because of two reasons: (a) To ensure fair comparison with benchmark algorithms [30], and (b) To be consistent with the research community, thereby making it convenient for future computational and applied users of these algorithms. We conducted all tests on a machine equipped with an Intel Core i7 processor (3.8 GHz), utilizing a single thread, and 32 GB RAM.

Throughout all test instances, we consider Wasserstein ambiguity set with the l_1 norm. Note that with the Wasserstein ambiguity set, the distribution separation problem (25) becomes a linear program and the subproblem (11) in the DRR-SDDP-R algorithm becomes a mixed-binary linear program.

7.1 Computational Results for Instances of MS-MFIP with Distributional Ambiguity

Using a tractable reformulation of MS-MFIP, presented in Appendix B, i.e., (39), we obtain DRR and DRA variants of MS-MFIP (8), denoted by DRR- and DRA-MFIPs. In the following sections, we report the results from the comparative analysis of the algorithms and demonstrate the significance of risk-aversion and risk-receptiveness for MS-MFIP.

7.1.1 Instance generation and computational results

Networks are randomly generated, following the method presented in Cormican et al. [13]. First, we place all nodes, excluding the source and sink nodes, in a grid pattern. Next, we establish connections between the leftmost and rightmost nodes in the grid to the source and sink nodes, respectively, using non-interdictable arcs with infinite capacity. Then, every pair of adjacent nodes in the grid is connected by an arc. Horizontal arcs are oriented from left to right, and vertical arcs, connecting the leftmost or rightmost nodes, are oriented from up to down. The orientations of the remaining arcs are randomly chosen. To avoid trivial solutions, e.g., removing all horizontal arcs in the same column, we set 80 percent of all arcs to be interdictable. Following the above procedure, we generate two distinct networks with different sizes. For the first network and the second network, we sample realizations of the random capacity of each arc uniformly distributed on $[30, 60]$ and $[20, 90]$, respectively, to construct the support Ω_t of size $|\Omega|$ for each stage t. The Wasserstein ball size parameter ϵ is set to 30 and the interdiction budget for each stage is set to one, i.e., the interdictor can remove up to one arc for each stage.

Table 1 summarizes the details of the test instances. Each row corresponds to a single instance that is labeled accordingly under the *Instance* column. The naming convention NI-i-T-$|\Omega|$ is used for an instance with the ith network among the two aforementioned networks, T stages, and $|\Omega|$ realizations per stage. The labels $|N| \times |A|$ and #Scenario denote the number of nodes and arcs of the network and the total number of scenario paths, respectively. For termination conditions, we specify a time limit of 3 hours for all algorithms. Also, there is an early-termination condition

Table 1: Details of DRR- and DRA-MFIP instances

Instance	$\|N\| \times \|A\|$	T	$\|\Omega\|$	#Scenario
NI-1-3-5	37 x 73	3	5	25
NI-1-3-10			10	100
NI-1-3-15			15	225
NI-1-4-5		4	5	125
NI-1-4-10			10	1000
NI-1-4-15			15	3375
NI-1-5-5		5	5	625
NI-1-5-10			10	10000
NI-1-5-15			15	50625
NI-1-6-5		6	5	3125
NI-1-6-10			10	100000
NI-1-6-15			15	759375
NI-2-3-5	52 x 106	3	5	25
NI-2-3-10			10	100
NI-2-3-15			15	225
NI-2-4-5		4	5	125
NI-2-4-10			10	1000
NI-2-4-15			15	3375

based on the convergence of the lower bound. In particular, the algorithm is stopped if the lower bound fails to improve for 100 consecutive iterations.

In table 2, we report the lower bounds and the solution times in seconds for each algorithm, labeled LBound and Time (s), respectively. The results indicate that the DRA-SDDP-C algorithm provides better lower bounds and solution times than the DRA-SDDP-R algorithm for all instances. On average, the DRA-SDDP-C algorithm is 17.2 times faster than the DRA-SDDP-R algorithm. This performance advantage increases to 26.4 times (on average) for eight instances where both algorithms provide the same lower bounds. The results show that the DRA-SDDP-R algorithm's performance is more susceptible to T and $|\Omega|$ than the DRA-SDDP-C algorithm. For example, as we increase $|\Omega|$ from 5 to 15 for NI-1-3-5, the DRA-SDDP-R algorithm's solution time increases by 48.2 times, while the DRA-SDDP-C algorithm's solution time increases by 2.7 times. Similarly, when we increase T to 5 and 6 from 3 for NI-1-3-5, the DRA-SDDP-R algorithm failed to solve the 5-stage and 6-stage instances—NI-1-5-5 and NI-1-6-5—within the time limit, whereas the DRA-SDDP-C algorithm was able to solve all instances within the time limit. This is mainly because the DRA-SDDP-R algorithm adds a significantly larger number of cuts ($|\Omega|^2$ cuts for every subproblem solved) for each iteration, which increases the solution times for subproblems. Regarding DRR-MFIP, the DRR-SDDP-C algorithm outperforms the DRR-SDDP-R algorithm for all test instances. In terms of solution time, the DRR-SDDP-C algorithm is, on average, 22.3 times faster than the DRR-SDDP-R algorithm, and this advantage increases to 41.3 times for seven instances where both the algorithms provide the same bounds. This is because the DRR-SDDP-R algorithm solves larger subproblems that arise from (24c) and (24d), incorporated by the linearization and the ambiguity set, respectively.

7.1.2 Impact of risk-aversion and risk-receptiveness

To demonstrate the impact of risk-aversion and risk-receptiveness on MS-MFIP, we present the results from out-of-sample tests, which are conducted as follows. We first sample realizations of

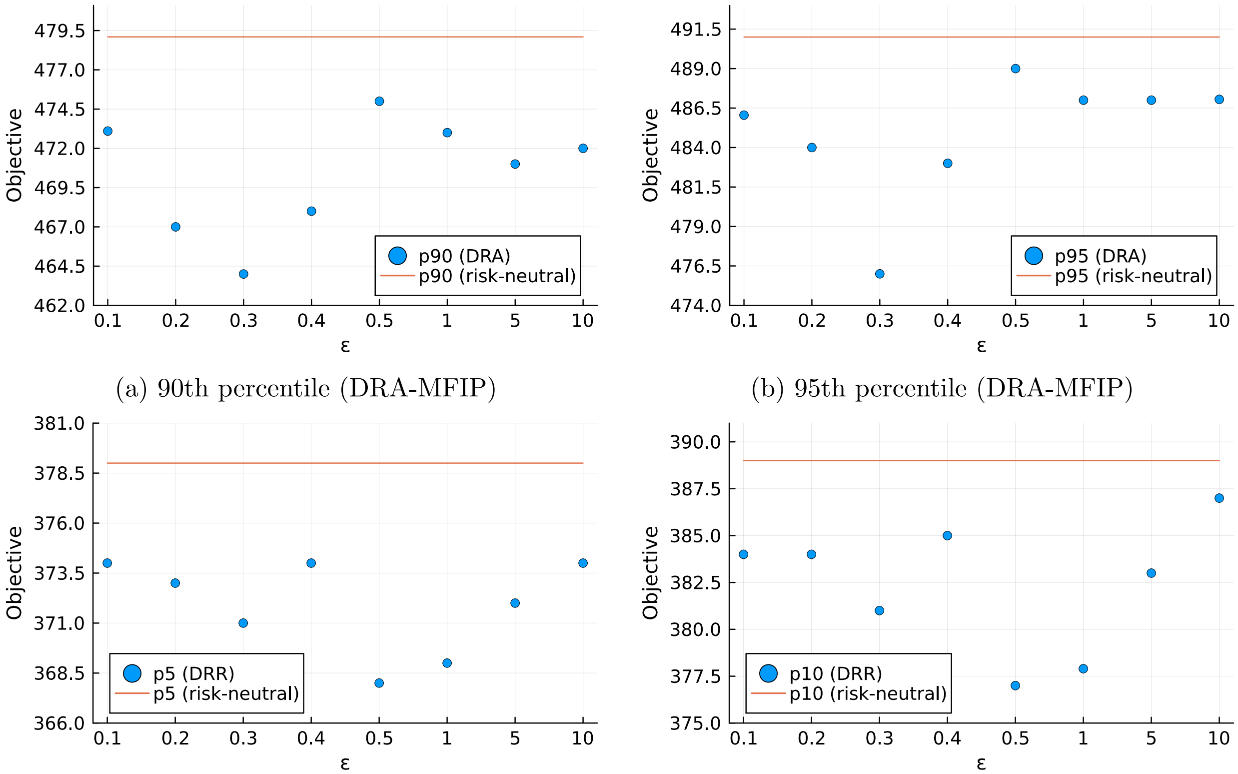
Table 2: Performance comparison of algorithms for DRR- and DRA-MFIP instances

	DRA-SDDP-C		DRA-SDDP-R		DRR-SDDP-C		DRR-SDDP-R	
Instance	LBound	Time (s)	LBound	Time (s)	LBound	Time (s)	LBound	Time (s)
NI-1-3-5	537.08	39.1	537.08	224.1	528.10	41.1	528.10	585.0
NI-1-3-10	535.00	50.3	535.00	1653.2	527.05	60.3	527.05	2172.9
NI-1-3-15	534.76	105.0	534.76	10800+	528.11	103.1	528.11	6774.8
NI-1-4-5	655.58	169.1	655.58	4794.1	644.76	173.8	644.76	10800+
NI-1-4-10	670.44	547.2	662.12	10800+	655.45	453.2	600.68	10800+
NI-1-4-15	657.02	401.6	641.21	10800+	643.02	525.3	590.32	10800+
NI-1-5-5	735.84	1019.2	729.85	10800+	727.05	1455.2	645.57	10800+
NI-1-5-10	718.36	1443.9	676.38	10800+	701.54	1958.1	583.27	10800+
NI-1-5-15	792.14	704.2	774.19	10800+	769.31	746.9	650.32	10800+
NI-1-6-5	737.51	3058.8	705.42	10800+	725.69	3491.4	631.55	10800+
NI-1-6-10	762.90	5066.5	672.96	10800+	744.84	6853.9	593.40	10800+
NI-1-6-15	748.97	10800+	594.48	10800+	715.37	10800+	576.73	10800+
NI-2-3-5	1018.51	54.2	1018.51	115.3	1013.16	61.7	1013.16	906.5
NI-2-3-10	892.61	73.9	892.61	994.0	884.66	96.8	884.66	3543.9
NI-2-3-15	903.28	148.8	903.28	263.7	891.10	171.7	891.10	10244.7
NI-2-4-5	1181.45	410.0	1181.45	9978.1	1173.41	436.3	1136.69	10800+
NI-2-4-10	1096.20	1426.5	1062.48	10800+	1086.77	1574.4	981.57	10800+
NI-2-4-15	1105.07	3387.2	1027.92	10800+	1089.51	3553.0	824.64	10800+

the capacity over the stages. Then, the DRA-MFIP and DRR-MFIP instances obtained over this sample ($\{\Omega_t : t \in [T]\}$) are solved using the algorithms. The resulting subproblems along with the cuts generated by the algorithms define a policy for MS-MFIP, i.e., the decision rule that selects the set of arcs to remove to minimize the maximal flow of the network for each stage. We simulate the DRR and DRA policies using scenario paths of the capacity that are sampled independently from the realizations used in solving the problems.

Throughout the out-of-sample tests, we consider the problem with 4 stages, 30 realizations per stage, 3000 independently-sampled scenario paths, and a network with 30 nodes and 60 arcs. All realizations are sampled from a truncated normal distribution where the mean is 30, the standard deviation is 5, and the capacity belongs to the interval $[10, 50]$. The policies are generated for the set of different Wasserstein ball size parameter ϵ belonging to $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 1, 5, 10\}$. We present the results from the simulations in fig. 1: figs. 1a and 1b show the 90th and 95th percentiles of the objective values obtained by the DRA policies, and figs. 1c and 1d show the 5th and 10th percentiles of the objective values obtained by the DRR policies. Each figure contains the result from the policy generated by SDDiP with $\epsilon = 0$, i.e., the risk-neutral policy, as an orange horizontal line for comparison.

The DRA policies give the smaller 90th and 95th percentiles compared to the risk-neutral policy. As ϵ increases from 0.1 to 0.3, the corresponding policies yield the lower 90th and 95th percentiles, indicating the better out-of-sample performance. As ϵ increases further, to $0.5, 1, 5$, and 10, the overall performance of the policies drops, yet their 90th and 95th percentiles remain smaller than those of the risk-neutral policy. This demonstrates that the DRA policies achieve the robustness of interdiction solutions by incorporating conservatism over unfavorable realizations in $\Omega_t, t \in [T]$. On the other hand, the DRR policies give the smaller 5th and 10th percentiles compared to the risk-neutral policy. This indicates that the DRR policies yield more effective interdiction solutions than the risk-neutral policy for certain scenario paths and reflect a more pessimistic perspective of the network user on the network performance. Consequently, this pessimistic view can identify



(a) 90th percentile (DRA-MFIP)

(b) 95th percentile (DRA-MFIP)

(c) 5th percentile (DRR-MFIP)

(d) 10th percentile (DRR-MFIP)

Figure 1: Plots of percentiles from out-of-sample tests on DRA-MFIP (90th and 95th percentiles) and DRR-MFIP (5th and 10th percentiles)

the network vulnerabilities that are unnoticed by the risk-neutral policy. The level of pessimism increases as ϵ increases from 0.1 to 0.5, except for $\epsilon = 0.4$. However, as it continues to increase up to $\epsilon = 10$, the pessimistic view on the performance diminishes.

7.2 Computational Results of MS-FLIP with Distributional Ambiguity

In this section, we report the computational performances of the algorithm for DRR and DRA variants of MS-FLIP (10), referred to as DRR-FLIP and DRA-FLIP, respectively. To generate instances, we use the following method that is similar to the one used in Yu and Shen [44]. We first randomly sample $(L + M)$ points from a 100×100 grid and place demand points and facilities. For each demand point $l \in [L]$, we randomly choose μ_l from a uniform discrete distribution $[20, 40]$. Then, to construct the support Ω_t of size $|\Omega|$ for each stage t, we randomly sample $|\Omega|$ realizations of the random demand for $l \in [L]$ from a truncated normal distribution, where the mean is μ_l, the standard deviation is $\sigma_l = \mu_l/4$, and the truncation interval is $[1, 60]$. For all test instances, we set the Wasserstein ball size ϵ to 10 and set the interdiction budget for each stage to one, i.e., $r_t = 1, t \in [T]$. The details of the test instances are given in table 3. Each instance, denoted by LI-i-T-$|\Omega|$, involves the ith network out of two randomly generated networks, T stages, and $|\Omega|$ realizations per stage. For each row of the table, the labels $L \times M$ and #Scenario denote the number of demand points and facilities and the number of total scenario paths, respectively. The termination conditions are identical to those used for the DRR- and DRA-MFIP instances.

Table 4 presents the upper bounds and solution times in seconds obtained by each algorithm

Table 3: Details of DRR- and DRA-FLIP instances

Instance	$L \times M$	T	$\|\Omega\|$	#Scenario
LI-1-3-10	10 x 20	3	10	100
LI-1-3-20			20	400
LI-1-3-50			50	2500
LI-1-4-10		4	10	1000
LI-1-4-20			20	8000
LI-1-4-50			50	125000
LI-1-5-10		5	10	10000
LI-1-5-20			20	160000
LI-1-5-50			50	6250000
LI-2-3-10	15 x 30	3	10	100
LI-2-3-20			20	400
LI-2-3-50			50	2500
LI-2-4-10		4	10	1000
LI-2-4-20			20	8000
LI-2-4-50			50	125000
LI-2-5-10		5	10	10000
LI-2-5-20			20	160000
LI-2-5-50			50	6250000

under the label UBound and Time (s). The numbers in each row of the table correspond to the result for a single instance. Note that smaller bounds are better since they are upper bounds. For DRA-FLIP, the DRA-SDDP-C algorithm provides the upper bounds better than the DRA-SDDP-R algorithm for all instances. Also, on average, the DRA-SDDP-C algorithm is 38.9 times faster than the DRA-SDDP-R algorithm, and this advantage increases to 48.4 times for 11 instances where both the algorithms produce the same bounds. The performance of the DRA-SDDP-R algorithm is comparatively susceptible to the number of realizations per stage. For example, the DRA-SDDP-R algorithm takes 458.7 seconds to solve LI-2-3-10, but it fails to converge within the time limit when solving LI-2-3-50. The DRA-SDDP-C algorithm, however, converges for the both instances within the time limit. When comparing the results for DRR-FLIP, the DRR-SDDP-C algorithm provides better upper bounds than the DRR-SDDP-R algorithm for all the instances. In terms of solution time, the DRR-SDDP-C algorithm is, on average, 25.8 times faster than the DRR-SDDP-R algorithm for all instances, and 26.7 times faster for 13 instances where both the algorithms produce the same bounds. As discussed in the previous section, this shows that the both DRA-SDDP-R and DRR-SDDP-R algorithms require more time to solve due to the larger subproblems resulting from the reformulation techniques.

8 Conclusion

We studied multistage stochastic integer and disjunctive programs under distributional ambiguity, considering the distributional risk-receptiveness and risk-aversion in a decision making process. We developed reformulation-based and cutting plane-based algorithms for solving distributionally risk-receptive and distributionally risk-averse multistage stochastic integer programs (DRR- and DRA-MSIPs) and provided the finite convergence analysis for these algorithms. Furthermore, we extended the algorithms for distributionally risk-receptive and distributionally risk-averse multistage stochastic disjunctive programs (DRR- and DRA-MSDPs) and then applied them to solve DRR- and DRA-MSIPs using a hierarchy of relaxations for each stage subproblem. We compared

Table 4: Performance comparison of algorithms for DRR- and DRA-FLIP instances

	DRA-SDDP-C		DRA-SDDP-R		DRR-SDDP-C		DRR-SDDP-R	
Instance	UBound	Time (s)	UBound	Time (s)	UBound	Time (s)	UBound	Time (s)
LI-1-3-10	431.29	29.1	431.29	169.7	437.28	28.4	437.28	109.1
LI-1-3-20	327.32	49.8	327.32	1761.0	333.39	41.3	333.39	630.6
LI-1-3-50	384.83	75.8	384.83	10800+	392.50	74.5	392.50	3373.3
LI-1-4-10	489.80	53.2	489.80	1170.2	498.44	48.6	498.44	622.1
LI-1-4-20	479.23	68.5	479.23	6511.9	491.03	78.7	491.03	1963.0
LI-1-4-50	651.45	105.6	651.45	10800+	668.28	106.8	668.28	10800+
LI-1-5-10	654.12	73.6	654.12	1744.3	670.83	74.5	670.83	1501.3
LI-1-5-20	774.27	89.5	774.27	4853.6	792.99	80.3	792.99	4635.5
LI-1-5-50	929.85	156.2	931.32	10800+	955.35	162.9	978.63	10800+
LI-2-3-10	452.23	74.1	452.23	458.7	456.76	78.3	456.76	333.4
LI-2-3-20	463.56	119.4	463.56	1894.1	468.56	112.7	468.56	772.8
LI-2-3-50	489.47	459.4	518.37	10800+	495.10	453.1	495.10	10800+
LI-2-4-10	500.09	369.0	500.09	10800+	506.22	378.5	506.22	4815.8
LI-2-4-20	508.09	476.5	531.57	10800+	514.78	444.9	514.78	8376.8
LI-2-4-50	556.50	761.3	721.03	10800+	564.69	692.1	1194.59	10800+
LI-2-5-10	599.37	534.6	661.32	10800+	605.32	550.7	1009.53	10800+
LI-2-5-20	816.66	879.1	904.14	10800+	828.21	1026.6	1339.16	10800+
LI-2-5-50	756.16	1732.8	949.12	10800+	767.99	1927.1	1756.89	10800+

the algorithms for DRR- and DRA-MSIPs by solving multistage stochastic network interdiction problems under distributional ambiguity that are sequential two-player non-cooperative games and have not been addressed in the literature. The computational results show that the cutting plane-based algorithms outperform the reformulation-based algorithms in terms of both the solution bounds and times. In addition, we conducted out-of-sample tests, and their results demonstrate that the DRA policies provide robust decision rules for uncertainty, while the DRR policies may reveal the network vulnerabilities that are overlooked by risk-neutral policies for uncertainty.

A Proof of Proposition 1

Let $z_t = (x_t, y_t, \phi_t)$. For any $\bar{x}_{t-1} \in \{0,1\}^{d_x}$, stage t, and iteration l, the convex hull of $\mathcal{D}_t^l(\bar{x}_{t-1}, \omega_t)$ is equivalent to the convex hull of $\mathcal{F}_t^l(\omega_t) \cap \mathcal{E}(\bar{x}_{t-1})$ where

$$\mathcal{F}_t^l(\omega_t) := \Bigg\{ (x_{t-1}, z_t) \in \mathbb{R}_+^{d_x} \times \mathbb{R}_+^{d_x+d_y+1} : \\ \bigvee_{h \in H_t} \Big(\phi_t - (\pi_t^k)^\top x_t \geq \gamma_t^k,\ k \in [K_t^l],\ A_t^h(\omega_t) x_t + B_t^h(\omega_t) y_t + C_t^h(\omega_t) x_{t-1} \geq b_t^h(\omega_t) \Big) \Bigg\}, \quad (36)$$

and $\mathcal{E}(\bar{x}_{t-1}) := \{(x_{t-1}, z_t) : x_{t-1} = \bar{x}_{t-1}\}$. We claim that $conv(\mathcal{F}_t^l(\omega_t) \cap \mathcal{E}(\bar{x}_{t-1})) = conv(\mathcal{F}_t^l(\omega_t)) \cap \mathcal{E}(\bar{x}_{t-1})$. Clearly, $conv(\mathcal{F}_t^l(\omega_t) \cap \mathcal{E}(\bar{x}_{t-1})) \subseteq conv(\mathcal{F}_t^l(\omega_t)) \cap \mathcal{E}(\bar{x}_{t-1})$. To show that $conv(\mathcal{F}_t^l(\omega_t) \cap \mathcal{E}(\bar{x}_{t-1})) \supseteq conv(\mathcal{F}_t^l(\omega_t)) \cap \mathcal{E}(\bar{x}_{t-1})$, pick any point $(\bar{x}_{t-1}, \bar{z}_t) \in conv(\mathcal{F}_t^l(\omega_t)) \cap \mathcal{E}(\bar{x}_{t-1})$. Then, there exist $(x_{t-1}^j, z_t^j) \in \mathcal{F}_t^l(\omega_t)$ and $\lambda^j \in (0,1], j = 1, \ldots, J$ such that $\sum_{j \in [J]} \lambda^j = 1, \sum_{j \in [J]} \lambda^j x_{t-1}^j = \bar{x}_{t-1}, \sum_{j \in [J]} \lambda^j z_t^j = \bar{z}_t$. Since $\bar{x}_{t-1}$ is binary and x_{t-1}^j belongs to $[0,1]^{d_x}$, this implies that $x_{t-1}^j = \bar{x}_{t-1}$ for all $j \in [J]$. Consequently, $(x_{t-1}^j, z_{t-1}^j) \in \mathcal{F}_t^l(\omega_t) \cap \mathcal{E}(\bar{x}_{t-1}), j \in [J]$, and $(\bar{x}_{t-1}, \bar{z}_t) \in conv(\mathcal{F}_t^l(\omega_t) \cap \mathcal{E}(\bar{x}_{t-1}))$. This completes the proof of the claim.

To obtain $conv(\mathcal{F}_t^l(\omega_t))$, we first use Theorem 4 and derive a tight extended formulation of $\mathcal{F}_t^l(\omega_t)$, which is given by

$$
\begin{aligned}
\widehat{\mathcal{F}}_t^l(\omega_t) := \Bigg\{ & \sum_{h\in H_t} \zeta_{t,0}^h = 1, \sum_{h\in H_t} \zeta_{t,1}^h - x_t = 0, \sum_{h\in H_t} \zeta_{t,2}^h - y_t = 0, \\
& \sum_{h\in H_t} \zeta_{t,3}^h - x_{t-1} = 0, \sum_{h\in H_t} \zeta_{t,4}^h - \phi_t = 0, \\
& A_t^h(\omega_t)\zeta_{t,1}^h + B_t^h(\omega_t)\zeta_{t,2}^h + C_t^h(\omega_t)\zeta_{t,3}^h - b_t^h(\omega_t)\zeta_{t,0}^h \geq 0, \quad h \in H_t, \\
& \zeta_{t,4}^h - (\pi_t^k)^\top \zeta_{t,1}^h - \gamma_t^k \zeta_{t,0}^h \geq 0, \quad h \in H_t, k \in K_t^l, \\
& x_t \in \mathbb{R}_+^{d_x}, y_t \in \mathbb{R}_+^{d_y}, x_{t-1} \in \mathbb{R}_+^{d_x}, \phi_t \in \mathbb{R}_+, \\
& \zeta_{t,0}^h \in \mathbb{R}_+, \zeta_{t,1}^h \in \mathbb{R}_+^{d_x}, \zeta_{t,2}^h \in \mathbb{R}_+^{d_y}, \zeta_{t,3}^h \in \mathbb{R}_+^{d_x}, \zeta_{t,4}^h \in \mathbb{R}_+, h \in H_t \Bigg\}.
\end{aligned} \tag{37}
$$

Since $\mathcal{F}_t^l(\omega_t)$ is unbounded, the projection of the above formulation (37) onto the (x_{t-1}, z_t)-space is the closed convex hull of $\mathcal{F}_t^l(\omega_t)$. Consider $|H_t|$ polyhedra defined by disjunctive constraints for $h \in H_t$ in (36). They are nonempty and have identical recession cones. This implies that $conv(\mathcal{F}_t^l(\omega_t))$ is a polyhedron, and thus the closed convex hull of $\mathcal{F}_t^l(\omega_t)$ is equivalent to $conv(\mathcal{F}_t^l(\omega_t))$. Hence, using the tight extended formulation (37), the convex hull of $\mathcal{D}_t^l(\bar{x}_{t-1}, \omega_t)$ is given by $Proj_{x_{t-1},z_t}(\widehat{\mathcal{F}}_t^l(\omega_t)) \cap \mathcal{E}(\bar{x}_{t-1})$, which is equivalent to the projection of formulation (30) onto the z_t-space.

B Mixed-binary linear programming reformulation of MS-MFIP

Let $\lambda \in \mathbb{R}_+^{|A|}$ and $\mu \in \mathbb{R}^{|N|}$ be the dual multipliers corresponding to (9b) and (9c), respectively. Given x_t and ω_t, the dual of the maximum flow problem (9) is

$$\min \sum_{a\in A} c_{t,a}(\omega_t)(1 - x_{t,a})\lambda_a \tag{38a}$$

$$\text{s.t. } \lambda_a + \mu_i - \mu_j \geq 0, \quad \forall (i,j) = a \in A \setminus \{(t,s)\} \tag{38b}$$

$$\lambda_{(t,s)} + \mu_t - \mu_s \geq 1 \tag{38c}$$

$$\lambda \geq 0. \tag{38d}$$

Note that by the max-flow min-cut theorem a feasible dual solution $\lambda_a \in \{0,1\}$ indicates whether arc $a \in A$ is in a solution cut-set or not. Therefore, we can introduce a variable $w_a \in \mathbb{R}_+$ to replace $x_{t,a}\lambda_a$ in the objective function such that $\lambda_a - w_a \geq 0, x_{t,a} - w_a \geq 0, w_a - \lambda_a - x_{t,a} \geq -1$, and obtain a linear programming reformulation of the dual (38). Note that we can omit the constraints $w_a - \lambda_a - x_{t,a} \geq -1$ for $a \in A$, since we only need to restrict the upper bound on w_a. By embedding the resulting linear program into the bellman equation (8), we obtain a mixed-integer linear programming reformulation:

$$\min \sum_{a\in A}\left(c_{t,a}(\omega_t)\lambda_a - c_{t,a}(\omega_t)w_a\right) + \sum_{i\in\mathcal{N}_{t+1}} p^i_{t+1}Q_{t+1}(x_t,\omega^i_{t+1}) \tag{39a}$$

$$\text{s.t. } (8b)\text{–}(8d) \tag{39b}$$

$$(38b)\text{–}(38d) \tag{39c}$$

$$\lambda_a - w_a \geq 0, \quad \forall a \in A \tag{39d}$$

$$x_{t,a} - w_a \geq 0, \quad \forall a \in A \tag{39e}$$

$$w \geq 0. \tag{39f}$$

C Tight extended formulation of disjunctive set

Consider nonempty polyhedra $\mathcal{G}^h := \{x \in \mathbb{R}^n_+ : A^h x \geq b^h\}, h \in H$, and let $\mathcal{G} := \bigcup_{h\in H}\mathcal{G}^h$. Then, the closed convex hull of $\mathcal{G}$ is given as follows.

Theorem 4. *(Theorem 2.1 in Balas [1]) The closed convex hull of $\mathcal{G}$ is the projection of the following extended formulation onto the x-space.*

$$\begin{aligned} x - \sum_{h\in H}\zeta^h &= 0, \\ A^h\zeta^h - b^h\zeta^h_0 &\geq 0, \\ \sum_{h\in H}\zeta^h_0 &= 1, \\ (\zeta^h,\zeta^h_0) &\geq 0, \quad h \in H. \end{aligned} \tag{40}$$

References

[1] Egon Balas. *Disjunctive programming.* Springer, 2018.

[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] Manish Bansal and Sanjay Mehrotra. On solving two-stage distributionally robust disjunctive programs with a general ambiguity set. *European Journal of Operational Research*, 279(2):296–307, 2019.

[4] Manish Bansal and Yingqiu Zhang. Scenario-based cuts for structured two-stage stochastic and distributionally robust p-order conic mixed integer programs. *Journal of Global Optimization*, 81(2):391–433, 2021.

[5] Güzin Bayraksan and David K Love. *Data-driven stochastic programming using phi-divergences*, pages 1–19. INFORMS TutORials in Operations Research, 2015.

[6] John R Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, 33(5):989–1007, 1985.

[7] John R Birge and François Louveaux. *Introduction to stochastic programming.* Springer Science & Business Media, 2011.

[8] Pierre Bonami and François Margot. Cut generation through binarization. *Mathematical Programming*, 154:197–223, 2015.

[9] Esra Büyüktahtakın. Stage-t scenario dominance for risk-averse multi-stage stochastic mixed-integer programs. *Annals of Operations Research*, 309(1):1–35, 2022.

[10] Junyu Cao and Rui Gao. Contextual decision-making under parametric uncertainty and data-driven optimistic optimization, 2021.

[11] Claus C Carøe and Rüdiger Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1):37–45, 1999.

[12] Richard L Church, Maria P Scaparra, and Richard S Middleton. Identifying critical infrastructure: The median and covering facility interdiction problems. *Annals of the Association of American Geographers*, 94(3):491–502, 2004.

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

[14] Oscar Dowson and Lea Kapelevich. Sddp. jl: a julia package for stochastic dual dynamic programming. *INFORMS Journal on Computing*, 33(1):27–33, 2020.

[15] John C Duchi, Peter W Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 46(3):946–969, 2021.

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

[17] Daniel Duque and David P Morton. Distributionally robust stochastic dual dynamic programming. *SIAM Journal on Optimization*, 30(4):2841–2865, 2020.

[18] Martin Dyer and Leen Stougie. Computational complexity of stochastic programming problems. *Mathematical Programming*, 106:423–432, 2006.

[19] Harsha Gangammanavar and Manish Bansal. Stochastic decomposition method for two-stage distributionally robust linear optimization. *SIAM Journal on Optimization*, 32(3):1901–1930, 2022.

[20] Harsha Gangammanavar and Suvrajeet Sen. Stochastic dynamic linear programming: A sequential sampling algorithm for multistage stochastic linear programming. *SIAM Journal on Optimization*, 31(3):2111–2140, 2021.

[21] Rui Gao and Anton Kleywegt. Distributionally robust stochastic optimization with wasserstein distance. *Mathematics of Operations Research*, 48(2):603–655, 2023.

[22] Jun-ya Gotoh, Michael Jong Kim, and Andrew Lim. A data-driven approach to beating saa out-of-sample, 2021.

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

[24] Sumin Kang and Manish Bansal. Distributionally risk-receptive and risk-averse network interdiction problems with general ambiguity set. *Networks*, 81(1):3–22, 2023.

[25] Daniel Kosmas, Thomas C Sharkey, John E Mitchell, Kayse Lee Maass, and Lauren Martin. Multi-period max flow network interdiction with restructuring for disrupting domestic sex trafficking networks. *Annals of Operations Research*, pages 1–64, 2022.

[26] Guanghui Lan. Complexity of stochastic dual dynamic programming. *Mathematical Programming*, 191(2):717–754, 2022.

[27] 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.

[28] Guglielmo Lulli and Suvrajeet Sen. A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems. *Management Science*, 50(6):786–796, 2004.

[29] Ajay Malaviya, Chase Rainwater, and Thomas Sharkey. Multi-period network interdiction problems with applications to city-level drug enforcement. *IIE Transactions*, 44(5):368–380, 2012.

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

[31] Hideaki Nakao. *Distributionally Robust Optimization in Sequential Decision Making.* Ph.d. thesis, University of Michigan, Ann Arbor, 2021.

[32] Hideaki Nakao, Ruiwei Jiang, and Siqian Shen. Distributionally robust partially observable markov decision process with moment-based ambiguity. *SIAM Journal on Optimization*, 31(1):461–488, 2021.

[33] Viet Anh Nguyen, Soroosh Shafieezadeh-Abadeh, Man-Chung Yue, Daniel Kuhn, and Wolfram Wiesemann. Optimistic distributionally robust optimization for nonparametric likelihood approximation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[34] Jangho Park and Güzin Bayraksan. A multistage distributionally robust optimization approach to water allocation under climate uncertainty. *European Journal of Operational Research*, 306(2):849–871, 2023.

[35] Mario VF Pereira and Leontina MVG Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(1):359–375, 1991.

[36] Andrew B Philpott, Vitor L de Matos, and Lea Kapelevich. Distributionally robust SDDP. *Computational Management Science*, 15(3):431–454, 2018.

[37] Utsav Sadana and Erick Delage. The value of randomized strategies in distributionally robust risk-averse network interdiction problems. *INFORMS Journal on Computing*, 35(1):216–232, 2023.

[38] Herbert Scarf. *A min-max solution of an inventory problem*, pages 201–209. Stanford University Press, 1958.

[39] Alexander Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, 2011.

[40] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 3rd edition, 2021.

[41] Jun Song and Chaoyue Zhao. Optimistic distributionally robust policy optimization, 2020.

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

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

[44] Xian Yu and Siqian Shen. Multistage distributionally robust mixed-integer programming with decision-dependent moment-based ambiguity sets. *Mathematical Programming*, 196(1):1025–1064, 2022.

[45] Shixuan Zhang and Xu Andy Sun. Stochastic dual dynamic programming for multistage stochastic mixed-integer nonlinear optimization. *Mathematical Programming*, 196(1):935–985, 2022.

[46] Jikai Zou, Shabbir Ahmed, and Xu Andy Sun. Stochastic dual dynamic integer programming. *Mathematical Programming*, 175(1):461–502, 2019.